Title:	Simulated Exhibit Experience (SEE)	
Team Name:	Agents of Science	
Team Number:	2	
Members:	Andrew Brown	*ab6370@gmail.com*
		(561) 222-1561
	Jacob Millsaps	*jmillsaps@knights.ucf.edu*
		(321) 289-6372
	Joshua Linge	*jlinge@knights.ucf.edu*
		(904) 563-1597
	Xue (Allen) Lin	*pureallenlin@gmail.com*
		(917) 868-4059
Sponsor:	Presagis	http://www.presagis.com/
	Joshua West	Joshua.West@presagis.com
	Jeremy Joseph	Jeremy.Joseph@presagis.com
Customer:	Orlando Science Center	http://www.osc.org/
	Brandan Lanman	BLanman@osc.org
	Kellen Nixon	KNixon@osc.org
Date:	Fall 2013 – Spring 2014	

Table of Content

1. Executive Summary

The goal of this project was to provide an online application which simulates at least one exhibit from the Orlando Science Center by using Unity and the Unity Web Player. Multiple users can run a standalone version of the simulation through their preferred web browser with the Unity Web Player. By putting this simulation on the web for free, the OSC can be experienced by anyone who desires, even if they don't live near Orlando. The exhibits aim to educate the users about the Engineering Design Cycle which is an implementation of the Scientific Method.

1.1 Simulated Exhibit Experience System

1.1.1 Introduction

The Simulated Exhibit Experience is the main system which connects all the exhibits together. It was developed in the Unity Game Engine. The system itself will be a virtual science center with many ways to learn and interact. The major feature of the system is that it will be a 3D environment. The user will be able to walk around in a virtual science center, interact with objects, and learn about the Orlando Science Center.

1.1.2 Requirements, Goals, and Objectives

The Orlando Science Center is a large building filled with many exhibits, but as students with limited time, we cannot recreate the entire facility. Our goal was to create a simplified virtual science center lobby within the Unity Web Player and attach exhibits to this lobby. From within this lobby, the user is able to walk around, interact with objects, and be able to enter and exit specific exhibit rooms. One exhibit was required to be recreated, but accommodations for other exhibits were made for later revisions.

1.1.3 Software Design

Section 2 talks about the design plans for the Simulated Exhibit Experience System. We have descriptions of the major functional parts for the virtual simulation. Many use cases are also listed for the actions the user may perform inside the overall system.

1.2 Gravitron Exhibit

1.2.1 Introduction

The Gravitron Exhibit at the Orlando Science Center is an exhibit to teach visitors about gravity. Given a set of tubes, create a path for the ball to fall down into the tubes and exit into a goal cup. There are different challenges such as use only a limited set of tubes or try to take the longest possible time to get into the goal cup.

1.2.2 Requirements, Goals, and Objectives

Our goal was to recreate the Gravitron Exhibit inside Unity but add some of our own virtual twists. Some basic requirements needed to recreate the exhibit are multiple and placeable tubes, a board to place tubes, a ball object to fall through the tubes, a goal location, a way to test the configuration, a way to time how long it takes for the ball to fall through the configuration, and a way to save the time.

1.2.3 Software Design

Section 3 talks about the design plans for the Gravitron exhibit. We have descriptions of all the major objects in the exhibit, including the grid background, tubes, a ball, and the cup as the main objective for the ball. Several camera modes are utilized implicitly by the system. Many use cases are also listed for the actions that users may perform inside the exhibit. A description of each major component of the Gravitron exhibit is also explained.

1.3 Pinewood Derby Exhibit

1.3.1 Introduction

The Pinewood Derby exhibit at the Orlando Science Center uses Pinewood Derby cars to teach visitors about the physics of gravity. Visitors can choose from a given set of parts to construct their derby car, then race it against other visitors on a huge track. Visitors can choose different sized wheels, adjust the wheelbase of their car, and add weights to many different locations on the car to see what effects they have on how fast the car goes.

1.3.2 Requirements, Goals, and Objectives

Our goal was to recreate the Pinewood Derby Exhibit inside Unity but add some of our own virtual twists. Some basic requirements needed to recreate the exhibit are constructing a vehicle from several different parts, racing the vehicle on a track, recording the results of the race, and then modifying the parts to achieve a better result.

1.3.3 Software Design

Section 4 talks about the design plans for the Pinewood Derby exhibit. We have descriptions of all the major objects in the exhibit, including the derby car and its parts, the workbench, and the track. An alternate camera mode is given for the user to view their car running down the track. Many use cases are also listed for the actions that player may perform inside the exhibit. A description of the physical model for the derby car is also explained.

1.4 Other exhibits

These other exhibits were potential exhibits for the simulation but were not developed due to time and resource constraints. They could be potentially created by future groups.

1.4.1. Wind tubes

The objective of the Wind Tubes exhibit is to build a device out of paper which will float for the longest time.

1.4.2. Water Works

The objective of the Water Works exhibit is to build a device which will hold the maximum amount of weight while still staying afloat.

1.4.3. Dinosaur Dig Site

The objective of the Dinosaur Dig Site exhibit is to plan out where you think the dinosaur bones are going to be by observation before digging.

1.5 Personal motivations

Andrew

For those who don't know, Eyewitness is the name of a series of science videos produced by Dorling Kindersley and the BBC. Growing up, these videos were heavily utilized in my elementary and middle school science classes and had a big impact on me as a child. Part of my interest in the series was based purely on the introduction. The intro begins with the camera flying into a 3d wireframe of a camera and into a CGI museum. As the camera moves through the digital museum, various animals are seen roaming about such as birds, fish, a tiger, and an elephant. Surrounding the animals are the typical museum exhibits such as models, a dinosaur skeleton, and images on the walls. But even these static objects feel alive when combined with the moving animals and the motion of the camera.

For a time, I believed that the museum in the intro was a physical place and I would've done anything to go to it. Of course, that was until I found out that it didn't exist anywhere physically. However, I'm still very interested in creating this sort of magical environment where all sorts of things seem to come alive and exist in much closer proximity than they ever would in real life.

I realize that the scope of this project is to recreate the real-life version of the science museum and not create a fantasy version. However, I'll still have this intro in mind throughout the project.

Allen
I first discovered Unity few months ago through browsing different game engines to determine which one will be great for developing a personal game that I can be proud of. It has always been a dream of mine to make one of the most terrifying horror game in existence. Though it might be a giant goal to accomplish, however by taking one step at a time, I'll be able to develop the potential skills necessary to make it happen. Thus, learningy and it's game logic and algorithm is part of that first step. Not only that, I'll be able to interact with professionally made 3D models for this project, so my desire to contribute grows exponentially! Personal growth aside, the Virtual Exhibit Hall for the OSC is meant for public education that will be interactive online through any web browsers, so to know that it is supporting mass educational purposes will also be a great satisfaction. I support free public educations to those who have no access to them, thus picking this project is like signing a consequence-free contract, so to speak.

Jacob
Since I was a child, I always loved to take things apart and see how they worked. I was always very curious. When I started playing video games, I felt the same desire to figure out how video games and software work. This curiosity is what eventually led me to learn how to program, and to develop video games. I taught myself a lot about programming before coming to UCF, and my education here has taught me so much more. Not just about programming, but many other fields of study as well. In a way, I got to experience what science is all about: testing ideas about how the world works, learning new information, and putting that knowledge to use. My education has increased my appreciation for science, and when I learned more about this project for the Orlando Science Center to bring some of their exhibits online, I became interested in working on this project. This project allows me to do what I enjoy and make a positive contribution at the same time. Bringing the OSC exhibits online will allow young people to learn more about science and critical thinking, in a fun and interactive environment. Before this project, I had never visited the OSC, but now that I have learned about what they do and their mission, I am very motivated to work on this project, and instill the same curiosity in young people that has driven myself.

Josh
I enjoy learning and teaching others. Creating a virtual exhibit hall to inspire people to learn is motivating to me. Not only will I get to apply the physics and science knowledge I've obtained through my degree, but I will also learn new technologies, such as the

Unity game engine and the C# scripting language. I have always had an interest in video games and am excited to get the chance to work with a team to create one as a group. This project may be large, but it is a challenge I want to accomplish.

1.6 Project Terminologies

The following list contains the terminologies that we will be using in this design document. Some words are interchangeable.

- System - representing the Virtual Science Center as a whole.
- Simulated Exhibit Experience is interchangeable with the OSC Virtual Exhibits Hall
- Simulation - to represent the Simulated Exhibit Experience
- User - The person using the system.
- Player - The object within the system that the user controls.
- Tubes or Pipes - The physical pipes such as plumbing typed.
- High scores or Leaderboard - A table that displays results
- Scores or Standings - The final results of an attempt
- Abyss - The area outside of the walls of the Virtual Science Center.

1.7 Software used for Development

Many different programs will be used for the development of the Simulated Exhibit Experience System.

1.7.1 Unity3D Game Engine

Unity is a powerful cross-platform game engine for developing 2D and 3D games. Most of the physics are already included so most of our focus is on:

- scripting
- object and environment modeling
- vector placements
- manipulation of physics
- lighting

Below are some of the major features of Unity and how they will be used or may be useful for our project:

Rendering (graphic engines)

Unity uses either Direct3D or OpenGL depending on the user's environment.

Art assets and file formats (managed through Unity's graphical user interface)

Initially, Presagis mentioned an artist intern from Full Sail University would provide the necessary 3D models to use in Unity. Since the process of making models is time consuming, we created our own placeholder objects for the Virtual Exhibit Hall using Blender, Adobe Photoshop, and/or GIMP. Ultimately, Christopher Culverwell from Presagis provided upgraded versions of our models which we used in the final product.

Scripting

Unity has the MonoDevelop IDE integrated within. MonoDevelop is a free GNOME IDE designed for C# and other .NET languages. C# is the language we used for our scripts. There is the option for javascript as well but our sponsor suggested C#, since it is more powerful with its immense libraries.

Unity Asset Server

Unity also provides controls for lighting, camera, and audio. All we need to do is instruct Unity with our scripts.

Platform

We developed the Virtual Exhibit Hall for the web browsers, thus we needed the Unity Web Player plugins for testing purposes.

Physics

Unity has built-in support for Nvidia's PhysX physics engine to provide us with realistic actions and reactions for our Virtual Hall Exhibit game objects.

Unity Licenses

We used Unity Free for this project. Although initially, we considered asking Presagis to develop a few things for us with the features in Unity Pro, ultimately, those weren't necessary.

1.7.2 MonoDevelop

MonoDevelop is a cross-platform IDE for C# and other .NET Languages. It is also the primary IDE for the Unity3D Game Engine.

C# Programming Language

- Object-oriented programming language
- Provides powerful libraries to use in Unity
- Allows great control over objects
- Vector manipulations in 3D space

1.7.3 Blender

Blender is a free, open-source 3D modeling software. We used Blender to create models and environments for the virtual simulation.

1.7.4 Gimp

Gimp or the GNU Image Manipulation Program is a free image manipulation and editing program. Gimp was used for creating simple graphics and diagrams, as well as graphic user interface elements.

2 Simulated Exhibit Experience System

2.1 Introduction

The current system for which Simulated Exhibit Experience is based off of the physical Orlando Science Center and other virtual museum systems.

The Orlando Science Center is a physical location which contains a vast number of exhibits for visitors to explore and learn about. In order to continually attract new visitors, the exhibits on display will change. When new exhibits are added, the old exhibits are completely removed and are no longer available for visitors to interact with. This can be disappointing for visitors who visit the museum infrequently and expect a particular exhibit to be there.

Other virtual museum systems have been created before. Most of these are virtual tours of physical museums and only give information on what is on display without any interactability. Some virtual museums may not even have a real life counterpart giving visitors no way to truly interact with physical exhibits.

Thus, an advantage of this virtual exhibit hall is to provide a hybrid experience. The exhibits in our simulation are currently present in the physical museum so after visiting the museum, users can then access the simulation and continue to experiment with various designs. Children no longer have to constantly bug their parents to spend more on repeat admission fees in today’s tough economy.

2.2 High Level Requirements

2.2.1 Introduction and Goals

The goal of the Virtual Exhibit Hall for the Orlando Science Center project is to provide a virtual and interactive experience of multiple exhibits at the OSC. Using Unity, we will create a system that is accessible online and can be enjoyed by all age groups around the world. Multiple users will be able to run a standalone version of the simulation through their preferred web browser with the Unity Web Player. By putting this simulation on the web for free, the OSC can be experienced by anyone who desires, even if they don’t live near Orlando. The exhibits aim to educate the users about the Engineering Design Cycle which is an implementation of the Scientific Method. Each exhibit presents users with a unique challenge, requiring the application of the scientific method. Users will be able to devise and test their solutions to the exhibit challenges, and compare their scores and solutions with others.

The Orlando Science Center has many exhibits that visitors can interact with and learn from. In order to improve the experience, some exhibits have to be replaced by newer exhibits. Once an exhibit has been removed, the exhibit no longer exists and visitors can no longer experience it. The Simulated Exhibit Experience, or SEE, is being developed to create virtual exhibits that should mimic their real life counterparts. SEE can even be used to create new exhibits that may have never existed in the Orlando Science Center.

Using the proposed system, the Orlando Science Center can recreate current exhibits, preserving the experience for later visitors. This allows for exhibits to change more frequently at the Orlando Science Center without being as concerned about keeping old exhibits around that everyone enjoys . The new exhibits will attract more visitors to the physical science center and the virtual exhibits will attract more visitors to the Orlando Science Center’s website.

The virtual exhibits within the Simulated Exhibit Experience shall promote learning and education all while being fun and easy to use. There will be a variety of exhibits which have the user following the engineering design cycle of figuring out the problem, designing a solution, and testing the proposed solution.

2.2.2 User Roles and Operations

There will be only one user role for our system, the Player role. Currently there is no plan for the multi-player functionality. The Player can walk around in the environment, enter and exit exhibits, view information about the exhibits, and interact with the exhibits.

2.2.3 High Level Requirements and Features

1. Functional:

1. The program shall have a lobby area with a door or portal leading to each exhibit.
2. At least one exhibit shall be recreated
    a. Gravitron
    b. Pinewood Derby
3. The program shall be able to accommodate future additions through thorough documentation and modularity.
4. Users will move through the simulation with a first-person perspective but will not be able to see their body or limbs.
5. The user shall be able to move freely around the simulation using the WASD keys.

2. Non Functional:

1. The program shall be able to run within a browser via the Unity Web Player
2. Running on all systems that are supported by the Unity Web Player. Windows XP or later; Mac OS X 10.5 or later. Pretty much any 3D graphics card will work, depending on the complexity of our simulation. A modern web browser such as IE, Firefox, Safari, or Chrome, among others, is required.
3. The program size shall be less than 50 megabytes.
4. The program shall accurately monitor key and mouse presses and respond accordingly.
5. The program shall play the appropriate sound effects with less than 10ms of delay after the trigger for the sound occurs.
6. The program shall be accessible to third graders and older with varying degrees of technical knowledge.
7. The program shall provide a brief tutorial for basic commands and tutorials for each exhibit.
8. The program shall be available only in English.
9. The system shall be constructed without an operating budget.
10. The system shall be available and responding within 1-15 seconds
11. Errors encountered by users must not disrupt the normal operation of the system.
12. The system shall be thoroughly tested with several sets of abnormal and normal inputs
13. Response times must be within 1-15 seconds for all times of operation

3. Desired Plans:

1. More than one fully developed exhibit.
2. Online high-score table for each exhibit with 10 entries (where applicable)
3. Customizations for each exhibit, such as rockets for cars in the Pinewood Derby

4. Particle effects within the exhibits for added visuals.
5. Collect information about our users such as name, age, and email address. This will require a separate privacy policy.
6. Walkthrough of the entire Orlando Science Center building. This would include the interactive exhibits we create as well as the other exhibits which would be non-interactive.
7. Third-person perspective with customizable avatars or 15 pre-configured avatars.
8. Music that plays while in the Main Lobby.
9. A list of current events and exhibits that can be modified easily by OSC administrators

2.3 System Design

2.3.1 Operational Scenarios

1) User starts the Simulated Exhibit Experience in the Unity Web Player:

The Unity Web Player will download all the required assets and load them on the user's machine. Once the loading has finished, the Lobby Splash Screen is displayed to the user. On this screen, the user is given some text about the simulation and also the options to explore the virtual science center or to view tutorial. If the tutorial option is selected, then a series of text and images is displayed indicating how to move around and interact with the Simulated Exhibit Experience system. If the explore option is selected, the menu will disappear and the user will then be free to move about the environment.

2) User walks around environment:

The user controls the first person camera using the keyboard and mouse. The keyboard will be used for movement and the mouse will be used for the view direction. The default keyboard settings have the W key set for moving forward, the S key set for moving backwards, the A key set for moving to the left, and the D key set for moving to the right. The default mouse settings have moving the mouse forward set as looking up, moving the mouse backwards set as looking down, moving the mouse to the left set as looking to the left, and moving the mouse to the right set as looking to the right.

3) The User access the Pause Menu:

Any time the user has control within the Virtual Science Center, the pause menu may be brought up. The default method of bringing up the Pause Menu will be through the escape key. When the Pause Menu is displayed, the mouse will be unlocked from the center of the screen allowing the user to click on different pause options. The main options that will always be displayed on the pause menu are the “Resume”, “View Tutorial”, and “Exit” options.

If the “Resume” option is selected, the pause menu is removed from the display and the user is then able to explore and interact as before.

If the “View Tutorial” option is selected, the tutorial for the current area will be displayed. Areas include the Lobby Center, Gravitron, and Pinewood Derby. From the tutorial, the user can resume interactions and movement within the Virtual Science Center.

While in an exhibit room, the Exit option will return the user to the lobby. If the user choose yes, the user is then moved to the main lobby environment. If the user chooses no, the prompt is removed and the user remains on the pause menu. While in the lobby, the Exit option will close the program without prompting.

4) User enters exhibit room:

Within the lobby, if the user clicks on an exhibit door, the user will be prompted if they would like to enter that room. If they select yes, the selected exhibit will load and the user will appear in the exhibit room. If they select no, the prompt disappears and the user remains in the lobby.

Once the user enters the room, the user is then given information about the exhibit. The user is also given the option to view the tutorial or start the exhibit. If the tutorial option is selected, then a series of text and images is displayed indicating how to move around and interact with the exhibit. If the enter option is selected, the menu will disappear and the user will then be free to move about the environment and interact with the exhibit.

5) User interacts with exhibit:

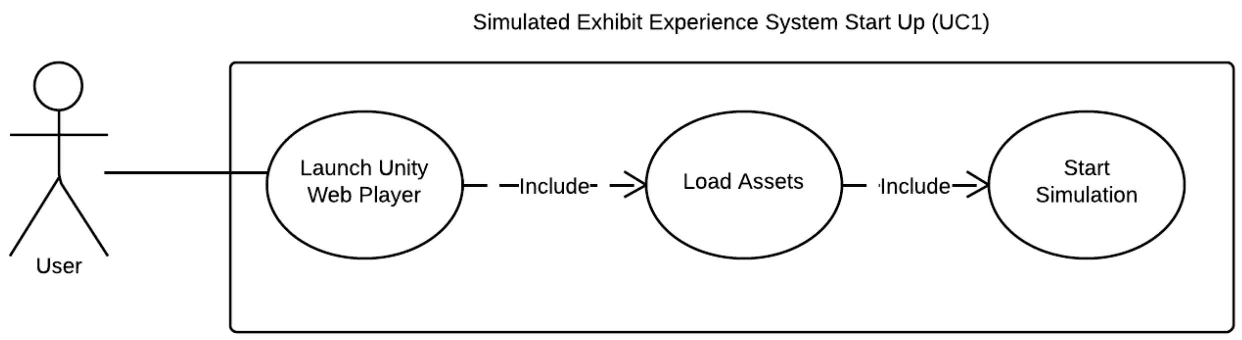
While the user is within an exhibit room, they are able to interact with the exhibit. Each exhibit will have different methods of interaction. The mouse and keyboard will be used to control these interactions.

6) User exits exhibit room:

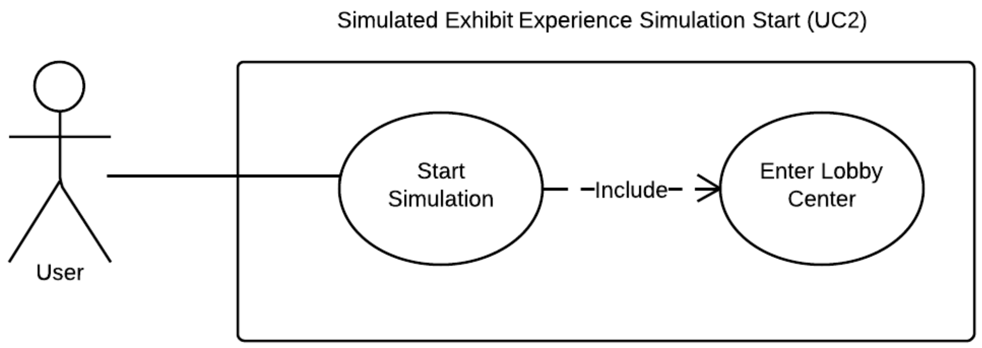
Each exhibit room will also include an exit door. If the user clicks on the exit door, they are prompted about leaving the exhibit and given two options: leave the exhibit room or stay in the exhibit room. If the leave exhibit room option is selected, the user is then moved back to the main lobby environment. If the stay in exhibit room option is selected, the prompt is removed and the user can interact with the exhibit again.

2.3.2 Simulated Exhibit Experience System Use Cases:

UC1. Simulated Exhibit Experience System Start Up – The Simulated Exhibit Experience System will be accessed through the Unity Web Player within the Orlando Science Center's website. Once the webpage has finished loading, the Unity Web Player can be launched. At start up, the Unity Web Player will download any non-cached assets for the Simulated Exhibit Experience System. Once these assets have been downloaded, they can then be loaded for the execution of the system. When loading has been complete, the simulation will start. See Use Case Diagram 1 below.

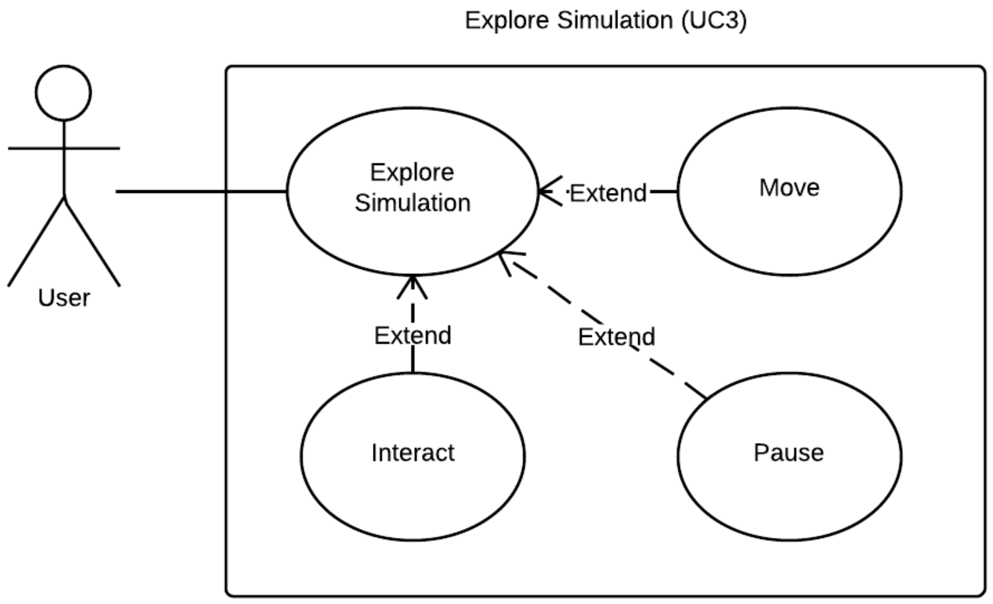


UC2. Simulated Exhibit Experience Simulation Start – Once all the assets have been downloaded and loaded, the simulation can start. The starting room for the simulation is the Lobby Center. See Use Case Diagram 2 below.

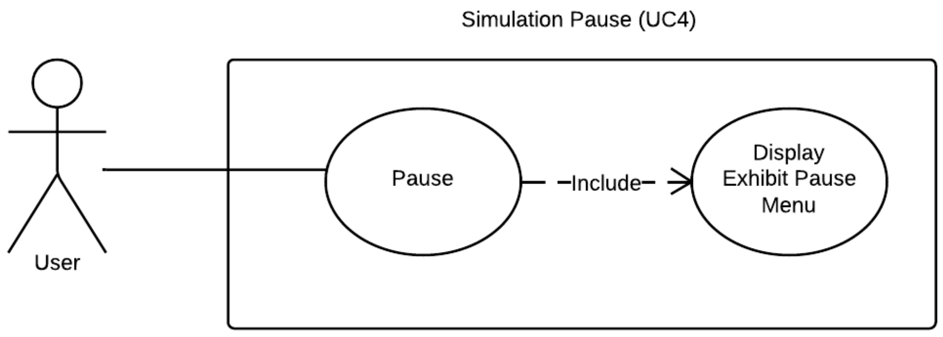


The following Use Cases are included in all areas of the Simulation regardless of which exhibit the user is in.

UC3. Explore Simulation – Throughout the simulation, the user has the ability to explore. From exploring the simulation, the user has three options: “Move”, “Interact”, and “Pause”. Movement and interactions may happen while the user is exploring, but pausing will halt the simulation and take the user to another interface. See Use Case Diagram 3 below.

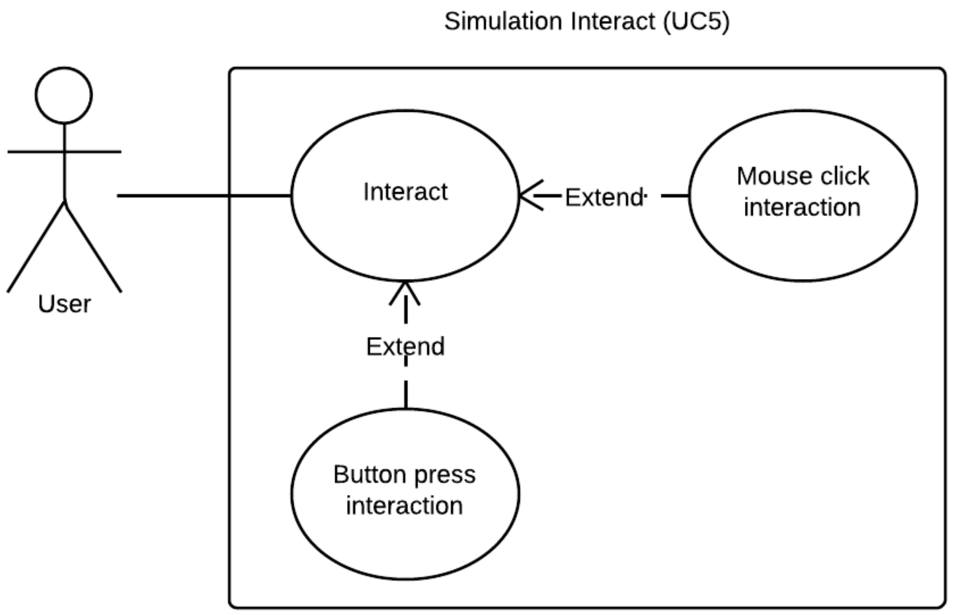


UC4. Simulation Pause – When a specific event occurs while the user is exploring the simulation, the pause menu may be brought up to display information to the user, get user input, or just to halt the simulation. When the simulation pauses, the specific exhibit’s pause menu is displayed to the user. See Use Case Diagram 4 below.

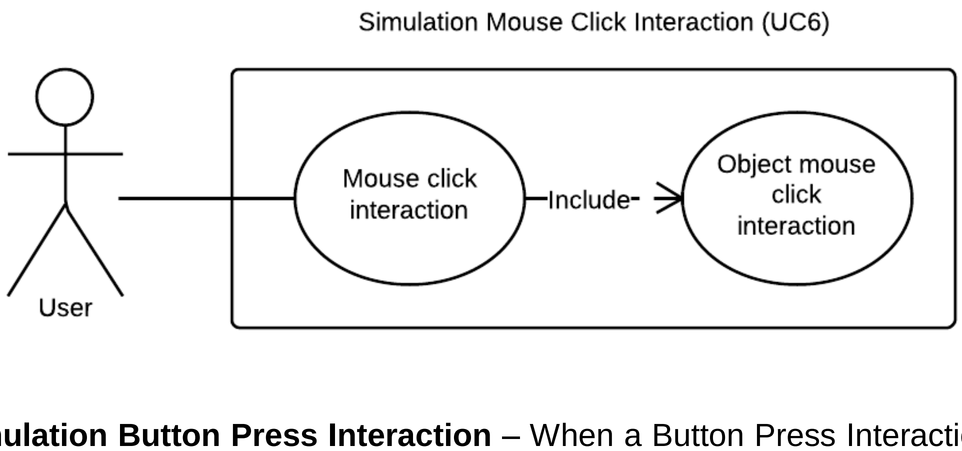


UC5. Simulation Interact – There are two ways to interact with the objects within the simulation: Mouse Click Interactions and Button Press Interactions. When you have your mouse cursor over an object and you click a mouse button, a mouse click

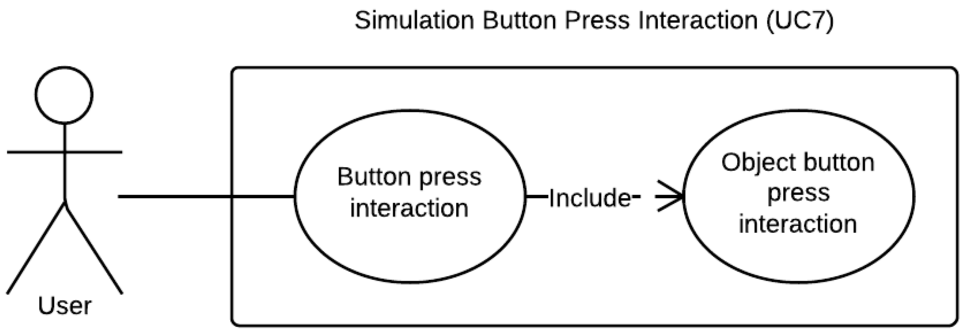
interaction occurs. When you have your mouse cursor over an object and you press a monitored keyboard button, a button press interaction occurs. See Use Case Diagram 5 below.



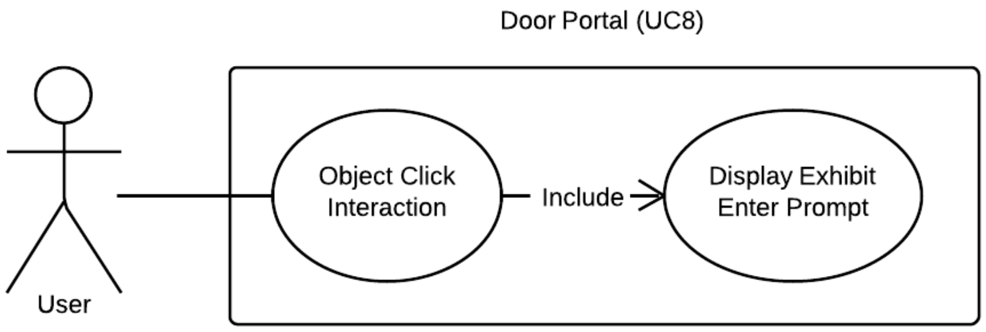
UC6. Simulation Mouse Click Interaction – When a Mouse Click Interaction occurs, the object within the center of the user’s view will then become selected and have its Mouse Click Interaction method called. Not every object will have a Mouse Click Interaction. See Use Case Diagram 6 below.


Simulation Mouse Click Interaction (UC6)
User
Mouse click interaction
Include
Object mouse click interaction

UC7. Simulation Button Press Interaction – When a Button Press Interaction occurs, the object currently selected will have its Button Press Interaction method called. Not every object will have a Button Press Interaction. See Use Case Diagram 7 below.

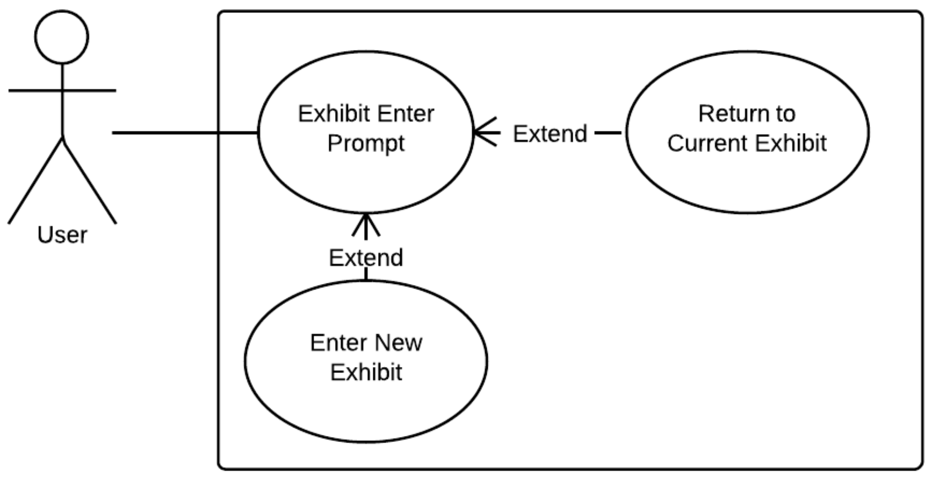

Simulation Button Press Interaction (UC7)
User
Button press interaction
Include
Object button press interaction

UC8. Door Portal – One of the objects placed within the simulation is the Door. The door acts as a portal transporting the user to another exhibit. When the user clicks on the door the Door's click interaction method is called. This method will then display an Exhibit Enter prompt depending on which exhibit the door is set to. See Use Case Diagram 8 below.



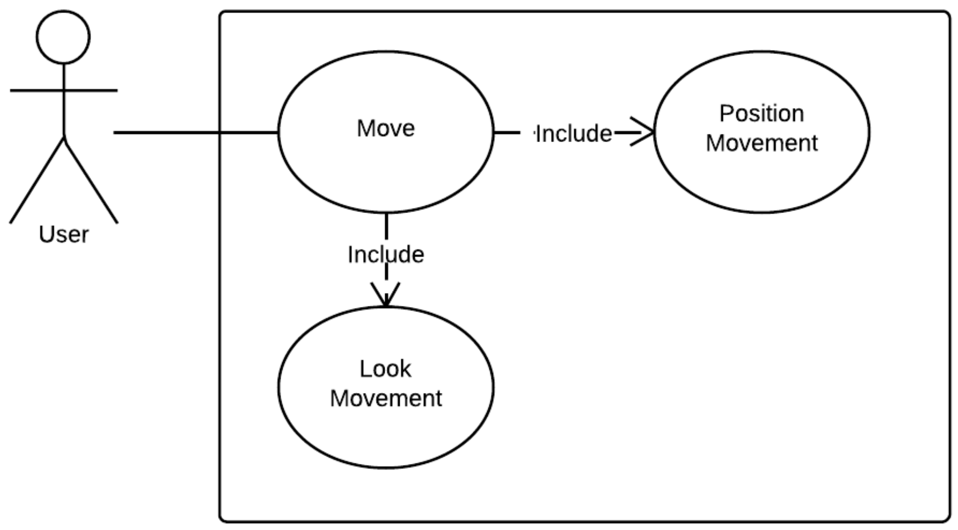
UC9. Exhibit Enter Prompt – The Exhibit Enter prompt is a dialogue box displayed from the pause menu. The Exhibit Enter prompt asks the user if they would like to move to another exhibit. The Exhibit Enter prompt also gives the user two options: "Return to the current Exhibit" and "Enter the New Exhibit". When the "Return to the current Exhibit" option is selected, the Exhibit Enter prompt is removed and the simulation resumes. When the "Enter the New Exhibit" option is selected, the user is then transported to the new exhibit. See Use Case Diagram 9 below.

Exhibit Enter Prompt (UC9)



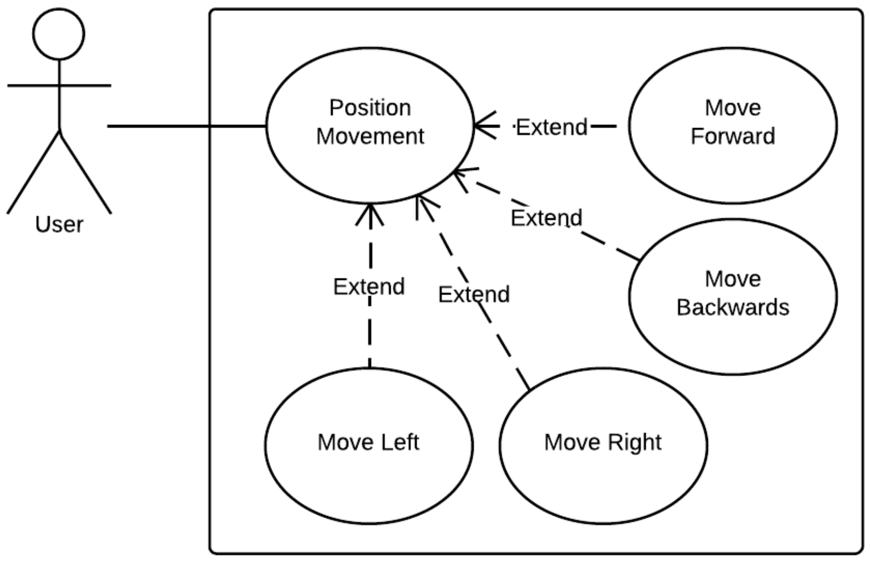
UC10. Simulation Move – There are two types of movement within the Virtual Science Center: Position Movement and Look Direction Movement. Both are controlled by different methods. See Use Case Diagram 10 below.

Simulation Move (UC10)



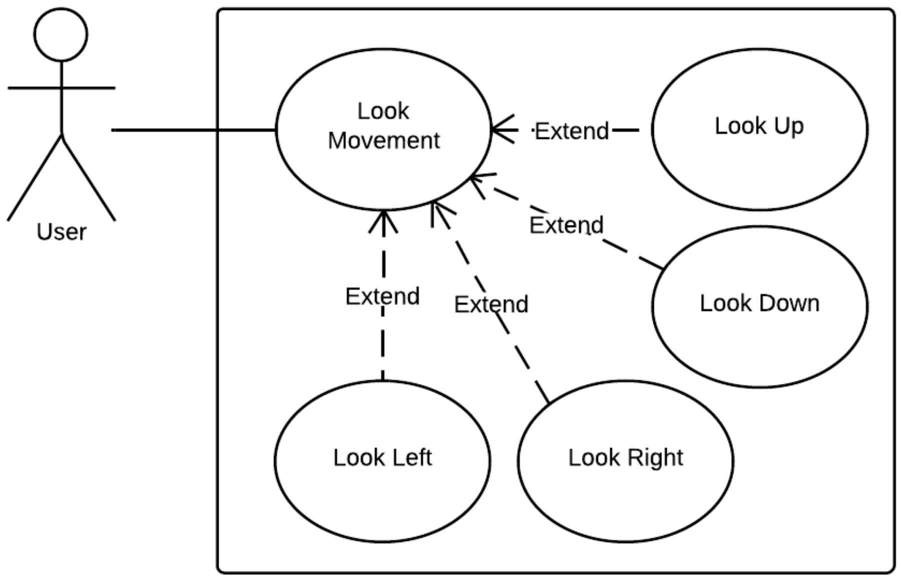
UC11. Simulation Position Movement – There are four directions of position movement for the user: forward movement, backward movement, leftward movement, and rightward movement. The user may only move into free space and not into space that is occupied by another object within the simulation. As an example, the user may not walk through walls or fall under the floor. See Use Case Diagram 11 below.

Simulation Position Movement (UC11)



UC12. Simulation Look Movement – There are four directions of look movement for the user: upward movement, downward movement, leftward movement, and rightward movement. The user's view will be restricted so that the user cannot look up or down past a certain point. See Use Case Diagram 12 below.

Simulation Look Movement (UC12)



2.4 General Framework

The General Framework consists of the components that can exist in any exhibit but are not dependent on any specific exhibit. These components make the system modular and will help future developers in creating new exhibits.

2.4.1 Player Controller

The Player Controller is the object that allows the user to move and interact within the simulation. The Player Controller consists of two main sections: Unity’s First Person Controller and a set of scripts we developed. Unity’s First Person Controller handles taking user input such as “W”, “A”, “S”, and “D” keys and mouse movement and translates them into player movement. The “W”, “A”, “S”, and “D” keys will move the Player Controller around in the world. The mouse movement will move the Player Controller’s view around in the world. The set of scripts we developed consists of the Mouse Cursor Manager, the Camera Switcher, and the Pause Menu. These items are explained in more detail below.

2.4.2 Mouse Cursor Manager

The Mouse Cursor script handles a small image (refer to as the crosshair) that appears in the center of the screen. The crosshair icon is there to let the user know what objects they are selecting and is positioned by the screen width divided by 2 and screen height divided by 2. The purpose of using screen width and height is so that the crosshair icon will always be located in the center of the user’s screen regardless of their screen resolution. A boolean variable mouseCursor will determine whether the crosshair icon will appear on the screen, the mouseCursor will be set to true initially but change to false if the camera is not in Player Controller perspective.

2.4.3 Camera Switching

The GenericSwitchCamera class will handle all cases for switching to different camera perspectives. The initial perspective is the Player Controller camera for which the user will be using to move around and interact with the environment. The Player Controller camera will be the default mode since it allows the user to be able to interact within the simulation. The script will automatically switch the camera object to another camera when the user clicks on a Button object (for example, the Drop Ball Button in the Gravitron Exhibit will switch to the Ball Camera perspective). Only one camera perspective will be active at any time since the simulation is only dedicated to one user.

2.4.4 Pause Menu Components

The Pause Menu components are a set of abstract classes that comprise of a pause menu. There are three components to a pause menu: Pause Menu, GUI Finite State Machine, and GUI State. Each Pause Menu contains a GUI Finite State machine which then contains many GUI States.

1. Pause Menu

The Pause Menu class contains all the necessary methods for pausing the simulation and displaying a menu to the user. The pause menu contains a GUI Finite State Machine that will contain all the GUI States. At the start of the simulation, the Pause Menu will create the GUI Finite State Machine, initialize it, and set it to the default state. If the child Pause Menu returns a value other than null for the override initialPause method, the simulation is paused and the GUI Finite State Machine is set to this initial state. The two main methods that the Pause Menu provides are the pause and the unPause methods. These methods can be accessed in other scripts allowing specific events to bring up specific GUI States within the Pause Menu. The pause method takes in the GUI State to display and an optional status code for that GUI State. If no GUI State is provided, the default state for the GUI Finite State Machine is loaded. Inside the pause method, Unity's time scale is set to 0, the Mouse Cursor Manager reveals the cursor, the child Pause Menu's startPause method is called, and finally the GUI Finite State Machine transitions to the GUI State. Unity handles the FixedUpdate threads for all MonoBehavior scripts by calling the method after a small unit of time. When Unity's time scale is set to 0, this unit of time becomes infinite, therefore halting all FixedUpdate calls, and pausing the simulation. Inside the unPause method, Unity's time scale is set back to 1, the Mouse Cursor Manager hides the mouse cursor, and then the child Pause Menu's exitPause method is called. The Update MonoBehavior method is called by Unity every frame regardless of the time scale. Inside the Update method, the key presses for the escape key are checked and the GUI Finite State Machine is updated. If the escape key was pressed during a frame, the TogglePause method is called. TogglePause resumes the simulation if the simulation is paused, and pauses the simulation if the simulation is in progress. The OnGUI method is called by Unity every frame regardless of the time scale. Inside the OnGUI method, if the simulation is paused, the background screen is lightened by drawing a transparent white texture over the screen and then the current GUI State is displayed by calling the GUI Finite State Machine's display method.

2. GUI Finite State Machine

The GUI Finite State Machine class contains all the necessary methods for updating and displaying GUI States and also handles transitioning between GUI States. Each GUI Finite State Machine contains a set of GUI States that the GUI Finite State Machine has transitioned to. When transitioning to a new GUI State, the state is retrieved from this set of GUI States. For ease of use, if a GUI State is requested but does not exist in the set, the requested GUI State is created and added to the set. Child GUI Finite State Machines are required to override the initialize and the getDefaultState methods. The initialize method is for setting up values for the GUI Finite State Machine. It is not required, but all the GUI Finite State Machine's GUI States can be explicitly created within this method. After a GUI Finite State Machine has been initialized, the GUI Finite State Machine loads the default state by transitioning to the state returned by the overridden getDefaultState method. While the simulation is paused, the Pause Menu will call update on the GUI Finite State Machine. Inside the update method, the GUI Finite State Machine calls updateGraphics on the current GUI State. While the simulation is paused, the Pause Menu will also call display on the GUI Finite State Machine. Inside the display method, the GUI Finite State Machine calls displayGraphics on the current GUI State. The GUI Finite State Machine handles transitioning between GUI States with the changeState method. The changeState method takes in the GUI State to transition to and the status code for that GUI State. If there is a current GUI State, the GUI Finite State Machine calls the GUI State's exit method. The current GUI State is then pushed onto a stack to save the history of states traversed. If the given GUI State to transition to is null, the default GUI State is loaded. The current GUI State is set to this new state, the status code for the GUI State is set, and then the GUI Finite State Machine calls the GUI State's enter method.

3. GUI State

The GUI State class contains all the data for displaying and updating a single state in the GUI Finite State Machine. The main functionality is contained in the displayGraphics and updateGraphics methods. If the GUI State is the current state in the GUI Finite State Machine, the updateGraphics and displayGraphics methods are called. The displayGraphics method displays all the GUI components of the GUI State. Since Unity’s GUI Buttons return whether they were clicked or not in the same line to draw them, state transitions are performed within the displayGraphics method. The updateGraphics method updates any necessary data for the GUI State. User input should be monitored within this method. When a GUI State is loaded as the current state in the GUI Finite State Machine, the enter method is called. The enter method is used to initialize any values for the GUI State so that all data is prepared and ready to be displayed. When a GUI State transitions to another state, the exit method is called by the GUI Finite State Machine before setting the new state as the current state. The exit method is used to deinitialize any values for the GUI State so that all data is freed and cleaned up before finalizing the transition. While the GUI Finite State Machine is transitioning to a new GUI State, the GUI Finite State Machine sets the status code of the state to be transitioned to. This status code is used to combine similar states in code to reduce redundancy. Every GUI State must implement the isValidStatus method to know if the given status code is valid or not. Most GUI States will represent only one state, but when a GUI State represents more than one state, there must be a status code for each represented state. Two helper methods are provided in the GUI State class: drawButton and confirmationBox. The drawButton method takes in a rectangle area to draw the button, the GUI Content to draw on the button, and the button class itself. The drawButton method draws a button with the given GUI Content at the given rectangle’s location but makes sure to consider and remove the overflow regions of the button so that two buttons may be drawn over top of each other, but the clickable region’s do not overlap. The confirmationBox method takes in a display message, a accept message, and a decline message. The confirmationBox method then draws a dialog box with the given display message, a button with the accept message, and a button with the decline message. The dialogue box will dynamically adjusting the size based on the length of the message. This method will return a value based on whether the user clicks on one of the displayed buttons: -1 for decline, 0 for no action, and 1 for accept.

Exhibit Enter States

For each exhibit within the simulation, a GUI State called the Exhibit Enter State is used to transfer to that exhibit from another. These GUI States are generic and are added to all SEE GUI Finite State Machines. The Exhibit Enter State consists of a message dialogue with two buttons. The message displayed is a question asking the user if they want to travel to the specific exhibit room. The two buttons presented are the “Yes” and “No” buttons. When the “Yes” option is selected, the user is taken to the specific exhibit. When the “No” option is selected, the user is taken back to the previous interface. There are two ways to enter into an Exhibit Enter State: by clicking on the Door object and by transitioning from another GUI State. When the user enters an Exhibit Enter State by clicking on a Door object, the “No” option will resume the simulation. When the user enters an Exhibit Enter State by transitioning from another GUI State, the “No” option will take the user back to the previous GUI State.

2.4.5 Physical Button Objects

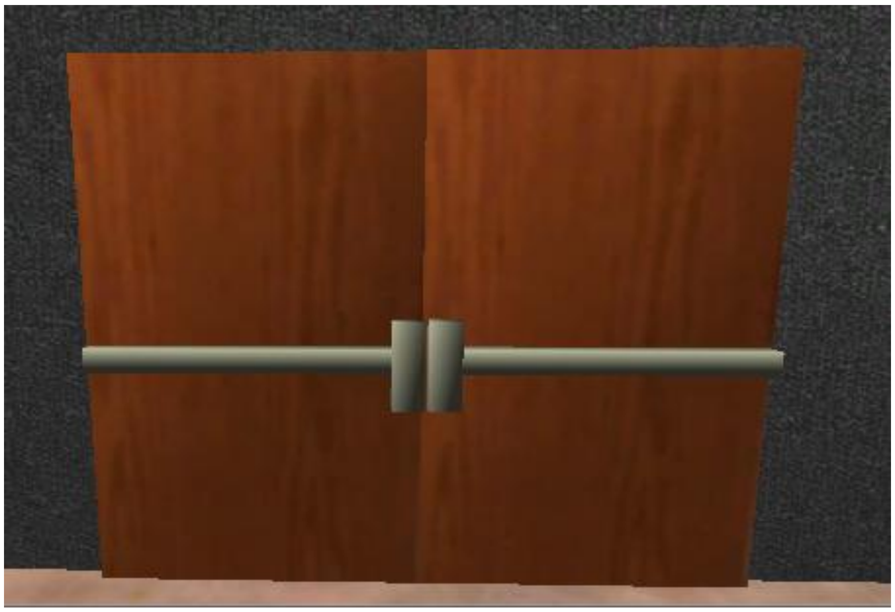
The Pause Menu Components handle all the GUI buttons presented to the user, but the Physical Button Objects handle all buttons that exist within the exhibits themselves. The PhysicalButton abstract class is the interface for handling buttons within the exhibits. The PhysicalButton class is an extension of MonoBehavior that implements the OnMouseUpAsButton method, OnMouseEnter method, and the OnMouseExit method. Every class that then extends PhysicalButton is required to override the ClickAction method, HoverEnter Method, and the HoverExit method. When the user places the cursor over a Physical Button object, the OnMouseEnter MonoBehavior method is called by Unity. Inside the OnMouseEnter method, HoverEnter is called. When the user removes the cursor from a Physical Button object, the OnMouseExit MonoBehavior method is called by Unity. Inside the OnMouseExit method, the HoverExit method is called. These two methods create a simple interface for handling mouse hover events. As an example, all GravitronPhysicalButtons will change their display texture when the mouse is hovering over it, and then restore the original texture after the mouse has stopped hovering. When the user has the mouse cursor over the PhysicalButton, presses the mouse, and also releases the mouse while still over the PhysicalButton, Unity calls the OnMouseUpAsButton MonoBehavior method. Inside the OnMouseUpAsButton method, the ClickAction method is called. The ClickAction method is how PhysicalButtons handle the click event.

2.4.6 Doors

The Doors within the Simulated Exhibit Experience consist of two components, the 3D Model and the DoorButton script. Together, these two components allow users to travel between exhibits.

1. Door Model

Prototype Door Model by Andrew Brown



Final Door Model by Chris Culverwell

2. Door Button

The Generic Door object comes attached with the DoorButton script. The DoorButton script is an extension of the PhysicalButton class. While the Door itself is not a button, it acts as a button when the user clicks on the Door. When placing Generic Doors into an exhibit within the Unity Editor, two values need to be initialized for the Generic Door to work properly. The first object is the PlayerController. The second object is a string representing the name of the exhibit scene that the Generic Door should teleport the user after the user confirms that they would like to go to that exhibit. Since the DoorButton is an extension of PhysicalButton, it overrides the ClickAction, HoverEnter, and HoverExit methods. The Start, FixedUpdate, and OnGUI MonoBehavior methods are also implemented. The Start method initializes values by getting the Pause Menu for the scene, getting the First Person Controller child object from the Player Controller, and also getting the Exhibit Enter State for the specified exhibit from the Pause Menu's GUI Finite State Machine. The FixedUpdate method calculates the distance between this Generic Door and the First Person Controller object which will be used later. The HoverEnter method only sets a boolean value to true, saving the data that the mouse is over the Door. The HoverExit method resets this boolean value back to false since the mouse is no longer over the Door. The OnGUI method displays a label representing the exhibit name that this door will transport the user to if clicked. The label is only displayed if the mouse is hovered over the door and if the user is within 10 in game units of the door. If the door does not contain a string representing an exhibit name, a red "X" will be displayed again notifying the user that this door will not take the user to another Exhibit. The ClickAction method will pause the simulation and display the Exhibit Enter State for the door's exhibit, only if the user clicks on the door within 10 in game units. If this door is located in the Lobby Center, the door location is saved so that the Player Controller can be positioned outside this door when the user returns to the Lobby Center. Another way to bring up the Exhibit Enter State is for the Player Controller to collide with the door.

2.4.7 Editor Scripts

Two Unity Editor scripts were created to help with the development and interactions of the system. These scripts include the SceneNames script and the CodeGenerator script.

1. Scene Names

While working with transportation between exhibits, it became apparent that there needed to be a way to access Unity Scene names during execution. The Scene Names script is the solution to solving this problem. The Scene Names script is a script attached to the Player Controller. It contains an array of strings representing all the enabled scenes in the build settings. Other scripts may access this string array, but the array itself is only changed in the Unity Editor. From within the Unity Editor, clicking the drop down menu on the SceneNames component of the Player Controller gives you a list of options. The first of these options is “Update Scene Names”. When selected, all of the enabled scenes contained in the Editor build settings are saved to the array of strings. This code was taken from the public domain and was originally written by Bunny83 in his post on answers.unity3d.com.

http://answers.unity3d.com/questions/33263/how-to-get-names-of-all-available-levels.html

The second option from the drop down menu on the SceneNames component is “Generate Files”. When selected, all scenes that are missing generated files will have them generated. See Code Generator below for more detail.

2. Code Generator

To help make creating new exhibits easier, we created a system to help simplify the process. The Code Generator script generates code for enabled scenes that do not already have the code. The Code Generator currently generates Exhibit Enter States and Pause Menu templates for each scene.

Exhibit Enter State Code Generator

As stated above in the GUI State section, each scene uses an Exhibit Enter State transport the user from one exhibit to another. These Exhibit Enter States all contain the same overall display but with only a different display name. To generate the code for an Exhibit Enter State, the Code Generator first reads a template for the Exhibit Enter State, and then replaces specific keywords within the template text to then make it compatible with the new exhibit scene. These keywords include the class name, the scene file name, and the scene display name. After the template text has been updated for the specific scene, it is saved into the Exhibit Enter State asset folder, thus completing the code generation for Exhibit Enter States.

Pause Menu Code Generator

The Pause Menu for each scene consists of many components. To make the creation of Pause Menus for new exhibits easier, the Code Generator will create a template for the new exhibit to help the developers. Four classes are generated for a new exhibit's pause menu: the Exhibit Pause Menu, the Exhibit GUI Finite State Machine, the Exhibit GUI State, and the Exhibit Main State. The Pause Menu Code Generator works similar to the Exhibit Enter State Code Generator except that four different files are used as templates and each edited individually. After each template text has been updated for the scene, the code files are saved into the Exhibit's folder within the asset folder. The directory will be created if it did not exist before code generation.

2.4.8 Sound Effects

The following events will trigger specific sound effects:

- Walking footsteps
- Selecting a tube in the Gravitron
- Ball rolling in the Gravitron tubes
- Ball entering Goal Cup
- Car passing finish line in Pinewood Derby

2.5 Final Assessment

For the final product we presented to our Senior Design committee, our system fully satisfied all functional and non-functional requirements.

Although we did not fully complete the Pinewood Derby exhibit, our first requirement states that we would have at least one functional exhibit which is what we achieved with the fully functional Gravitron.

Initially, several of our requirements were related to OSC's request that the simulation be hosted on their web site. However, after discussing this with our sponsor, Presagis, they informed us that they would take the project after we finished, improve it, and then hand it off to OSC.

3 Lobby Center

3.1 Introduction

The purpose of the Lobby Center is to serve as a central portal hub to the other exhibits as well as to provide a sense of exploration to the simulation. Even though each room is contained in its own independent Unity scene, the user can access them all through the Lobby Center. The Lobby Center will contain decorative objects that are similar to the actual Orlando Science Center (OSC) to give users a sense of familiarity while exploring the virtual OSC. The initial plan for the Lobby Center was a dome-shaped building encasing the overall layout in the scene, but this changed over time based on feedback from the team members since we used a phased prototyping process. The final design for the Lobby Center closely resembles the Orlando Science Center.

3.2 Lobby Center High Level Requirements

3.2.1 User Roles and Operations

There will be only one user role for our system, the Player role. The Player can walk around in the environment, enter and exit exhibits, view information about the exhibits, and interact with the exhibits.

3.2.2 High Level Requirements and Features

1. Functional:

1. The Lobby Center shall look like the Orlando Science Center.
2. There shall be doors leading to other exhibits.
3. Inaccessible areas shall be blocked off.

2. Non Functional:

1. The system shall play the correct sound effect when specific events occur, such as footstep sounds while walking

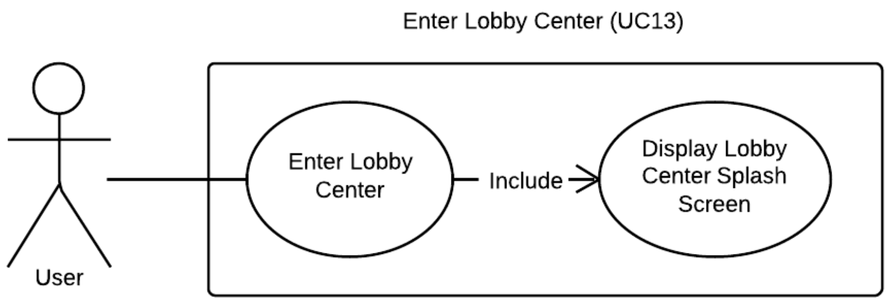
3. Desired Plans:

1. Simulated crocodiles and turtles in first floor pond.

3.3 System Design

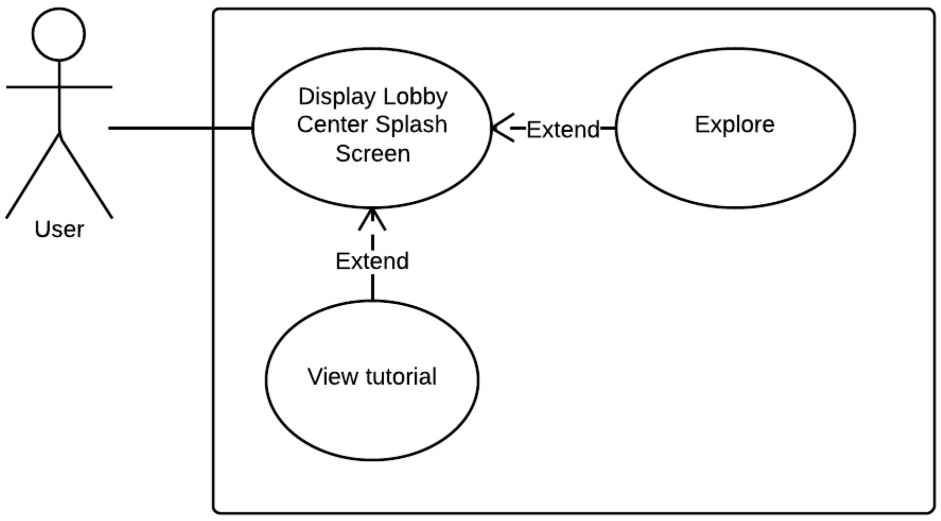
3.3.1 Lobby Center Use Cases

UC13. Enter Lobby Center – When the simulation has finished loading and is ready to begin, it places the user in the Lobby Center. When the user has entered the Lobby Center, the simulation pauses and the Lobby Center Splash Screen is displayed. See Use Case Diagram 13 below.



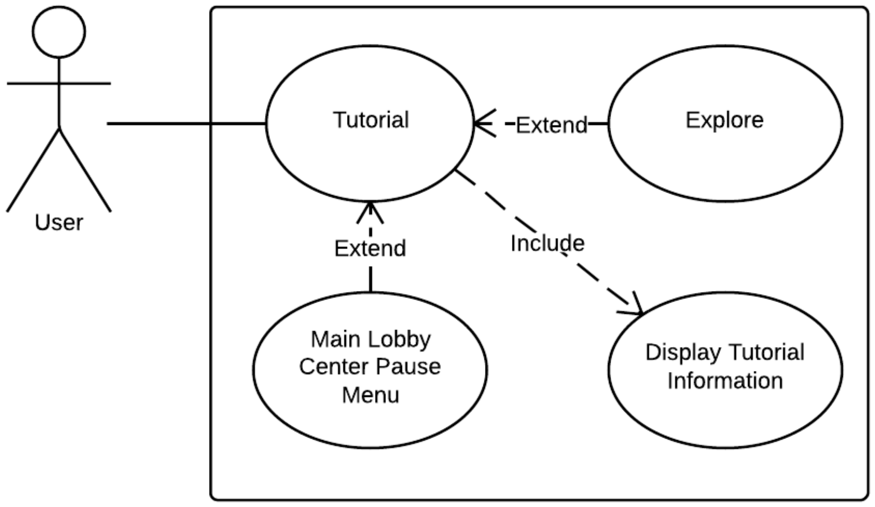
UC14. Lobby Center Splash Screen – Information on the Lobby Center is displayed on the Splash Screen. Information includes a welcome message and instructions on what to do next. The user will also be given two options: “Explore the Virtual Science Center” and “View tutorial”. Each of these options are selectable and will display a different interface upon selection. See Use Case 14 below.

Lobby Center Splash Screen (UC14)



UC15. Lobby Center Tutorial – When the “View Tutorial” button is selected from the Lobby Center splash screen, a new interface is brought up with information on how to explore the Lobby Center. Information will include how to move the camera around and how to move the player around within the Lobby Center. The Lobby Center Tutorial gives the user two options: “Explore” and “Main Menu”. See Use Case Diagram 15 below.

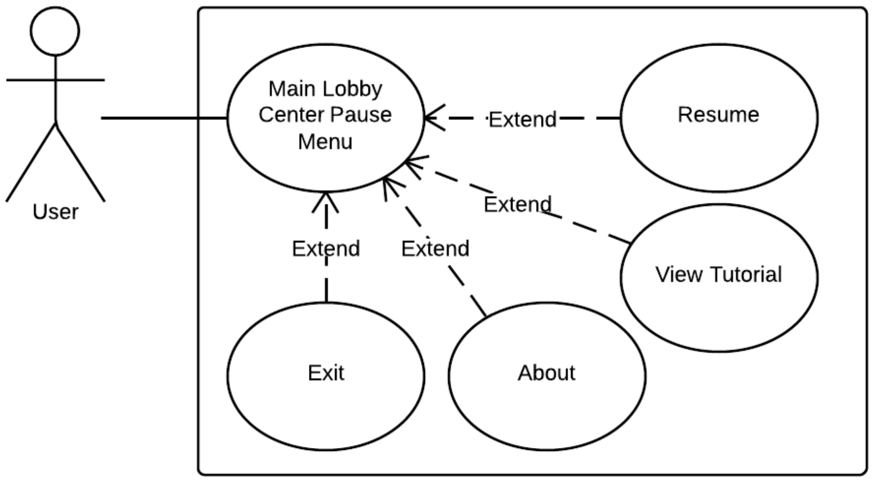
Lobby Center Tutorial (UC15)



UC16. Pause Exploring of Lobby Center – When the Pause Menu is brought up, the user is no longer able to control the player. Four options are displayed while the user is within the Lobby Center. These options are “Resume”, “View Tutorial”, “About”, and “Exit”. If the “Resume” option is selected, the pause menu is removed from the display

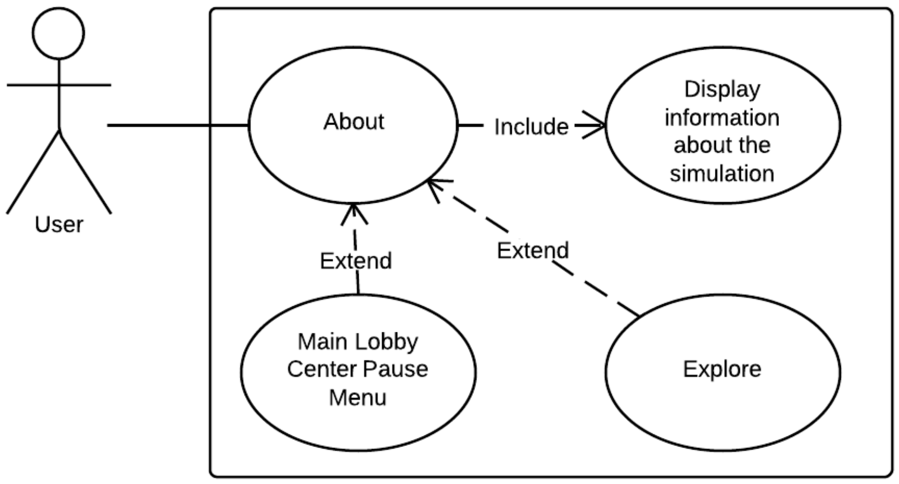
and the user is then able to explore and interact as before. If the "View Tutorial" option is selected, the tutorial for the Lobby Center is displayed. If the "About" option is selected, information about the simulation is displayed. If the "Exit" option is selected, the simulation will terminate. See Use Case Diagram 16 below.

Pause Exploring of Lobby Center (UC16)



UC17. About the Simulated Exhibit Experience – When the user clicks on the "About" option on the Lobby Center Main Menu, information about the simulation is displayed. The user is given two options: "Explore" and "Main Menu". If the "Explore" option is selected, the pause menu is removed from the display and the user is then able to explore and interact as before. If the "Main Menu" option is selected, the Lobby Center Main Menu is displayed. See Use Case Diagram 17 below.

About the Simulated Exhibit Experience (UC17)


User
About
Include
Display information about the simulation
Extend
Extend
Main Lobby Center Pause Menu
Explore

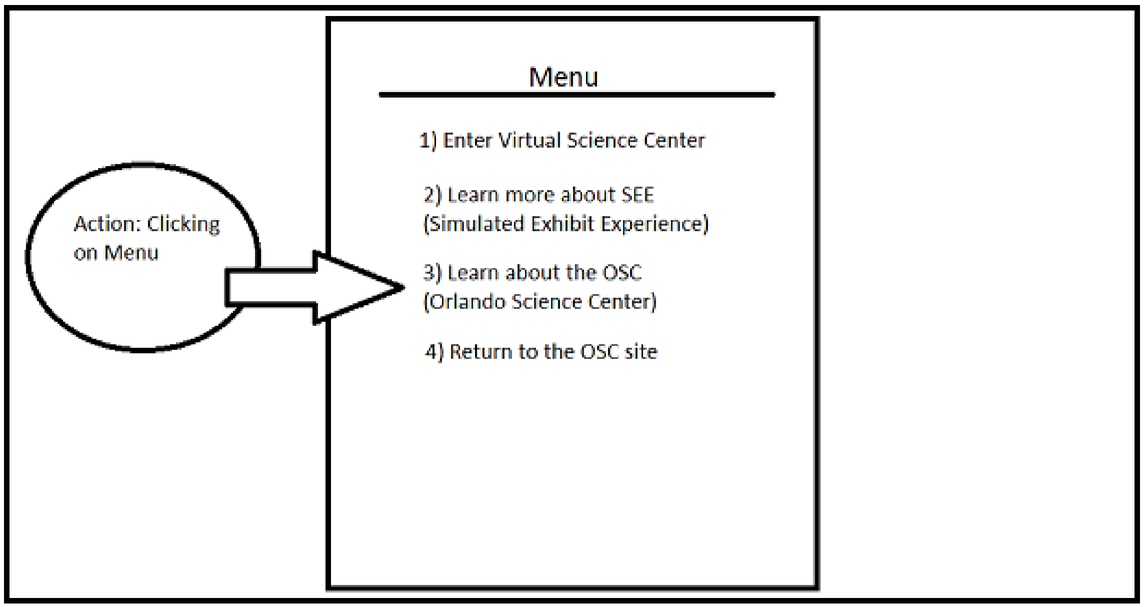
3.3.2 Prototype Storyboards

The beginning layout of the Virtual Science Center will be comprised of three sections: a Menu, the Bridge area and the Main Lobby. The Bridge and the Virtual Science Center Lobby are individual rooms but are located in the same level (meaning that contents are loaded at the same time). Each room will have optional aesthetic features to make it more appealing, but right now the designs are drawn to be simple while still portraying the main importances of each room functionalities.

The first thing the user will see once they start the Unity Web Player on the OSC website is a Menu (Figure 3.0) that displays the following options:

- Enter Virtual Science Center - will allow the user access to the virtual exhibits and start the content loading.
- Learn more about SEE (Simulated Exhibit Experience) - inform the user about the specialized team that worked on the OSC Virtual Exhibits Hall.
- Learn about the OSC (Orlando Science Center) - taken from the OSC site, display information regarding the OSC.
- Return to the OSC site - allows the user to exit out of the Unity Web Player by deactivating it.

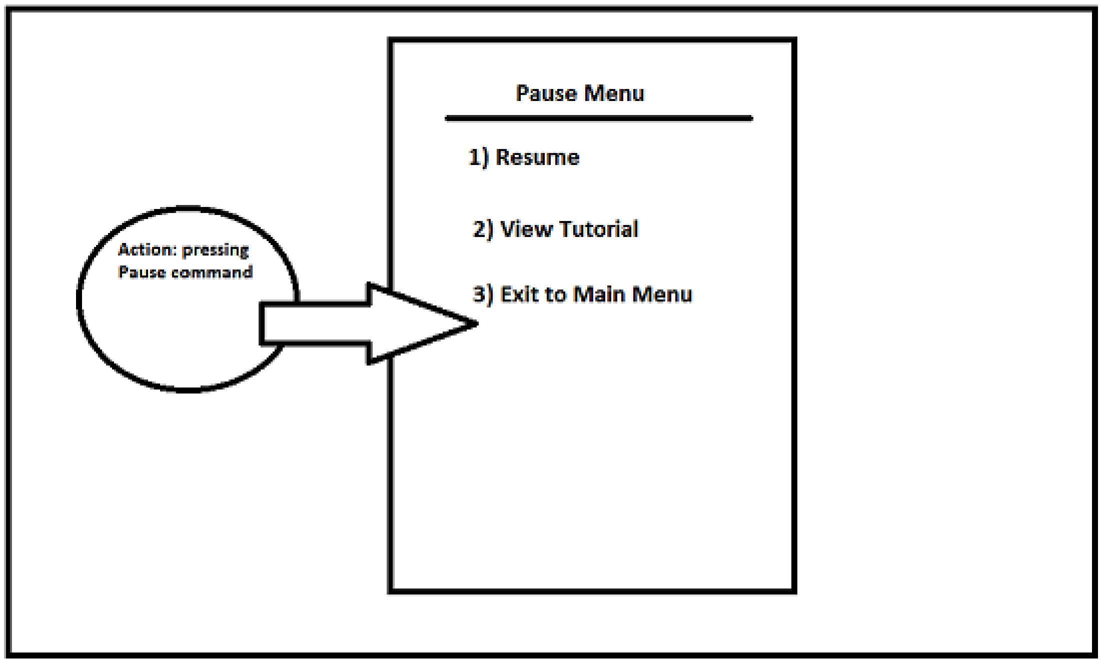
Figure 3.0


Menu
1) Enter Virtual Science Center
2) Learn more about SEE (Simulated Exhibit Experience)
3) Learn about the OSC (Orlando Science Center)
4) Return to the OSC site
Action: Clicking on Menu

The user is able to pause the virtual simulation at anytime during run time by pressing a Pause command to trigger the Pause Menu (Figure 3.1).The Pause Menu will give the user three options to choice once:

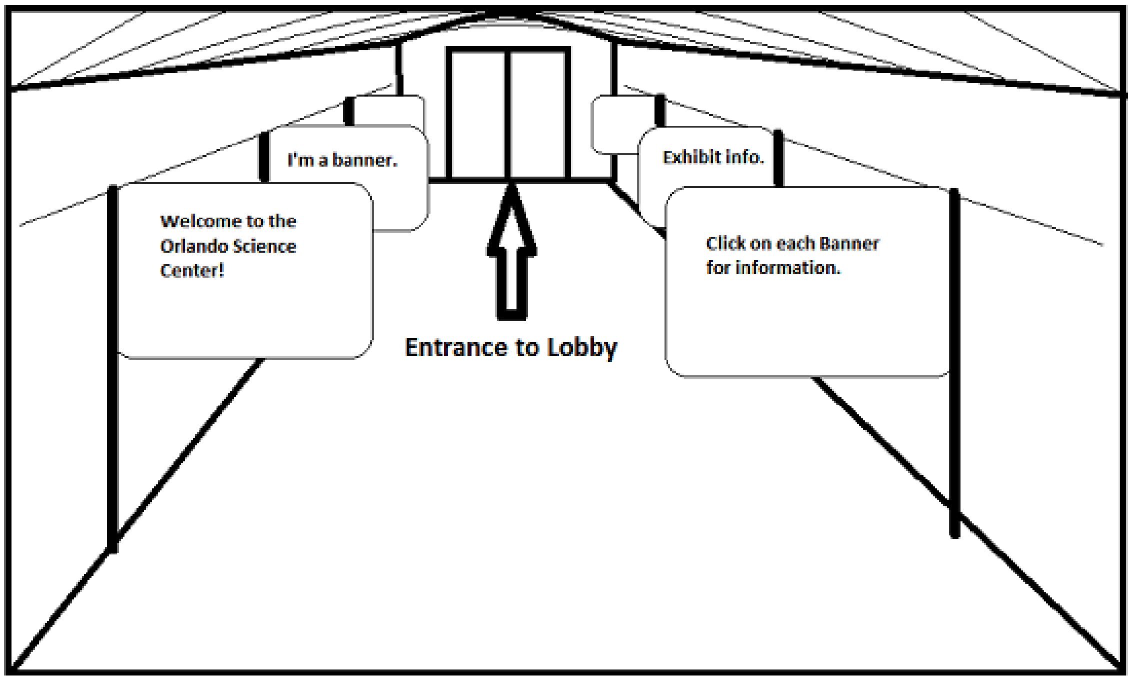
- Resume - the system will exit out of the Pause Menu and resume back to the virtual simulation.
- View Tutorial - the user will be able to view Tutorials when selecting this option, which will trigger a new pop-up window.
- Exit to Main Menu - the system will revert back to the Main Menu option.

Figure 3.1



The Bridge section (Figure 3.2) will contain Banner objects and a Door object (which leads the user to the Virtual Science Center Lobby). The Banners are interactive objects, allowing the user to be able to click on them and the system will initiate a pop up window that will display information relevant to the Banner that was clicked. Information will correspond to the Orlando Science Center exhibits as well as “Fun-Fact” type of Banners. This section will behave like an introduction to the OSC Virtual Exhibits Hall. The user is encouraged to explore each interactive objects for educational purposes as well as getting adjusted to the movements control of the virtual simulation.

Figure 3.2


I'm a banner.
Exhibit info.
Welcome to the Orlando Science Center!
Click on each Banner for information.
Entrance to Lobby

In the picture (Figure 3.3), the panel on the left is a simple Banner object with a topic related to the Orlando Science Center. Once the user clicks on it by hovering the mouse cursor over it, the system will initiate the Banner event which triggers a pop-up window (right panel) that will display the corresponding information. To exit out of the window, the user can press the ESCAPE key on the keyboard to get out of the Banner event.

Figure 3.3

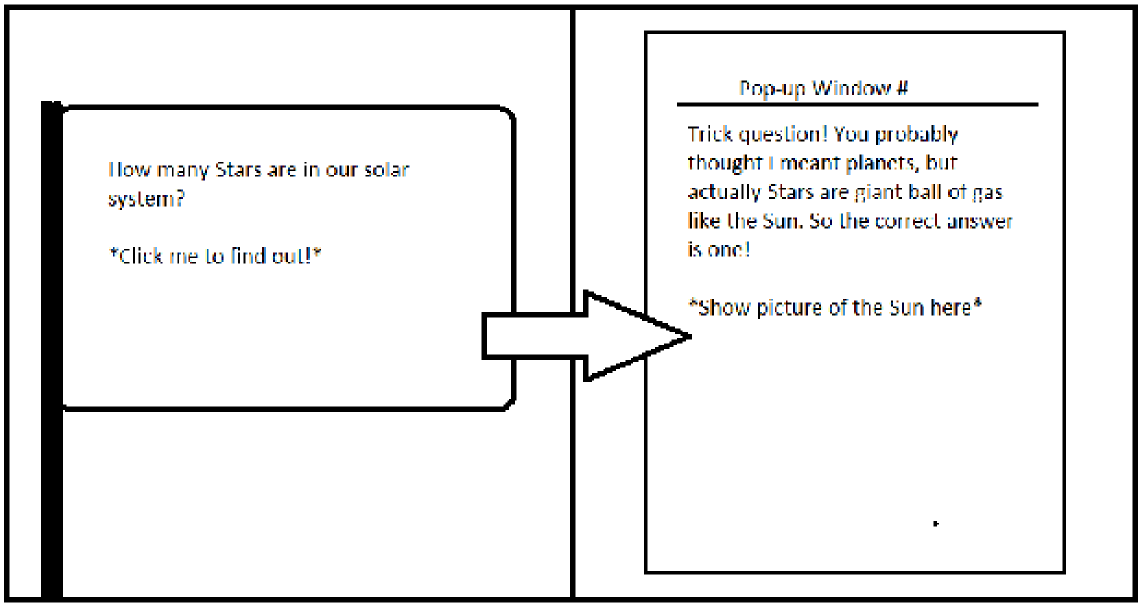
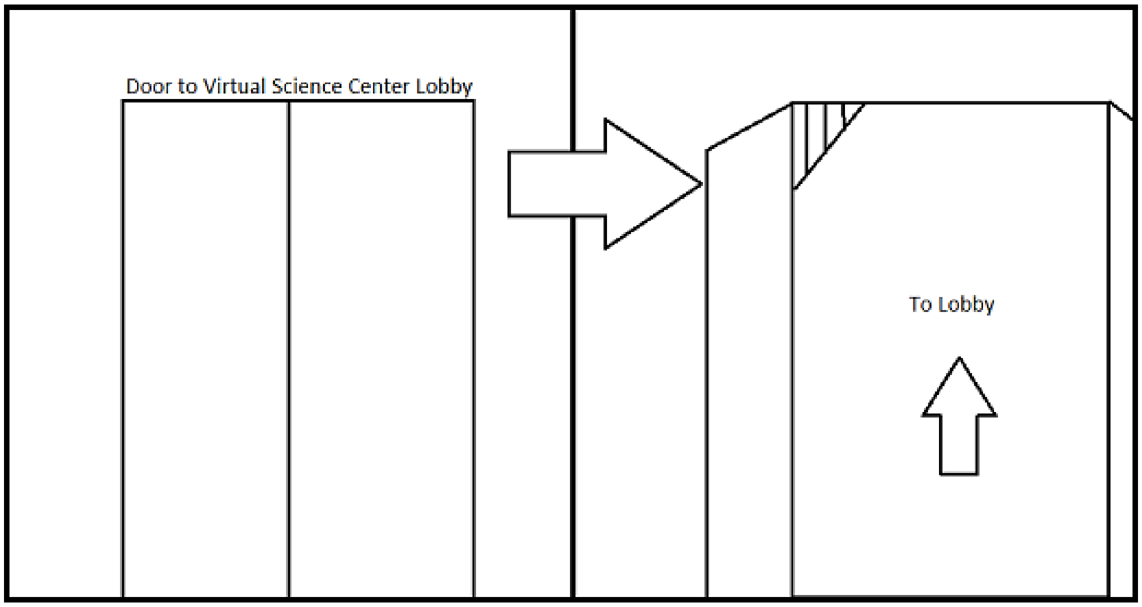

How many Stars are in our solar system?
Click me to find out!
Pop-up Window #
Trick question! You probably thought I meant planets, but actually Stars are giant ball of gas like the Sun. So the correct answer is one!
Show picture of the Sun here

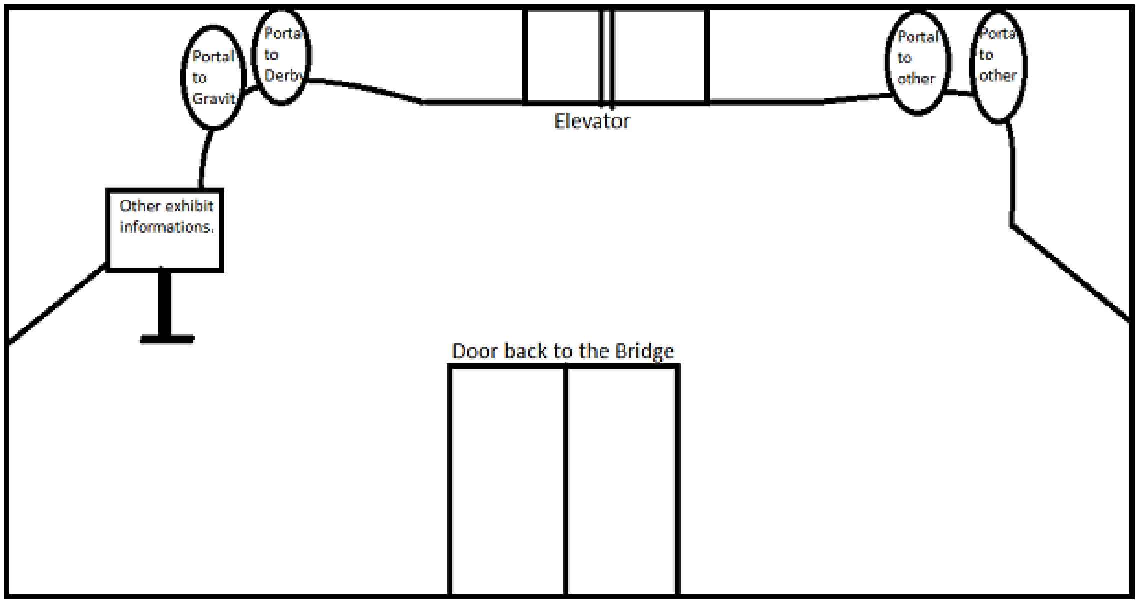
Figure 3.4 shows the Door object which leads the user into the Virtual Science Center Lobby. When the user comes in close to the Door, the system will automatically trigger an event to open the Door, allowing access to the Lobby section.

Figure 3.4



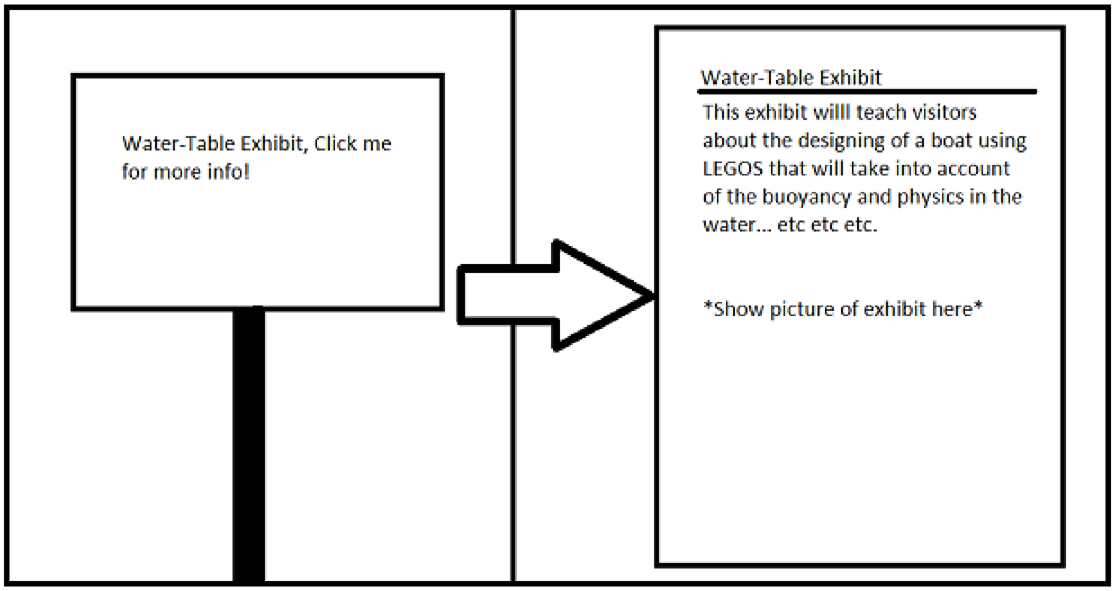
The Virtual Science Center Lobby (Figure 3.5) will contain some aesthetic features of the actual Orlando Science Center, as well as interactive portals that will provide access to each virtual exhibits in the system. There are also some informational objects that will act the same way as the Banner objects in Figure 3.3, in which the user is able to click on the objects. The shape of the objects will be a Sign Post design that will display the name of exhibits that are not yet implemented into the system.

Figure 3.5



When the user clicks on the Sign Post object (Figure 3.6), the system will trigger a pop-up window that will properly display information about the exhibits corresponding to the Sign Post displayed text. To exit out of the window, the user will follow the same key command (escape key).

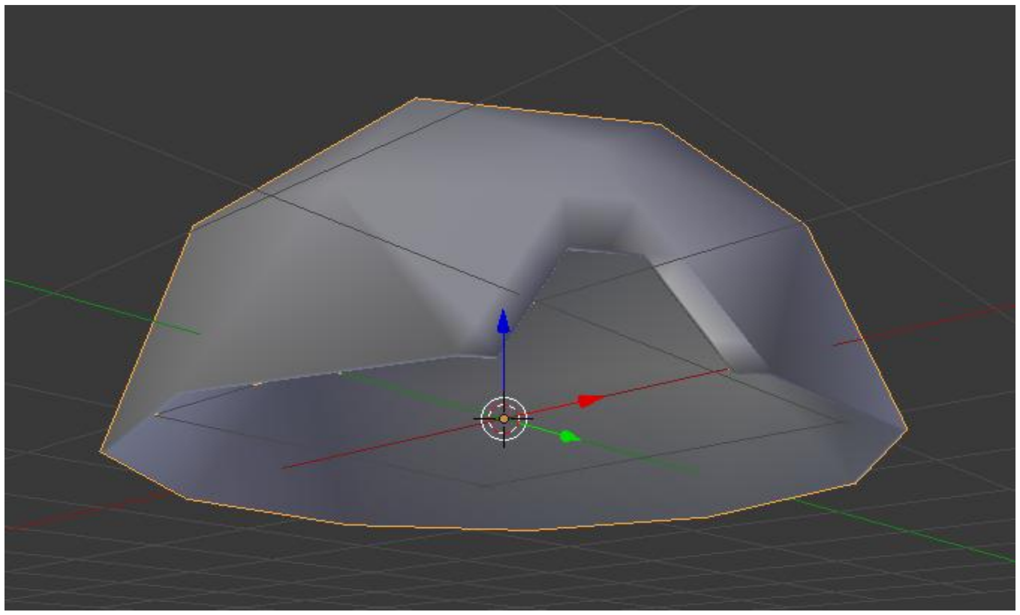
Figure 3.6

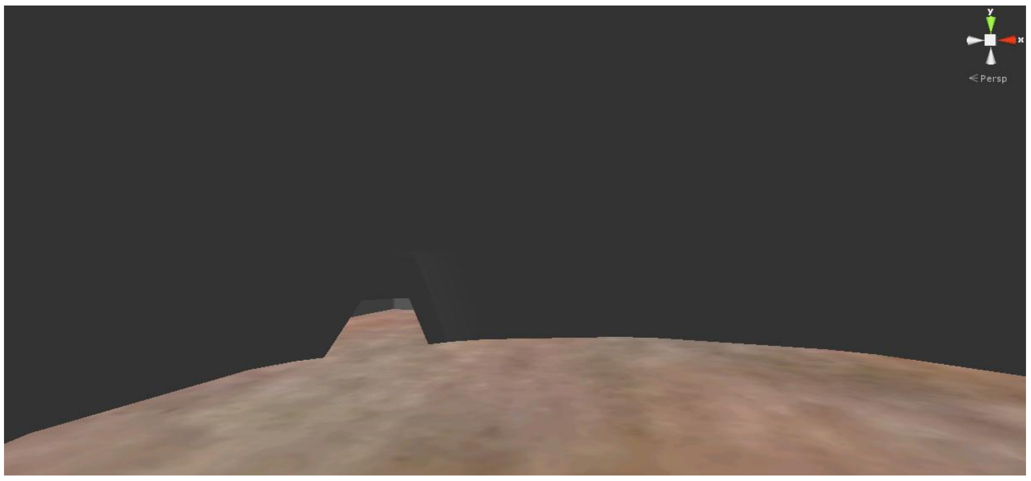

Water-Table Exhibit, Click me for more info!
Water-Table Exhibit
This exhibit willl teach visitors about the designing of a boat using LEGOS that will take into account of the buoyancy and physics in the water... etc etc etc.
Show picture of exhibit here

3.3.3 Room Layout

The first prototype of the Lobby Center:

The dome-shaped idea was a simple plan for testing purposes. However, the edges of the dome were not sharp enough. This creates a smooth cap-shape rather than the ideal model that we desired. Also, the lighting effect inside the dome did not create an illusion of open space, rather it felt too enclosed, claustrophobic, and the distance of the user to other objects in the room was confusing.

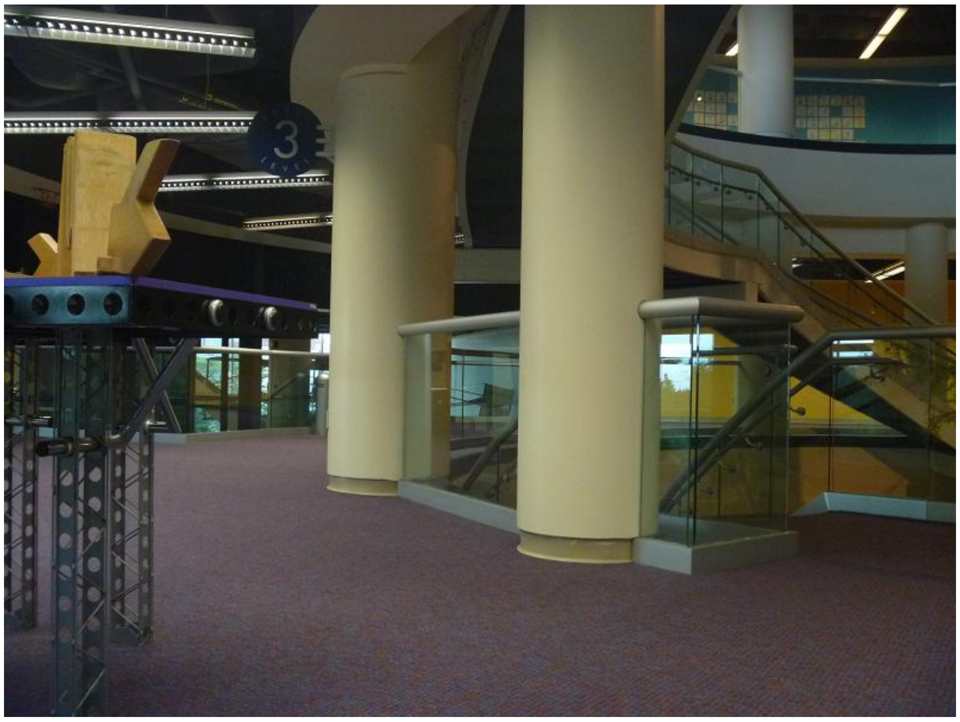


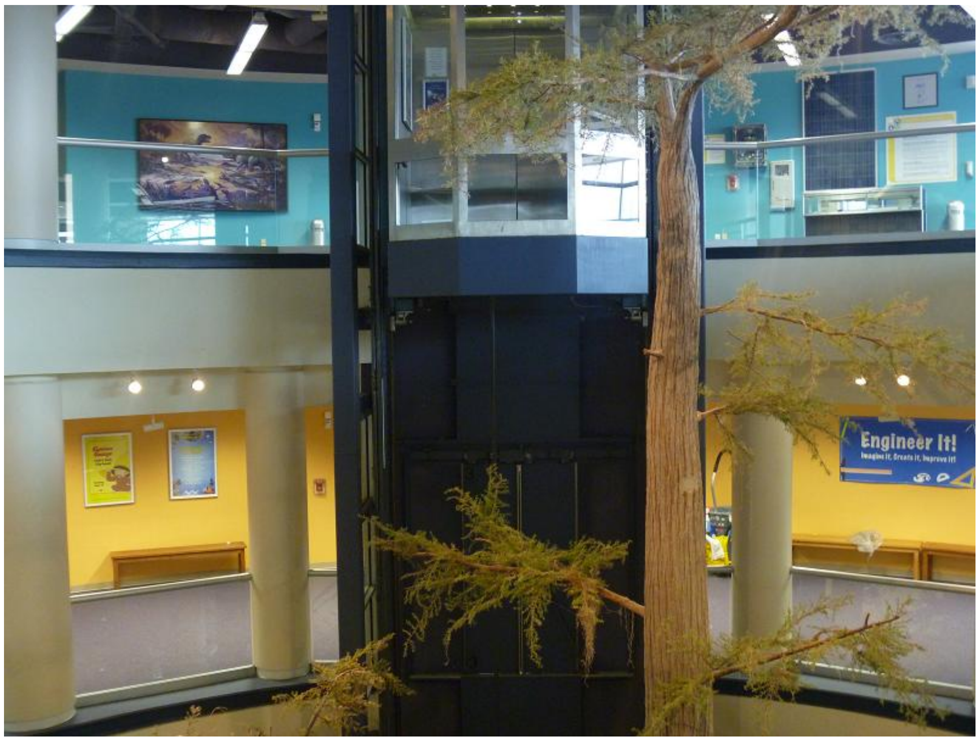


Users should feel welcomed when they first appear in the simulation, thus we scrapped the dome-shaped Lobby Center implementation and came up with a layout that is similar to the one in the OSC. With the current scope of the project, we will have at least one functional exhibit but still provide the illusion of multiple exhibits with doors that cannot be opened. These extra doors will make it easier for future developers to add more exhibits with our existing Lobby Center rather than creating a new building.

The second prototype for the Lobby Center:

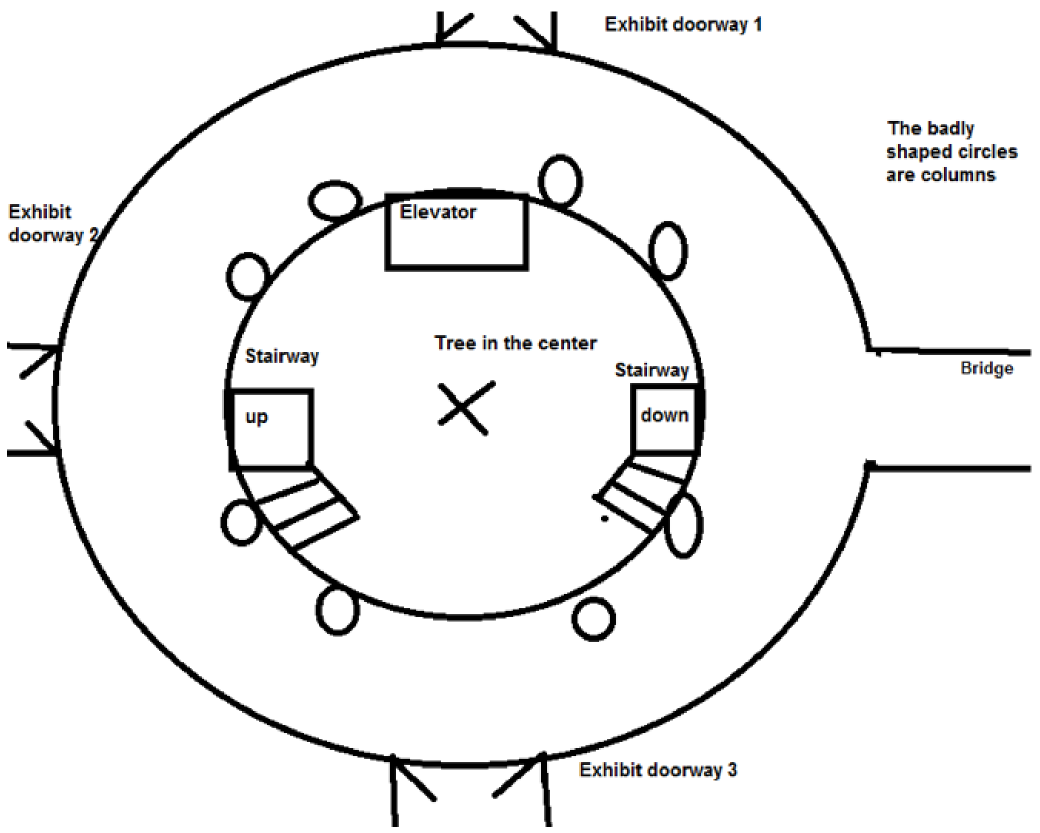
The next model of the Lobby Center that we created was a simplified version of the OSC, since there were parts that we could not make due to our limited 3D modeling experience. Using the pictures taken from the OSC tour, we picked out components that we thought were important to the overall structure and feel of the Lobby Center.





List of components desired:

- Walls
- Floor
- Ceiling
- Columns
- Stairs
- Elevator
- Doors
- Chairs
- Tables
- Lighting
- A tree in the center



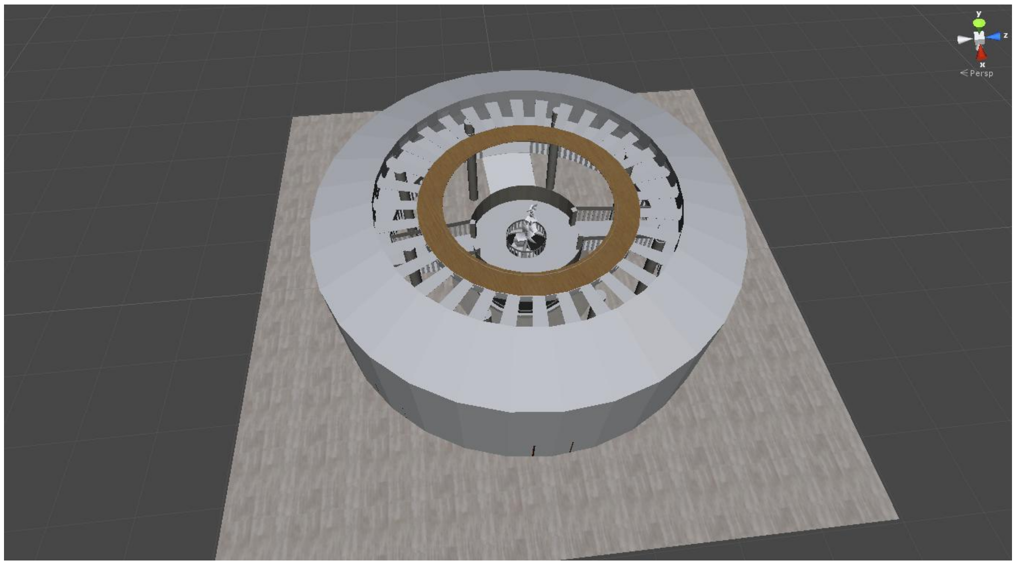
The blueprint for the desired Lobby Center was sent to Presagis while we developed a placeholder in the meantime. Due to Presagis's own deadlines, we weren't certain if they would have time to spare creating models. The Lobby Center placeholder is based on the idea of a circular room with stairs and doors leading to the other exhibits. This placeholder model was the next iteration in the phased prototyping process and again was developed with input from all group members.

First Floor Entrance View - As shown in the picture below, there are multiple doors on the outside wall of the Lobby Center. The two doors on the right, with signs above them, lead to the two exhibits we have developed.

Pinewood
Gravitro

Second Floor Ceiling View - By utilizing the skybox rendering of a night sky, the simulation feels more open and expansive. The user is able to get a better view on the second floor where there is a platform to stand on.

Aerial View - Even though the user is unable to see from this perspective, this is to show the overall layout of the Lobby Center placeholder design.



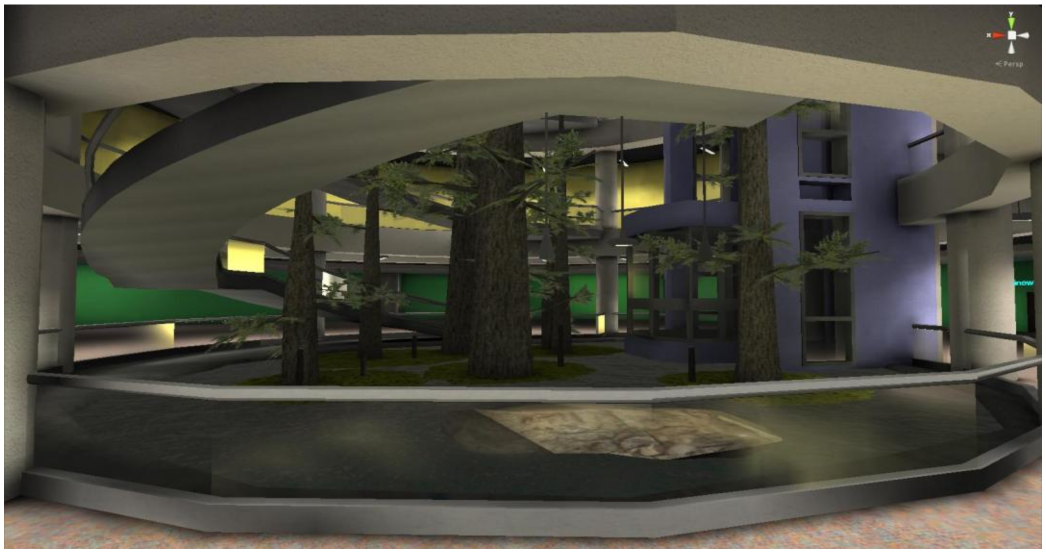
The second prototype of the Lobby Center allowed us to see how the user will navigate to the exhibits. The circular room model is definitely more visually appealing and provides the open space environment that we desired.

The third prototype of the Lobby Center:

The current model was created by Chris Culverwell, an artist at Presagis. The Lobby Center that he made is as close to the actual OSC as we imagined. Everything from the overall layout to the fine details are just stunning. He also included the iconic OSC Bridge for us. The bridge was a model that we really wanted to have but it was too difficult for us to create reflective surfaces and an outside environment with the limited time left in the project.

Second floor view of the Lobby Center

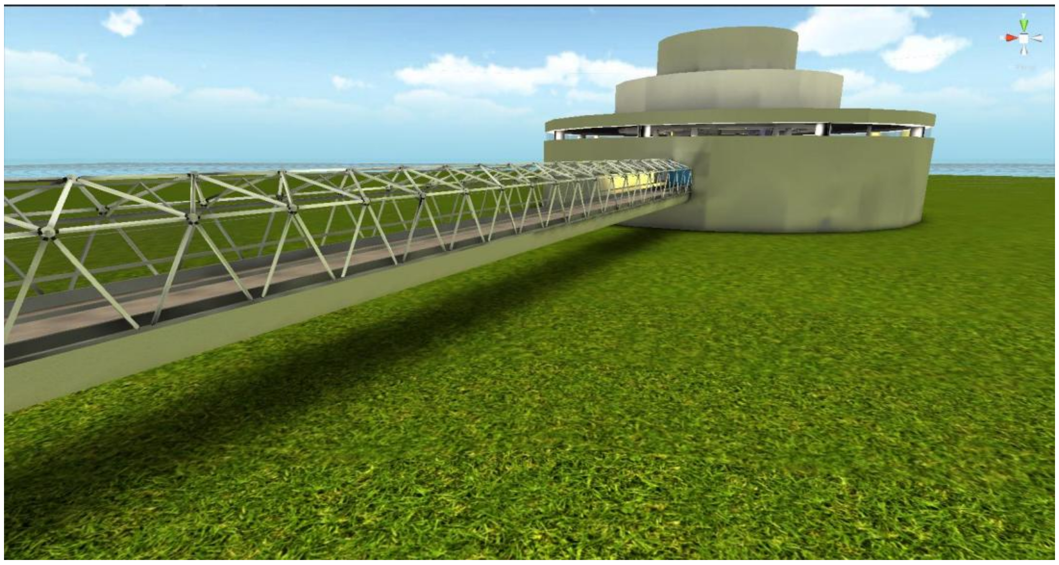
First floor view of the Lobby Center



Door view of the Lobby Center

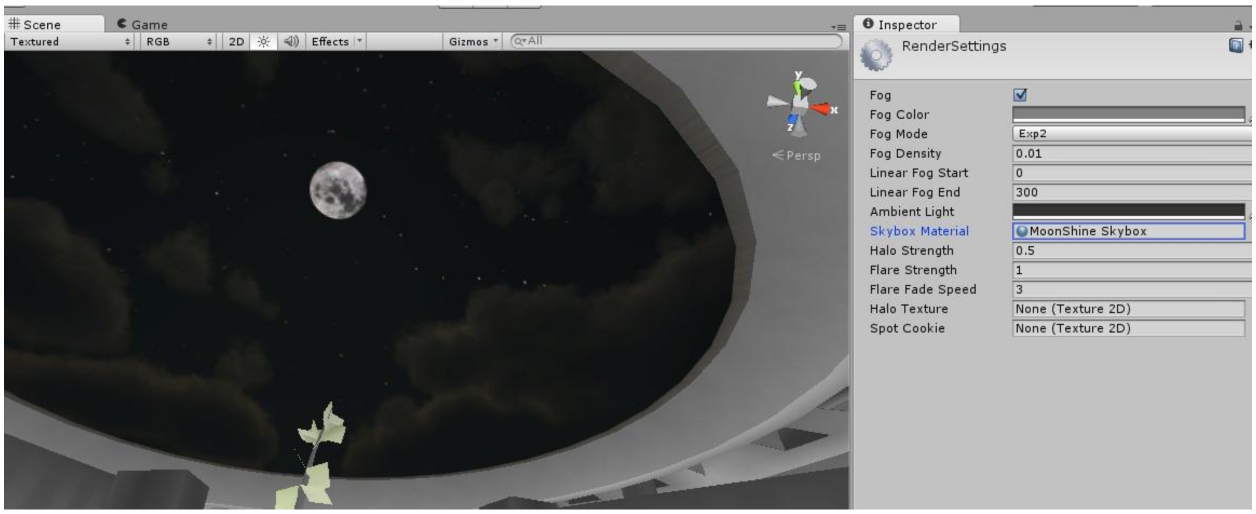
Gravitron
Pinewood Derby

Outside view of the OSC Bridge



3.3.4 Sky box

Unity Skyboxes are essentially a wrapper of images around a whole scene and are rendered first before any other objects in that scene. Since the Orlando Science Center buildings are mostly windows, we want the user to be able to see what's outside the building but unable to venture out there. Unity's Skybox object gave us the opportunity to create that desired illusion without having to make our own. There are several options of skyboxes that Unity provides for us but the one that we utilized can be seen in the previous pictures of the Lobby Center models and in the image below.
Skybox from second prototype selected in Unity

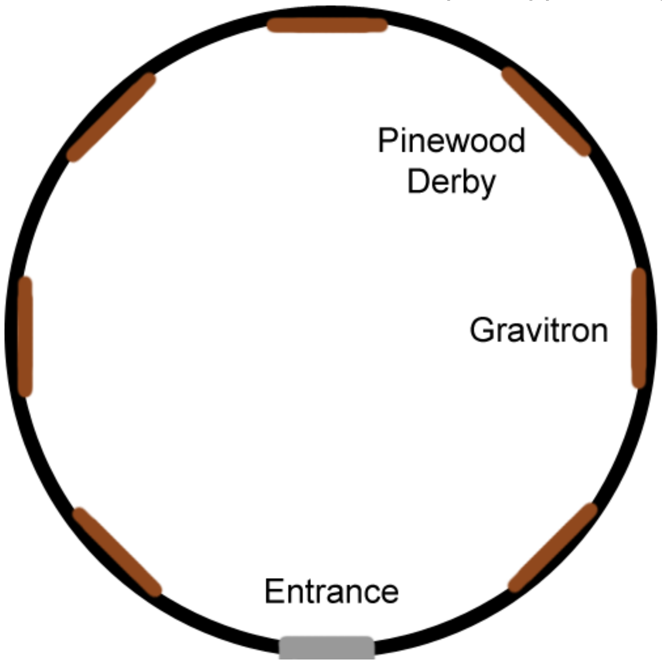


Fog	✓
Fog Color	▬
Fog Mode	Exp2
Fog Density	0.01
Linear Fog Start	0
Linear Fog End	300
Ambient Light	▬
Skybox Material	MoonShine Skybox
Halo Strength	0.5
Flare Strength	1
Flare Fade Speed	3
Halo Texture	None (Texture 2D)
Spot Cookie	None (Texture 2D)

3.3.5 Doors

Doors in the Lobby Center represent the portal entries to the other exhibits. The prototype model of the Door object that we have is a double door as requested by the OSC sponsors.

Diagram of the doors located on the first floor of the prototype Lobby Center.



There are seven doors on the first floor of the Lobby Center. Only two of the doors lead to active exhibits. From the entrance or starting location, the second door to the right leads to the Gravitron Exhibit. The door to the left of the Gravitron door, leads to the Pinewood Derby Exhibit.

Only two doors exist in the final prototype of the Lobby Center. These doors are on the first floor just past the stairs.

3.3.6 Lobby Pause Menu

The Lobby Center finite state machine has four main states that can be transitioned to and an additional state for each exhibit that can be reached from the Lobby Center.

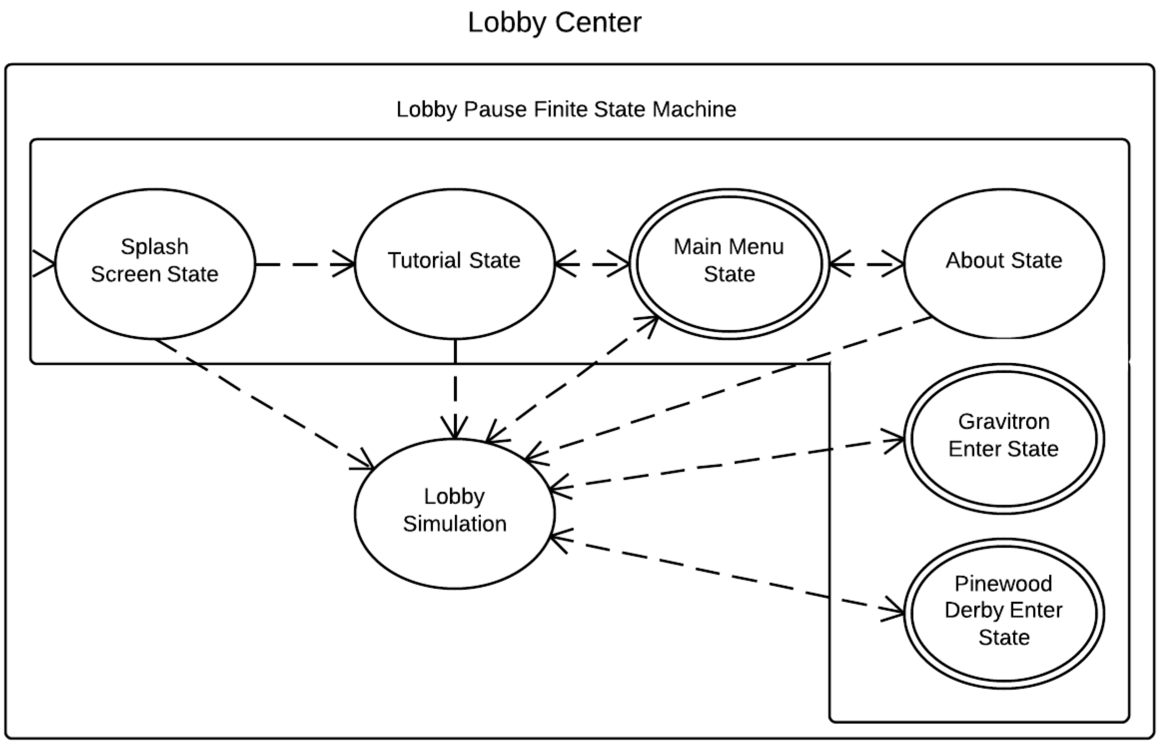
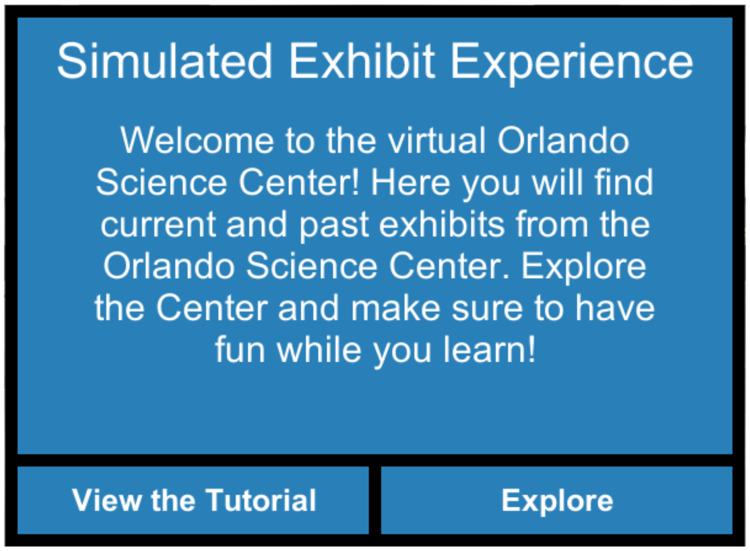


Diagram of the Lobby Center GUI Finite State Machine.

Splash Screen

The Splash Screen State is the state that is displayed when the Lobby Pause Menu is first loaded. This is the only way to reach this state. The Splash Screen State displays a welcome message to the user and gives the user two options: “View the Tutorial” and “Explore”. When the “View the Tutorial” option is selected, the finite state machine transitions to the Tutorial State. When the "Explore" option is selected, the simulation resumes.

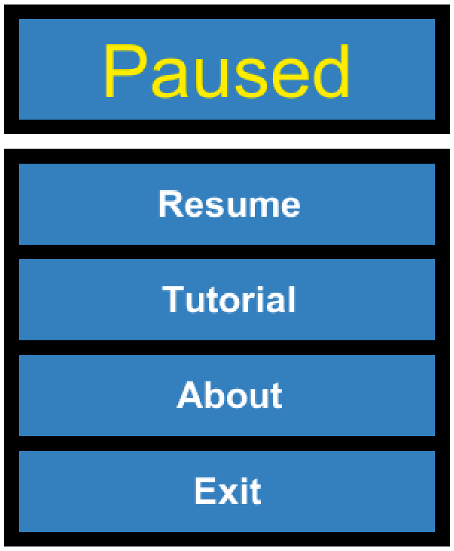


Simulated Exhibit Experience

Welcome to the virtual Orlando Science Center! Here you will find current and past exhibits from the Orlando Science Center. Explore the Center and make sure to have fun while you learn!

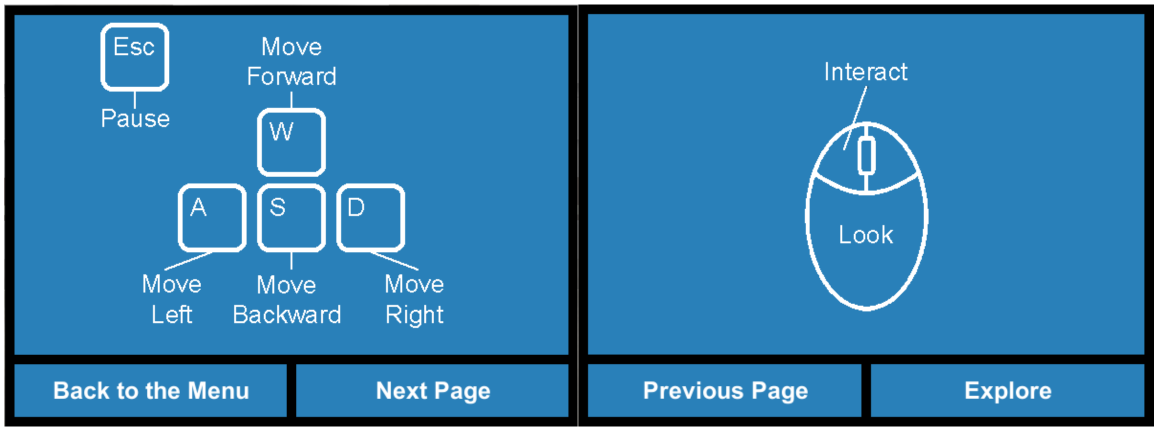
Main Menu

The Main Menu State is the state the default state for the Lobby Pause Menu and is loaded whenever the simulation is paused. There are three ways to enter the Main Menu State: transition from the Tutorial State, transition from the About State, or by pressing the escape key during the simulation. The Main Menu State gives a list of options to the user. This list includes: "Resume", "Tutorial", "About", and "Exit". When the "Resume" option is selected, the simulation resumes. When the "Tutorial" option is selected, the finite state machine transitions to the Tutorial State. When the "About" option is selected, the finite state machine transitions to the About State.



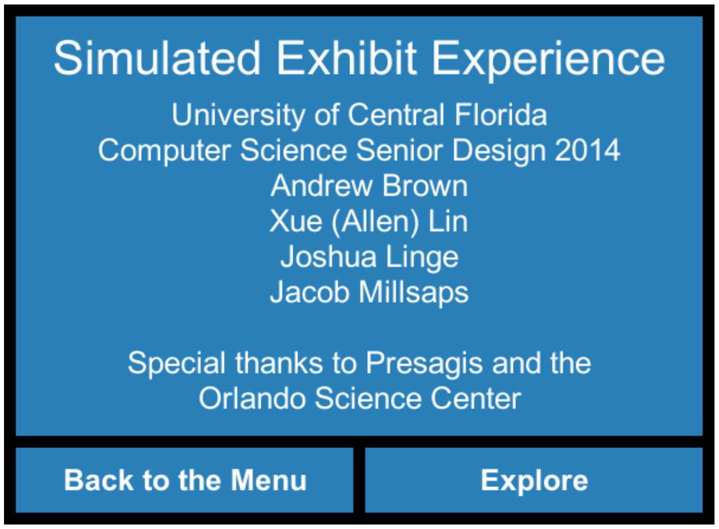
Tutorial

The Tutorial State is a state that displays a list of images that help describe how to move around in the Lobby Center. The Lobby Tutorial has two images to inform the user about the controls. The first image lists the keys used in the Lobby Center, which include the “W”, “A”, “S”, and “D” keys for movement and the escape key for pausing the simulation. The second image displays how the mouse interacts in the Lobby. The user can move the mouse to look around and the user can also click the left mouse button to interact within the room. There are two ways to enter the Tutorial State: transition from the Main Menu State or transition from the Splash Screen State. For each Tutorial page two options are given to the user. These options include “Previous Page” and “Next Page”. If the tutorial page is the first page, the “Previous Page” option is changed with the “Back to the Menu” Option. If the tutorial page is the first page, the “Next Page” option is changed with the “Explore” Option. When the “Back to the Menu” option is selected, the finite state machine transitions to the Main Menu State. When the “Explore” option is selected, the simulation resumes. When the “Next Page” option is selected, the next tutorial image is displayed. When the “Previous Page” option is selected, the previous tutorial image is displayed.



About

The About State is a state that displays information about the senior design team that created the Simulated Exhibit Experience application. The only way to enter the About State is by transitioning from the Main Menu State. The About State gives the user two options: “Back to the Menu” and “Explore”. When the “Back to the Menu” option is selected, the finite state machine transitions to the Main Menu State. When the “Explore” option is selected, the simulation resumes.

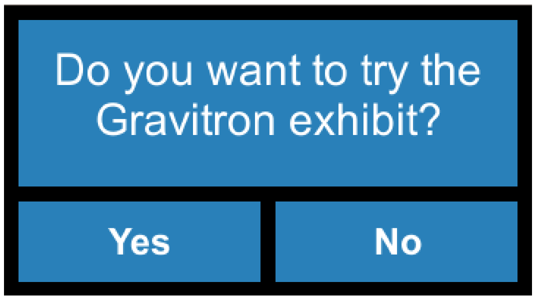

Simulated Exhibit Experience

University of Central Florida
Computer Science Senior Design 2014
Andrew Brown
Xue (Allen) Lin
Joshua Linge
Jacob Millsaps

Special thanks to Presagis and the
Orlando Science Center

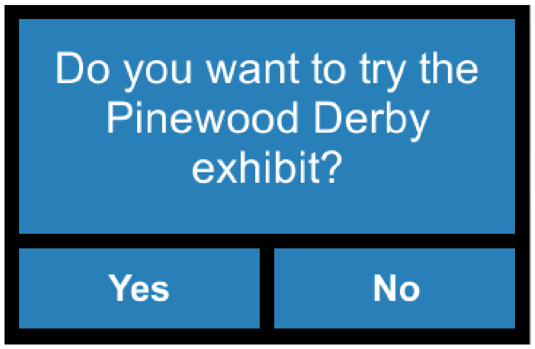
Gravitron Enter State

The Gravitron Enter State is the state that asks the user if they would like to enter the Gravitron Exhibit. There are two ways to enter the Gravitron Enter State: clicking on the door set to the Gravitron Exhibit or by running into the door set by the Gravitron Exhibit. The Gravitron Enter State gives the user two options: “Yes” to entering the Gravitron Exhibit, and “No” to not enter the Gravitron Exhibit. When the “Yes” option is selected, the user is then transported to the Gravitron Exhibit. When the “No” option is selected, the simulation resumes.


Do you want to try the
Gravitron exhibit?

Pinewood Derby Enter State

The Pinewood Derby Enter State is the state that asks the user if they would like to enter the Pinewood Derby Exhibit. There are two ways to enter the Pinewood Derby Enter State: clicking on the door set to the Pinewood Derby Exhibit or by running into the door set by the Pinewood Derby Exhibit. The Pinewood Derby Enter State gives the user two options: “Yes” to entering the Pinewood Derby Exhibit, and “No” to not enter the Pinewood Derby Exhibit. When the “Yes” option is selected, the user is then transported to the Pinewood Derby Exhibit. When the “No” option is selected, the simulation resumes.



3.4 Final Assessment

For the final product we presented to our Senior Design committee, our Lobby Center fully satisfied all functional and non-functional requirements.

Before receiving the artwork from Presagis, our second prototype looked somewhat similar to the Orlando Science Center, but still wasn’t as close as we had hoped. This was certainly no fault of Xue, who modeled the entire second prototype Lobby Center except for the doors, but just the reality that we are not professional 3D modelers. Thankfully, Presagis was able to provide us with the professional models which helped us better complete this requirement and impress users of the simulation.

4. Gravitron Exhibit

4.1 Introduction

The Gravitron exhibit in the Orlando Science Center provides an opportunity to experience the engineering design cycle in a fun and interactive way. The engineering design cycle starts with observing a problem, designing a solution to the problem, implementing the solution, and then testing the solution. The Gravitron exhibit consists of a Magnetic Wall, a variety of differently shaped Tubes, a Goal Cup, and a Ball. The main objective of this exhibit is to have the Ball at the top roll through a course of assembled Tubes into the Goal Cup at the bottom of the Magnetic Wall. The user has full control on where to place the Tubes on the Wall. They can slide the Tubes anywhere on the Wall and rotate each piece by 360 degrees. Before any Tubes are placed on the Wall, the user should think critically of the best route for the Ball to traverse along the Wall to the Goal Cup. Once they have a design in mind, they can start implementing the layout by placing the Tubes accordingly. After assembling the Tubes based on their design, the user then tests their layout by dropping the Ball into the first Tube at the top of the Wall. If they are satisfied with the results, they can move to another exhibit or go back through the cycle to refine their design and test again.

4.2 Gravitron High Level Requirements

4.2.1 Introduction and Goals

The Gravitron Virtual Exhibit simulation has replicated the main functions of the real life Gravitron exhibit.

The Proposed System: Expected Impacts
The purpose of the Gravitron Virtual Exhibit is to replicate the main idea of the physical version of the exhibit in the Orlando Science Center, along with some additional features that are not available in real life. The expected impact is that users will be able to learn and apply the same engineering design cycle principles as they would in real life but with more flexibility. At the physical Gravitron Exhibit, a visitor may have their design ruined by a fellow visitor or feel pressured to give those waiting a chance to use the exhibit. Not to mention that the physical OSC is constantly removing old exhibits and

replacing them with new ones. Thus visitors to the Simulated Exhibit Experience are welcome to freely experience the virtual version at any time online. Another reason is that kids using the physical exhibit are limited by their height in what they can build unless they have someone taller helping them. With the virtual version, users of any height or physical ability can be free to build whatever they can imagine as long as they can use a mouse and keyboard. At the same time these individuals are learning the valuable principles of the engineering design cycle that they may not have been able to experience before.

In order for the Gravitron Virtual Exhibit room to be considered "good enough to deliver" it must contain the core functionality of allowing users to assemble a layout of tubes on a wall, simulate a ball rolling through the tube layout, allowing users to modify their tube layout, and finally the system should provide some sort of assessment of their layout performance. There are some optional features that would provide a more exciting and attractive experience but those will be included as time and space permits.

4.2.2 Rules of the Gravitron Exhibit

There are different challenge modes to the Gravitron Exhibit at the Orlando Science Center. The challenges are typically set for a group of people (usually more than one person). However, the virtual version will allow the user to experiment with each challenge mode by themselves with the help of a Grading Score Generator (relevant to the High Scores Table). The virtual version also has a lot more options than the real version which allows for a greater variety of challenges. The challenges can be selected through the Challenge Mode menu accessed via the Challenge Mode button. Each challenge comes with their own set of rules that the user must follow. The set of rules for each challenge mode will be explained in this section.

Grading Score Generator
Depending on which challenge mode is selected, the Grading Score Generator will generate a score from A to F based on the Timer object The score will be displayed during run time so that the user can see their current progress and grade without having to wait at the end (whether the Ball drops into the Goal Cup or the Gutter). The displayed score will change once the Timer hits a certain value in which the grade either goes up or down depending on the challenge mode.

Challenge Modes
As was mentioned, there will be multiple challenge modes to keep things exciting for the user rather than having just one objective of getting the Ball object into the Goal Cup.

Each challenge mode will also change some aspects of the Gravitron Virtual Exhibit. The modes will be explained below with their purposes and restrictions:

Mode 1 (Fastest-Time Objective)

Purpose - *The user will be challenged to get the Ball object into the Goal Cup in the fastest time possible, as indicated by the Timer.*

Restriction(s) - *The user must use at least X number of Tubes in their design, X representing the minimum amount. This is to prevent the user from possibly abusing the Ball object to let it just fall straight down to the Goal Cup. The start point of the ball must also not be directly above the Goal Cup.*

Mode 2 (Slowest-Time Objective)

Purpose - *The user will be challenged to get the Ball object into the Goal Cup in the slowest time possible, as indicated by the Timer.*

Restriction(s) - *The user must use at most X number of Tubes in their design, X representing the maximum amount. This restriction is similar to Mode 1, except that it prevents the user from implementing too many Tubes in which the design may clog up the space on the Magnetic Wall (can cause multiple glitches. See section 4.3.8 Operational Scenarios (Tube Objects)). The start point of the ball must also not be directly above the Goal Cup.*

Mode 3 (Timed Objective)

Purpose - *When this mode is selected, the Timer will start as soon as the user place the first Tube onto the Magnetic Wall. The Timer will continue to start counting until the user has designed a layout and ran the implementation and has the Ball object hit the Goal Cup. The user will be tested on setting up the Tubes as fast as they can and have the Ball hit the objective. However, the Timer will still continue to count if the user's Tube design fails, thus the user must be quick on their feet and resume correcting the layouts.*

Restriction(s) - *The user can't restart the Timer once they place the first Tube on the Magnetic Wall.*

4.2.3 User Roles and Operations

There will be only one user role for our system, the Player role. The Player can walk around in the environment, enter and exit exhibits, view information about the exhibits, and interact with the exhibits.

4.2.4 High Level Requirements and Features

1. Functional:

1. The simulation shall contain major components of the real version of the Gravitron Exhibit.
2. The simulation shall allow the user to select different challenge modes.
3. The Goal Cup object shall be the main objective of the exhibit and return as a successful attempt.
4. The simulation shall display a prompt to the user if the Ball object enters the Goal Cup object and then remove the Ball from the room.
5. The Gutter object shall interact with the Ball object and return as a failed attempt.
6. The Floor object shall act as a safety-net in case the Ball misses the Gutter; will interact with the Ball object and return as a failed attempt.
7. The simulation shall display a prompt to the user if it detects the Ball object is stuck (if it doesn't move for 2 seconds).
8. The simulation shall display a prompt to the user if the Ball lands in the Goal Cup that notifies them of their completion time.
9. The simulation shall allow only one Ball to be present in the simulation at all times.
10. The program shall not allow the Ball to move through the Walls of the Tubes.
11. The simulated Ball shall move in a similar manner when compared to the version in real life.
12. The simulation shall allow the user to reset their design at any time with the reset button.
13. The simulation shall allow the user to exit the Gravitron virtual exhibit at any time with an interactive exit Door.
14. The simulation shall make sure only one camera is active at a time to prevent objects that are not meant to collide from colliding. Functional varieties of Tubes (pipes):
    - Each Tube shall be able to be cloned, the number of clones a Tube object can have depends on the set of rules the Gravitron Virtual Exhibit will provide.
    - Each individual Tube shall be able to follow the same response by the users' actions corresponding to the keys pressed on the keyboard. (Example: all Tubes must have the ability to be rotated and translated once selected).
15. The Ball object shall implement the physics of gravity and kinetic motion.
16. The Camera objects shall be accessible by different scripts such that proper user perspective will be displayed. (Example, if a script calls for Ball perspective, then the Camera objects will switch accordingly).
17. The Drop button shall trigger the activation of the Ball object into the implemented design layout of Tubes.

18. The Reset button shall reset the Ball or wrong implementations of the Tubes (pipes) (setting all Tubes back to their default positions).
19. Sounds shall be incorporated into the system.
20. The mouse cursor shall be fixed to the center of the screen
21. The user shall be able to use the mouse to move tubes across the Magnetic Wall

2. Non Functional:

1. The Gravitron Virtual Exhibit shall include set of rules such as different challenge modes for the user to interact and experiment with.
2. Each Tube shall have the same color that will change when the user hovers the mouse cursor over them (same as highlighting) and revert back to the original Tube color when the mouse cursor is not hovering over them.
3. The system shall perform collision checking to make sure the user or other solid objects do not pass through each other.
4. Lighting object shall be clear (not too bright and not too dim) to allow the user to visually see the objects in the Gravitron Virtual Exhibit.
5. The system shall play the correct sound effects when events are active (events such as the Ball rolling, the Ball colliding, the Door opening, etc…).
6. The system shall display textures on each object to provide a sense of realism as in the actual Gravitron Exhibit.

3. Desired Plans

1. Better textures to make the room more realistic
2. Better Tube object models
3. Higher quality sounds to increase realism
4. Higher quality of lighting
5. Smoother player movement control

4.2.5 Components of the Gravitron Virtual Exhibit

Functional Components

The functional components of the Gravitron Virtual Exhibit consist of the collection of different shapes of Tubes, a Ball, a Goal Cup, the Gutter, and the Magnetic Wall that keeps it all together. Each component will be under the category of objects.

Ball

To be used for the testing phase, in which player (or user) are able to test out their design layout of Tubes and observe how the Ball behaves.

Tube

There will be a variety of differently shaped Tubes for the exhibit. Each Tube will have the same diameter to ensure consistency for the Ball object.

Goal Cup

A simple holding object (a hollow cube with an opening on top), determines the goal for the exhibit.

Gutter

The Gutter will act as the failed attempt of the user's design layout since it is a good indication that the Ball did not go into the Goal Cup.

Magnetic Wall

A Wall that allows the Tubes to be manipulated on, There will be a start Tube at the top in which the Ball will fall through when player starts the testing process.

Timer

Will be located at the top left of the user's screen; main purpose is to keep track of the time it takes for the Ball to get to the Goal Cup, will stop once the Ball falls into the cup.

Drop Button

A button to be located near the bottom of the Magnetic Wall; once clicked, the Drop button will act behind the scene to start the testing phase.

Reset Button

A button to be located next to the Drop Button, will reset the room if the player wants to start over.

Challenge Modes Button

A button to be located next to the Reset Button, will allow the user to select different challenge modes for the Gravitron Virtual Exhibit.

Environmental objects

Objects such as the Walls, Ceiling and Floor connecting to form a room for the exhibit, as well as a Door for exiting the room.

High Scores Table

Will allow the player to see their corresponding score based on the Timer object.

Grading Score Generator

This generator is tied to the High Scores Table object and will generate a score based on Timer.

Non-functional Components

There are non-functional components that are not objects themselves but are tied to objects by attaching as components. These components are more of aesthetic purposes and can also serve as visually appealing to the user, else the virtual exhibits will look very bland.

Sound

In the Gravitron Virtual Exhibit, several sound effects will be implemented to further immerse the player in the exhibit. As in the Virtual Science Center Lobby, ambient noise will be played constantly. The triggering of specific sound effects will be initiated inside scripts during certain events. The list below will include events with their corresponding sound effects:

- *Highlighting Tube - ringing sound*
- *Select Tube - grabbing sound*
- *Deselect Tube - snapping sound*
- *Ball Drop - tap sound*
- *Ball roll - rolling sound*
- *Highlighting Button - ringing sound*
- *Button clicking - wood tap sound*
- *Particle effects - depending on what type of particles (fire, wind, etc)*
- *Finish - crowd cheering*

Textures

A wall will be very boring if it’s all the same gray color, so it is the same in the virtual exhibits if there are no textures. Textures are basically images that are wrapped around on an object’s surface, making it appear that it is real. Each functional component of the Gravitron Virtual Exhibit will have textures that provides a similar look to the physical exhibit.The quality of the textures varies but are required in the following:

- *Door object must have a door texture to distinguish that it is indeed a door.*
- *Wall objects must include some backgrounds so the user doesn’t feel like they’re in a prison.*
- *The Floor must include a floor pattern.*
- *The ceiling is debateable, but will also include a ceiling pattern.*
- *The Tubes texture will follow the same look as the Tubes in the physical Gravitron Exhibit.*
- *The Ball texture will be completely white.*

- *The Button objects will have the same universal texture theme to distinguish Buttons from the rest of the virtual exhibit components.*
- *The Magnetic Wall will have a grid background type of texture to distinguish it from a regular wall.*

4.3 System Design

4.3.1 Operational Scenarios

Player Functions in Gravitron Virtual Exhibit
There are some additional functions that the user can interact with specifically toward objects in the Gravitron Virtual Exhibit:

1. Button Interactions
The user is able to interact with buttons by mouse-clicking on them, which will initiate their corresponding scripts or actions,

2. Tube Interactions
The user is able to interact individually with a variety of different Tubes; performing actions such as selecting an individual Tube, then the ability to translate and rotate the Tube along the Magnetic Wall.

3. Door Interaction
The user is able to interact with the Door to exit at any time.

4.3.2 Camera Functions and Modes

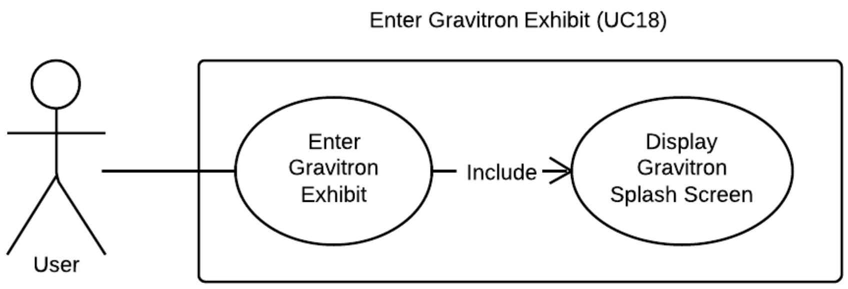
The way the Gravitron Virtual Exhibit room will function in terms of camera perspective, will be divided into 3 different camera modes (each one will ensure the other camera modes are disabled while the current one is enabled):

1. **Player Camera Mode** - *the user will start off with the Player Camera Mode on and the Ball Camera Mode and the Tube Camera Mode off when entering the Gravitron Virtual Exhibit room. This mode is technically a first-person perspective with regular functions such as moving in the 3D space, looking around with the mouse cursor, jumping in the y-plane and interacting with the objects in the Gravitron room.*

2. **Ball Camera Mode** - *once the user interacts with the Drop Button, the Player Camera Mode and the Tube Camera Mode will both be disabled and the Ball Camera Mode will be set to true or enabled. This allows the Ball Camera object to dynamically follow the Ball object in the x-y plane as the Ball traverses through the course by gravity. Once the Ball object has either reached or missed the Goal Cup at the bottom of the Magnetic Wall, the Ball Camera Mode will shut itself off and enable the Player Camera Mode while still leaving the Tube Camera Mode off.*

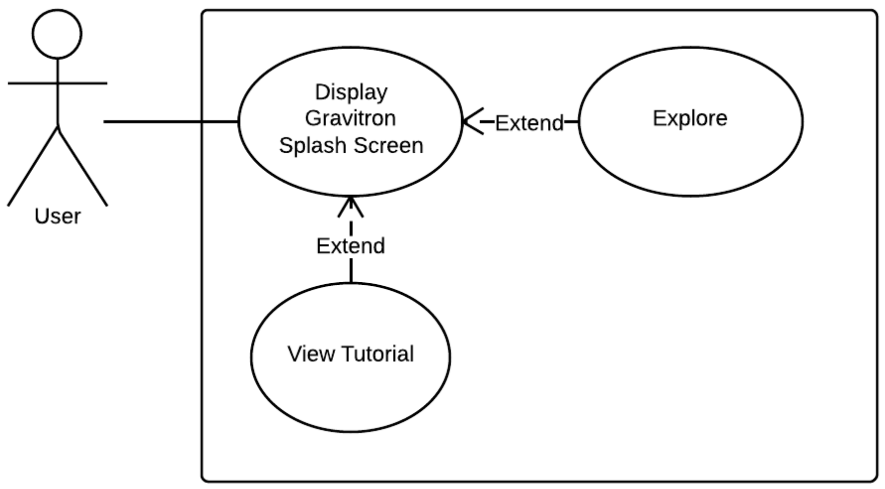
4.3.3 Gravitron Exhibit Use Cases

UC18. Enter Gravitron Exhibit – Upon clicking on the Gravitron Exhibit Door within the Lobby Center, the user is then taken to the Gravitron Exhibit Room. From this room, the Gravitron Splash Screen is then displayed. See Use Case 18 below.


Enter Gravitron Exhibit (UC18)
User
Enter Gravitron Exhibit
Include
Display Gravitron Splash Screen

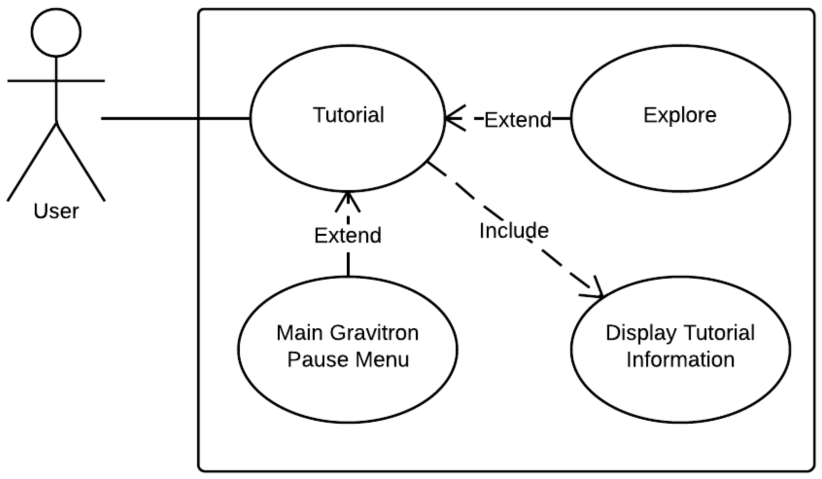
UC19. Gravitron Splash Screen – Information on the Gravitron Exhibit is displayed on the Splash Screen. Information includes a welcome message, what the Gravitron Exhibit is about, and instructions on what to do next. The user will also be given two options: “Explore the Gravitron Exhibit” and “View tutorial”. Each of these options are selectable and will display a different interface upon selection. See Use Case 19 below.

Gravitron Splash Screen (UC19)



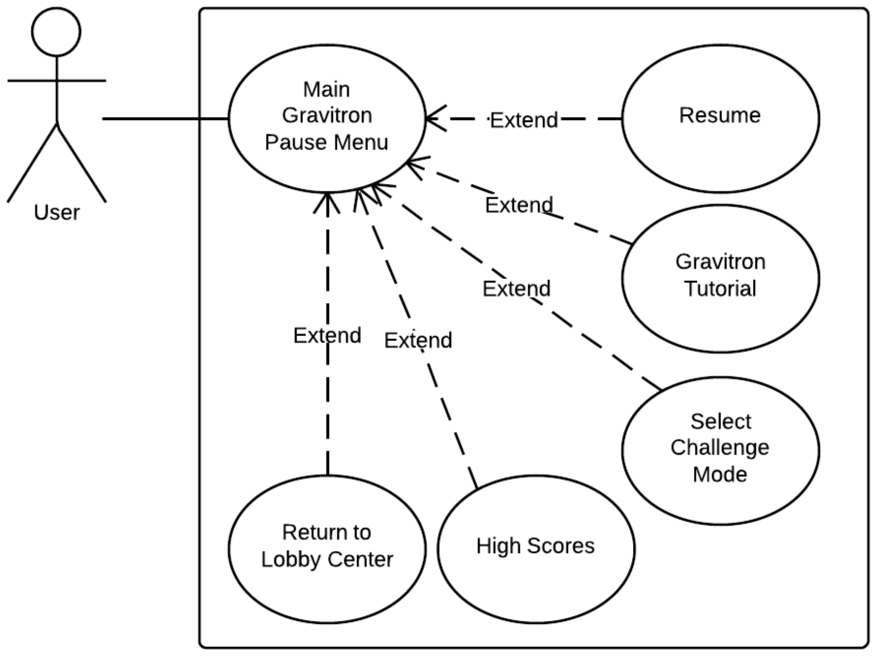
UC20. Gravitron Tutorial – When the “View Tutorial” button is selected from the Gravitron Exhibit Splash Screen or from the Main Gravitron Pause Menu, a new interface is brought up with information on how to use the Gravitron Exhibit. Information will include controls for within the exhibit and what to do next. The Gravitron Tutorial gives the user two options: “Explore” and “Main Menu”. See Use Case Diagram 20 below.

Gravitron Tutorial (UC20)

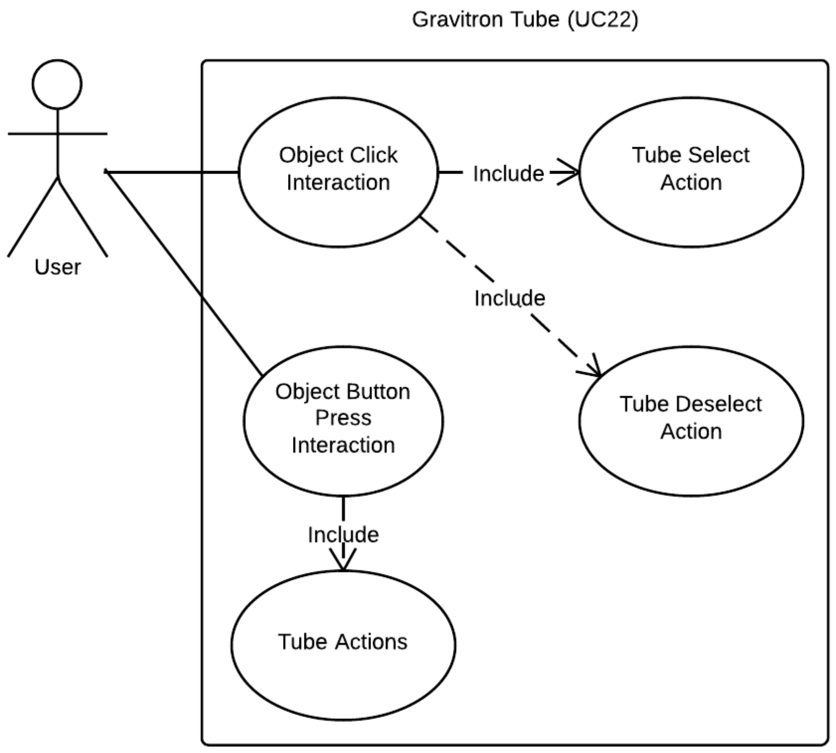


UC21. Pause Exploring of Gravitron Exhibit – When the Pause Menu is brought up, the user is no longer able to control the player. Five options are displayed while the user is within the Gravitron Exhibit. These options are “Resume”, “View Tutorial”, “Select Challenge Mode”, “View High Scores”, and “Return to Lobby Center”. See Use Case Diagram 21 below.

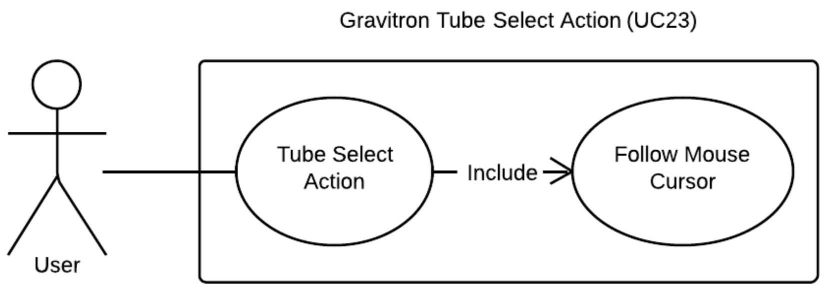
Pause Exploring of Gravitron Exhibit (UC21)



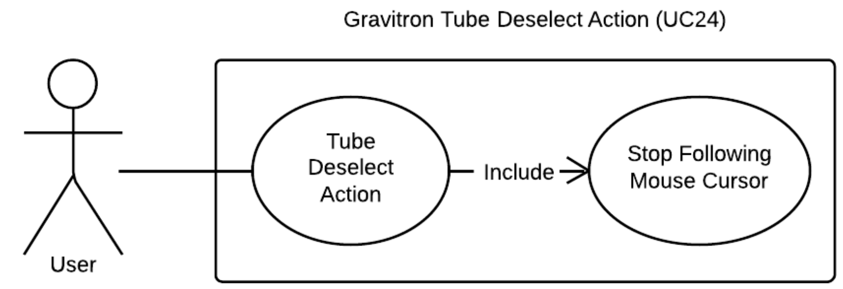
UC22. Gravitron Tube – The Gravitron Exhibit Room contains many different Tube objects along panel by the Magnetic Wall. When the user clicks on one of these Tubes, a new Tube (which is called a cloned Tube) is created on a fixed position on the Magnetic Wall object and is also selected. The Tube object will remain selected until the user clicks on the Tube object again. While a Tube is selected, the user may also press key commands on the keyboard for Advanced Tube Movement. See Use Case Diagram 22 below.



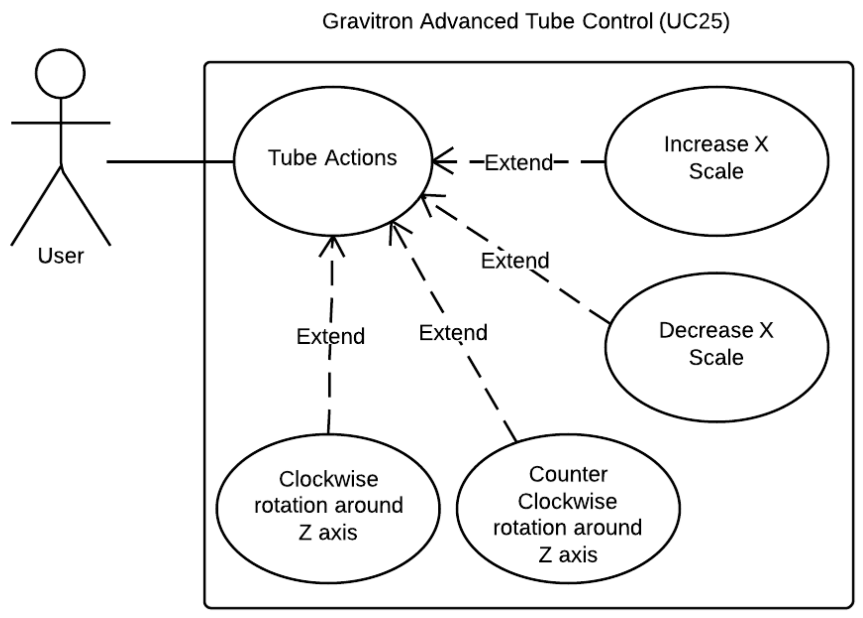
UC23. Gravitron Tube Select Action – When a Tube is selected, it performs the Tube Select Action. While a Tube is selected it will either follow the user's mouse cursor along the Magnetic Wall or follow the user's commands on the keyboard. If the user's cursor moves off the Magnetic Wall, the Tube object will not follow the cursor off the Magnetic Wall, likewise for the keyboard commands as well. The Tube object will stay within the bounds of the Magnetic Wall. See Use Case Diagram 23 below.



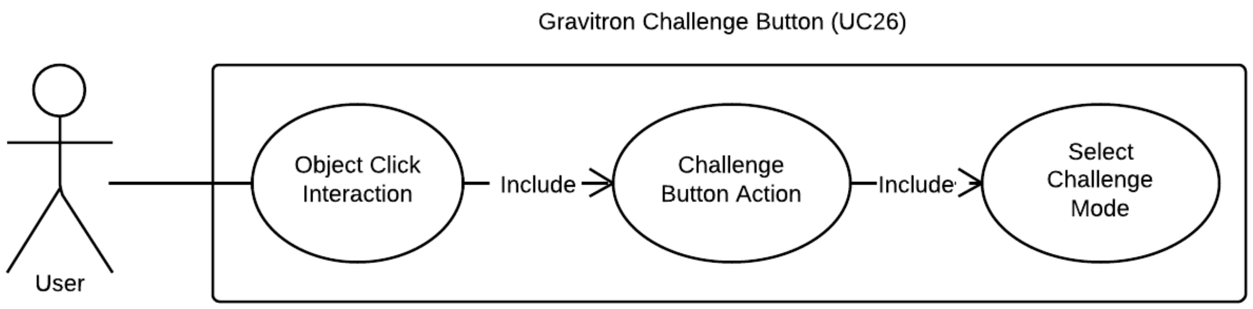
UC24. Gravitron Tube Deselect Action – When a Tube is deselected, the Tube stops following the user's mouse cursor and places itself at its current location. See Use Case Diagram 24 below.


Gravitron Tube Deselect Action (UC24)
User
Tube Deselect Action
Include
Stop Following Mouse Cursor

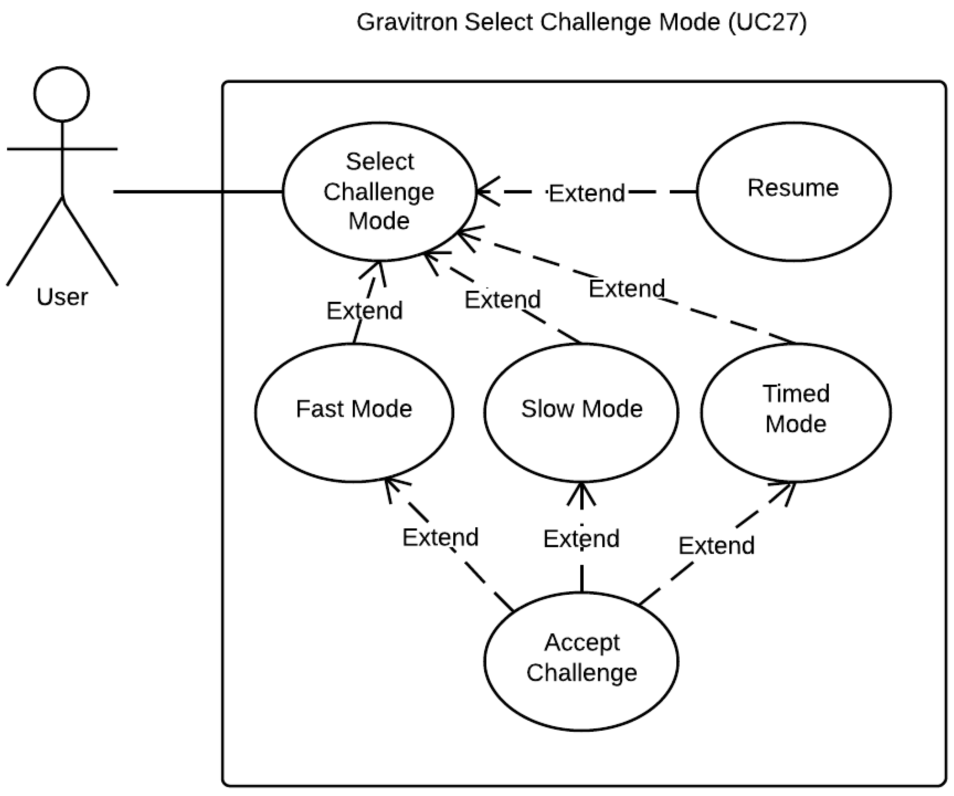
UC25. Gravitron Advanced Tube Control – While a Tube object is selected, the user has the option of pressing keyboard buttons for more advanced control. There are 4 different advanced Tube controls. The user can rotate the Tube slightly in a clockwise direction and also counter clockwise direction. The user can also increase the size of the Tube as well as decrease the size of the Tube. See Use Case Diagram 25 below.


Gravitron Advanced Tube Control (UC25)
User
Tube Actions
Extend
Increase X Scale
Extend
Decrease X Scale
Extend
Extend
Clockwise rotation around Z axis
Counter Clockwise rotation around Z axis

UC26. Gravitron Challenge Button – The Gravitron Exhibit Room contains one Challenge Button object. The Challenge Button is used to change which challenge the user is currently working on within the Gravitron Exhibit. When the user clicks on the Challenge Button object, the Challenge Button Action is called.The Challenge Button action displays the Challenge Selection Screen. See Use Case Diagram 26 below.

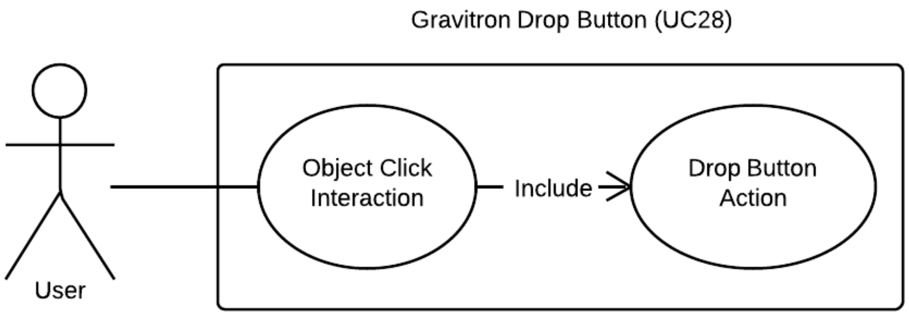


UC27 Gravitron Select Challenge Mode – The Challenge Selection Screen is used to change which challenge the user is currently working on within the Gravitron Exhibit. There are currently 3 different Challenge Modes: Fast Mode, Slow Mode, and Timed Mode. After the user has selected a Challenge Mode, they then can accept the challenge. Once a Challenge Mode has been accepted, the user is brought back to the Gravitron Exhibit. The user also has the option to return to the Gravitron Exhibit without accepting a new Challenge Mode. See section 4.2.2 Challenge Modes for more information on each challenge. See Use Case Diagram 27 below.



UC28. Gravitron Drop Button – The Gravitron Exhibit Room contains one Drop Button object. The Drop Button allows the user to test the Tube configuration on the Magnetic

Wall. When the user clicks on the Drop Button object, the Drop Button Action is called. See Use Case Diagram 28 below.

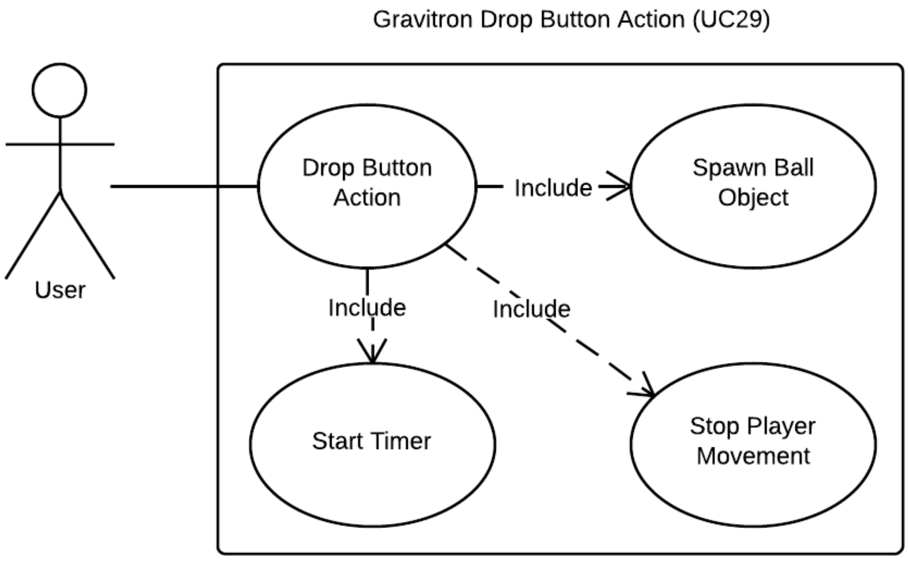


UC29. Gravitron Drop Button Action – When the Drop Button is clicked, the Drop Button Action is called. Within this action, three other actions occur:

First, the user loses control over the Player. The user will no longer be able to move the player or look around. Instead the camera will focus on the Magnetic Wall so that the user can see the Tube configuration.

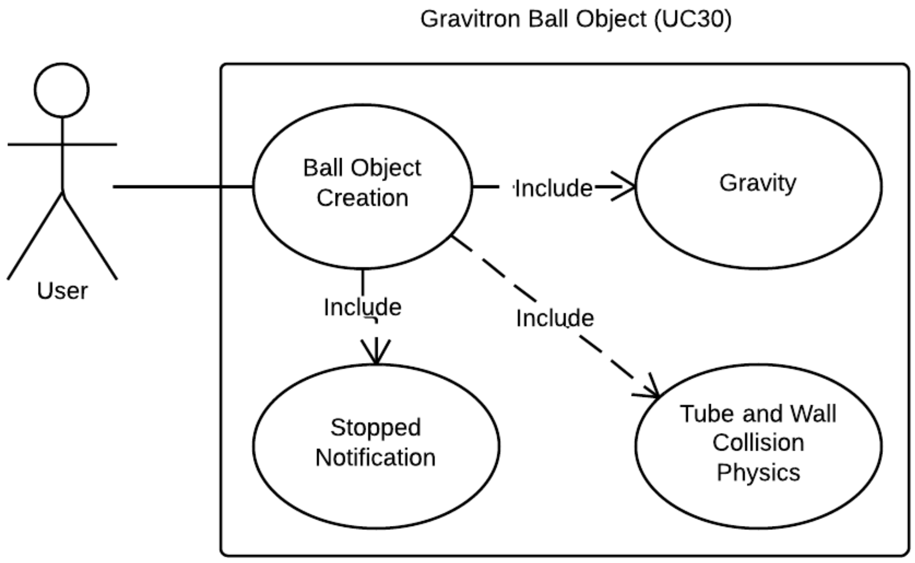
Second, a Ball object is spawned at the top of the Magnetic Wall.

Third, the Timer object will start counting (in seconds) so that the user can see how long it takes for the Ball object to reach either the Goal Cup object or the Gutter object. See Use Case Diagram 29 below.

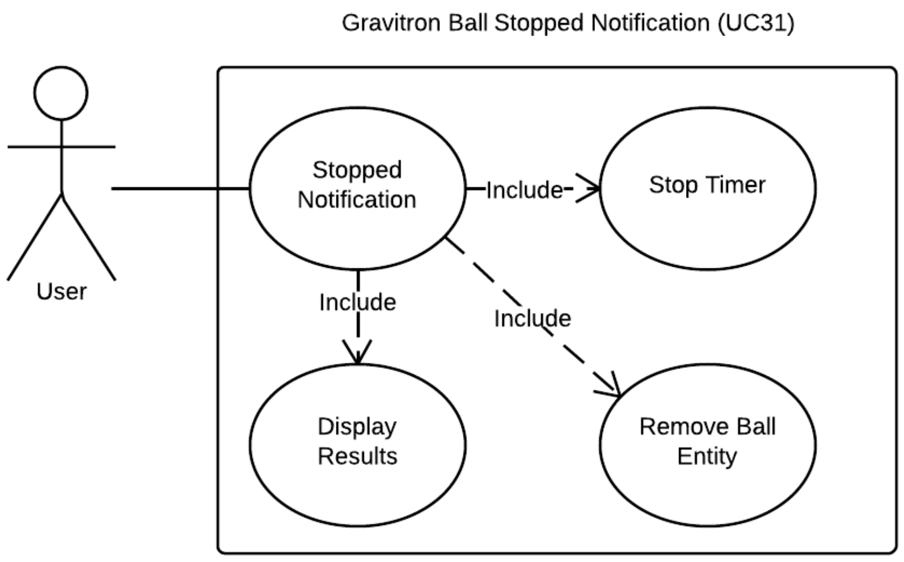


UC30. Gravitron Ball Object – The Gravitron Exhibit Room contains at most one Ball object. The Ball object falls from the top of the Magnetic Wall. It will collide with the

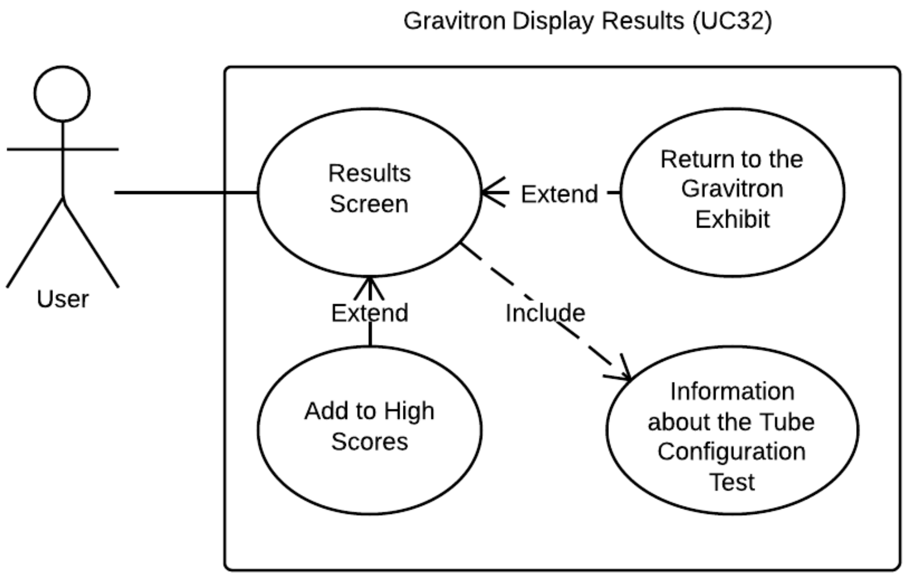
Tubes on the Magnetic Wall and cannot move outside of the bounds of the Magnetic Wall. When the Ball object stops moving, collides with the Gutter object, or collides with the Goal Cup object, the Stopped Notification is called. See Use Case Diagram 30 below.



UC31. Gravitron Ball Stopped Notification – When the Ball Stopped Notification has been called, the Tube configuration testing has finished. Three actions occur within this call. First, the Timer stops timing. Second, the Ball object is removed from the Exhibit Room. Third, the Tube configuration test results are displayed to the user. The results will depend on the location the Ball object stopped at. If the Ball object stopped within the Goal Cup object, then the Tube configuration passed. If the Ball object is within the Gutter or stopped on top of one of the Tubes, the configuration fails. The Floor object will act as a safety-net in rare cases the Ball went through the Gutter object. See Use Case Diagram 31 below.

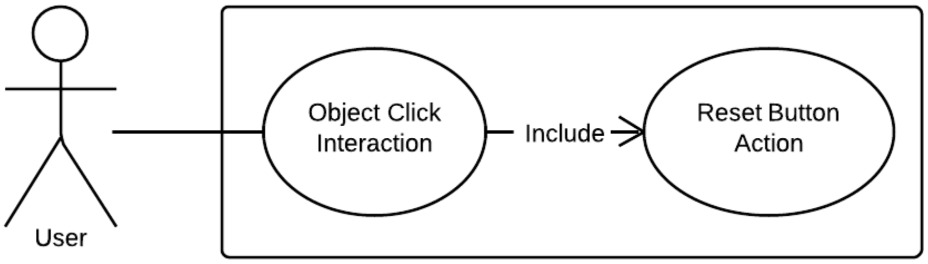


UC32. Gravitron Display Results – Information on the Tube configuration test is displayed on the Results screen. Information includes the current challenge, the amount of time in seconds it took for the Ball object to drop, and whether the Ball object successfully made it to the goal. If the Tube configuration results are better than saved results, the user's results will then be added to the High Scores. The user will be given only one option: "Return to the Gravitron Exhibit". See Use Case Diagram 32 below.


Gravitron Display Results (UC32)
User
Results Screen
Extend
Return to the Gravitron Exhibit
Extend
Include
Add to High Scores
Information about the Tube Configuration Test

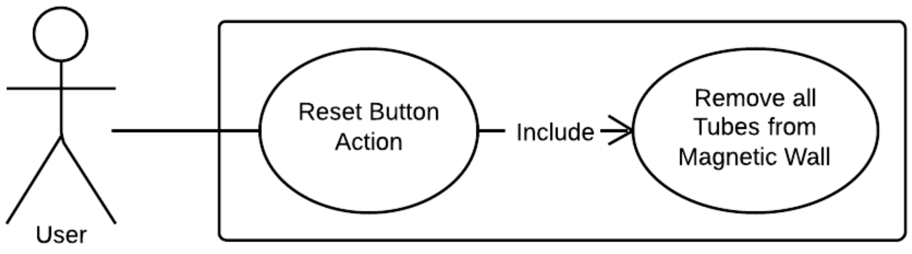
UC33. Gravitron Reset Button – The Gravitron Exhibit Room contains one Reset Button object. When the user clicks on the Reset Button object, the Rest Button Action is called. See Use Case Diagram 33 below.

Gravitron Reset Button (UC33)



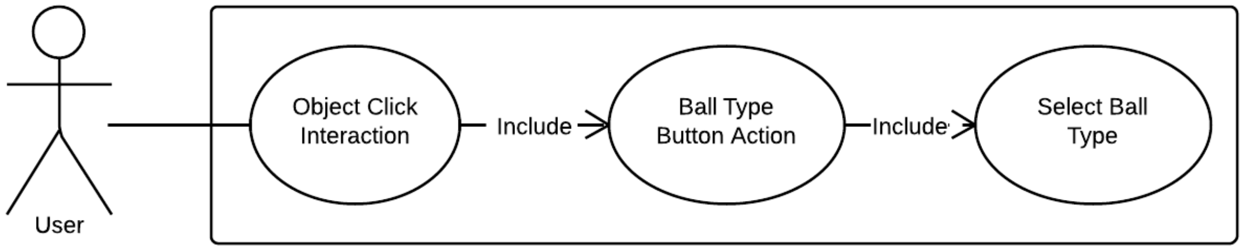
UC34. Gravitron Reset Button Action – When the Reset Button is clicked, the Reset Button Action is called. The Reset Button Action goes through all the Tubes on the Magnetic Wall and removes them, giving the user a clean wall to create a new Tube configuration. See Use Case Diagram 34 below.

Gravitron Reset Button Action (UC34)

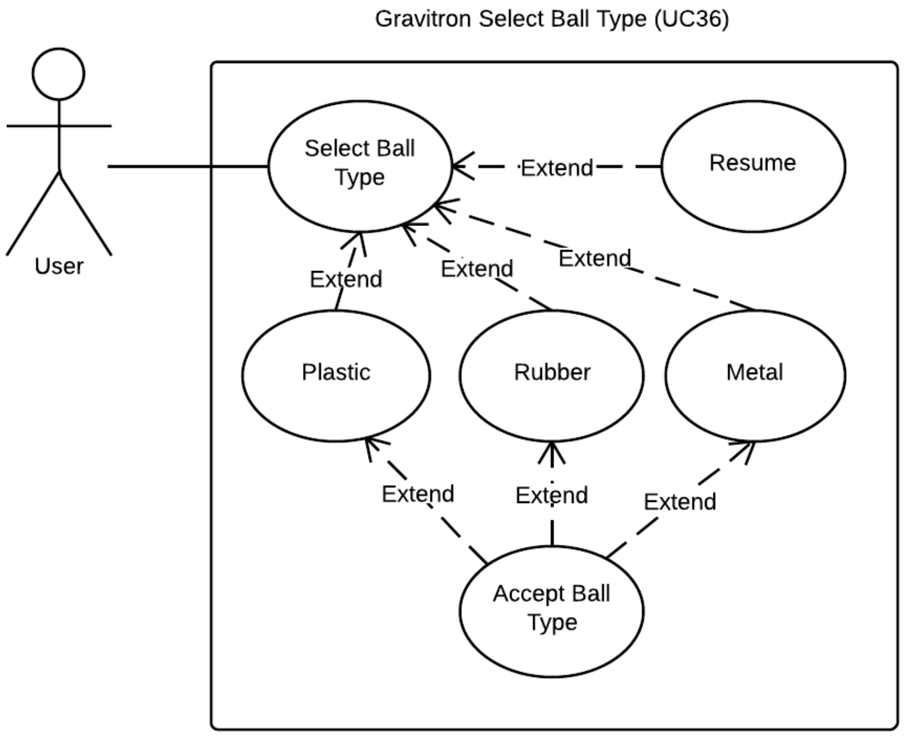


UC35 Gravitron Ball Type Button – The Gravitron Exhibit Room contains one Ball Type Button object. The Ball Type Button is used to change the physics for the Ball object within the Gravitron Exhibit. When the user clicks on the Ball Type Button object, the Ball Type Button Action is called.The Ball Type Button action displays the Ball Type Selection Screen. See Use Case Diagram 35 below.

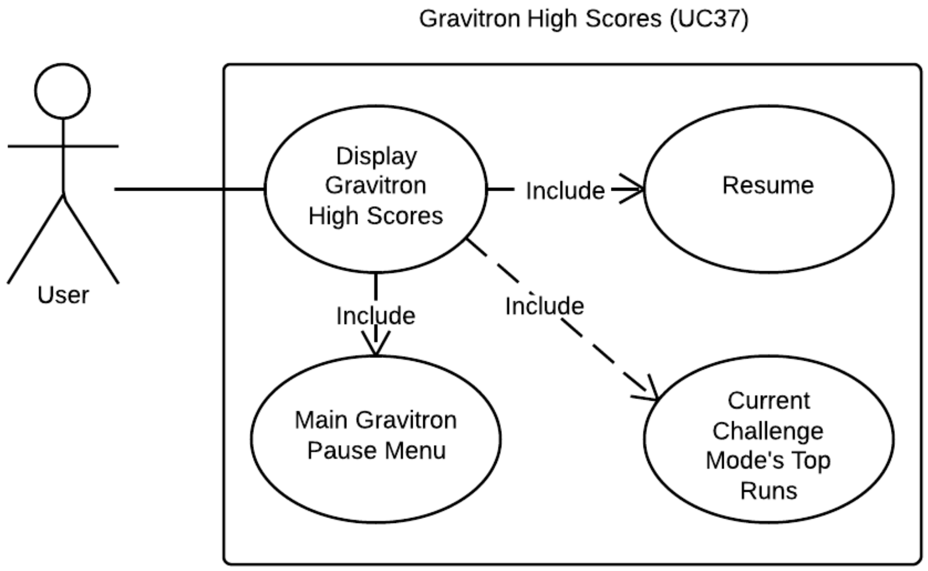
Gravitron Ball Type Button (UC35)



UC36 Gravitron Select Ball Type – The Ball Type Selection Screen is used to change which challenge the user is currently working on within the Gravitron Exhibit. There are currently 3 different Ball Type: Plastic, Rubber, and Metal. After the user has selected a Ball Type, they then can accept it and return to the Gravitron Exhibit. The user also has the option to return to the Gravitron Exhibit without accepting a new Ball Type. See Use Case Diagram 36 below.



UC37 Gravitron High Scores – When the “High Scores” button is selected from the Main Gravitron Pause Menu, a new interface is brought up with the top ten results the user has achieved while in the Gravitron Exhibit. Each Challenge Mode has its own top ten achieved runs. The Gravitron Tutorial gives the user two options: “Resume” and “Main Menu”. See Use Case Diagram 37 below.


Gravitron High Scores (UC37)
User
Display Gravitron High Scores
Include
Resume
Include
Include
Main Gravitron Pause Menu
Current Challenge Mode's Top Runs

4.3.4 Storyboards

The designing process of the Gravitron Virtual Exhibit room starts out with a storyboard of how we want the room to be displayed visually. From that point on, we focused on each object and define their corresponding purpose and relations to other objects in the room.

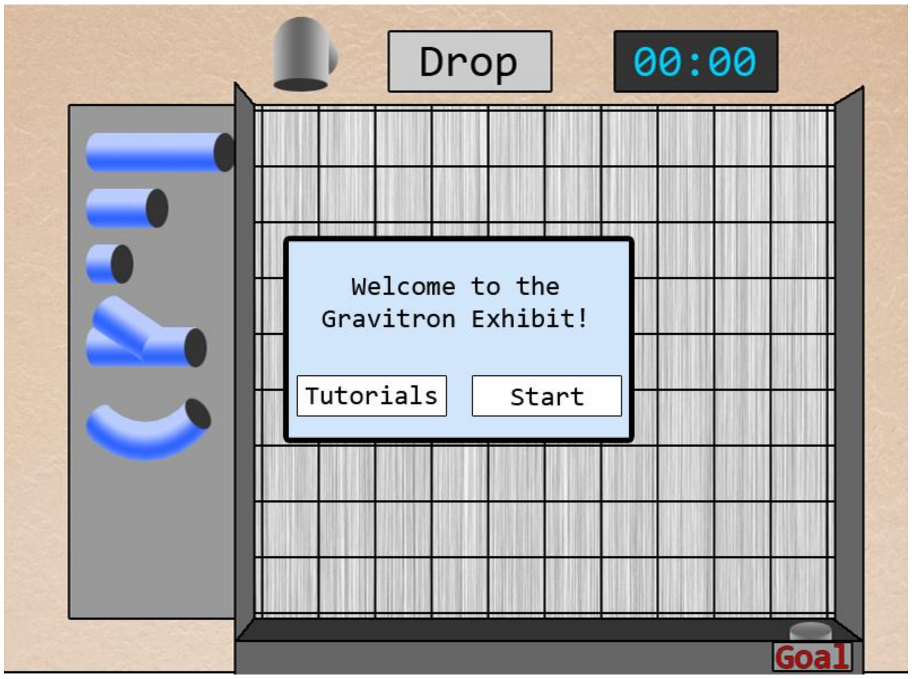
When the user enters the Gravitron Virtual Exhibit room (Figure 4.1), they will be greeted with a pop up window. At this point, the user can only interact with the pop up features and nothing else.

If the user clicks on the Tutorial button, a pop-up window of text regarding how to utilize the Gravitron Virtual Exhibit will be properly displayed to the user to read and navigate.

If the user clicks on the Start button, the pop up window will dissipate and the user is able to move freely in first-person perspective again.

The user will then be exposed to the default layout of the Gravitron Virtual Exhibit at this point, in which a Magnetic Wall with a set of Tube objects will be displayed on the left panel and the Button objects on the lower right panel.

Figure 4.1



As seen in the picture (Figure 4.2), the exhibit room is still in 3 dimensional space and the user can move around and exit at anytime through the Door object.

Noted additional player functions:

- Able to select Tubes
- Able to rotate Tubes (x and y plane only)
- Able to move Tubes (x and y plane only)
- Able to deselect Tubes
- Able to attach Tubes to the grid background
- Able to interact with Buttons
- Able to access the Pause command to bring up the Pause Menu

Figure 4.2

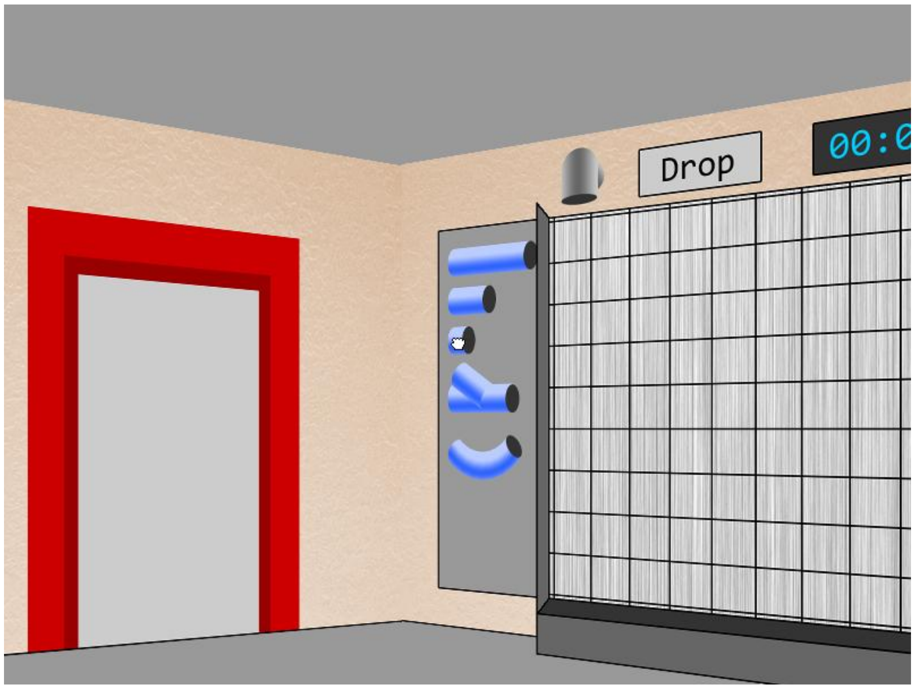
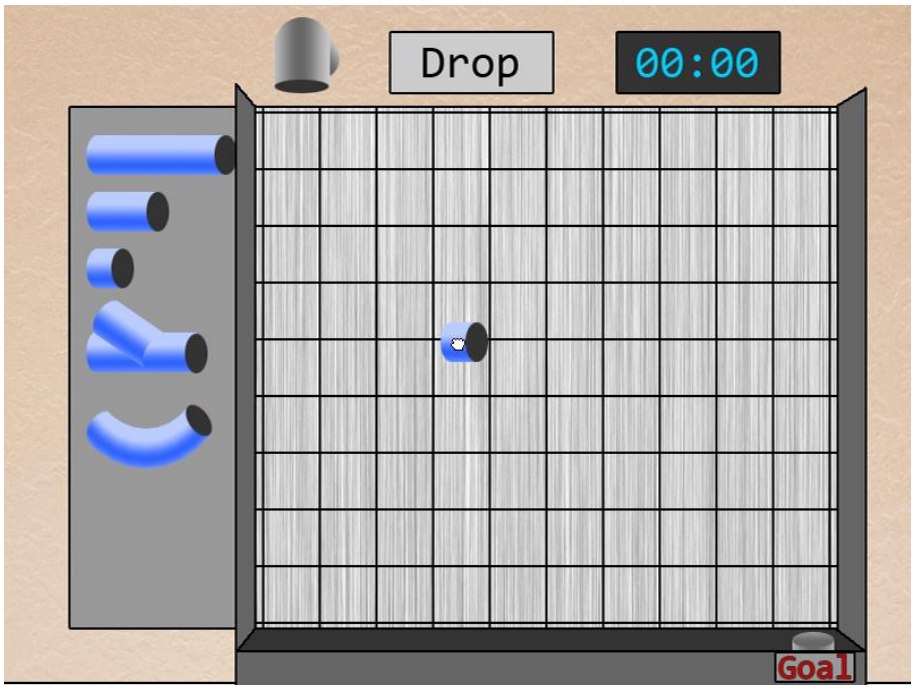


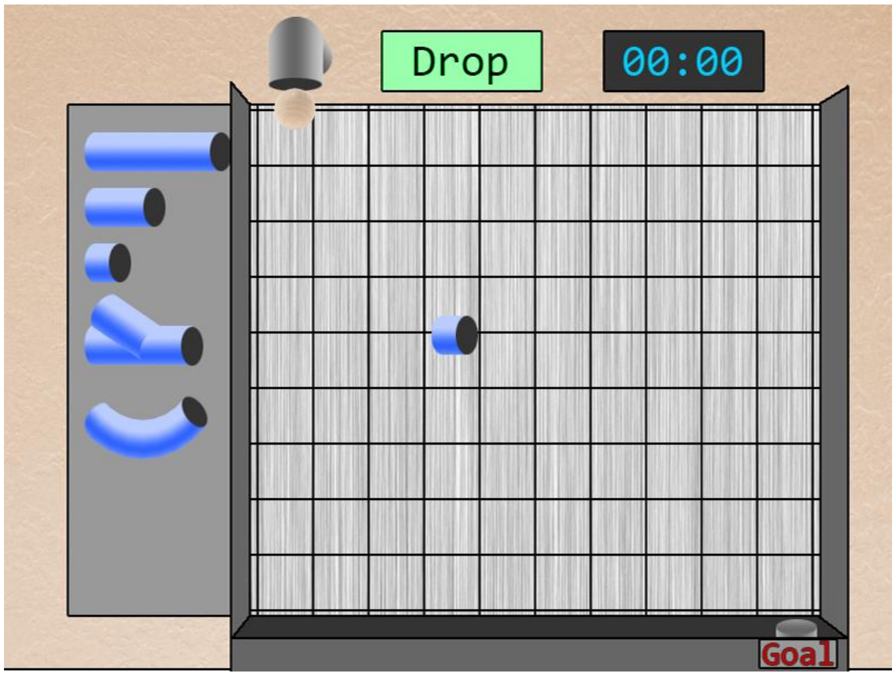
Figure 4.3



Once the user goes through the implementation process and assembles the set of Tubes to their preference, they then are able to press the Drop button (Figure 4.4), in which a Ball will be instantiated and appear from the top fixed-positioned, gray Tube.

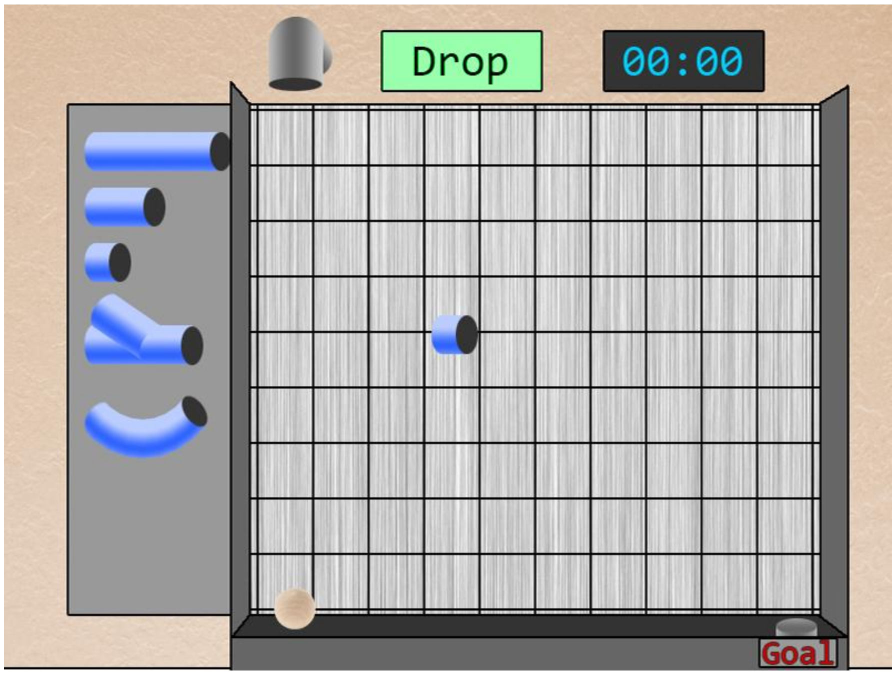
The Timer object that starts at 00:00 s (in seconds), will begin timing once the Drop button object is activated.

Figure 4.4



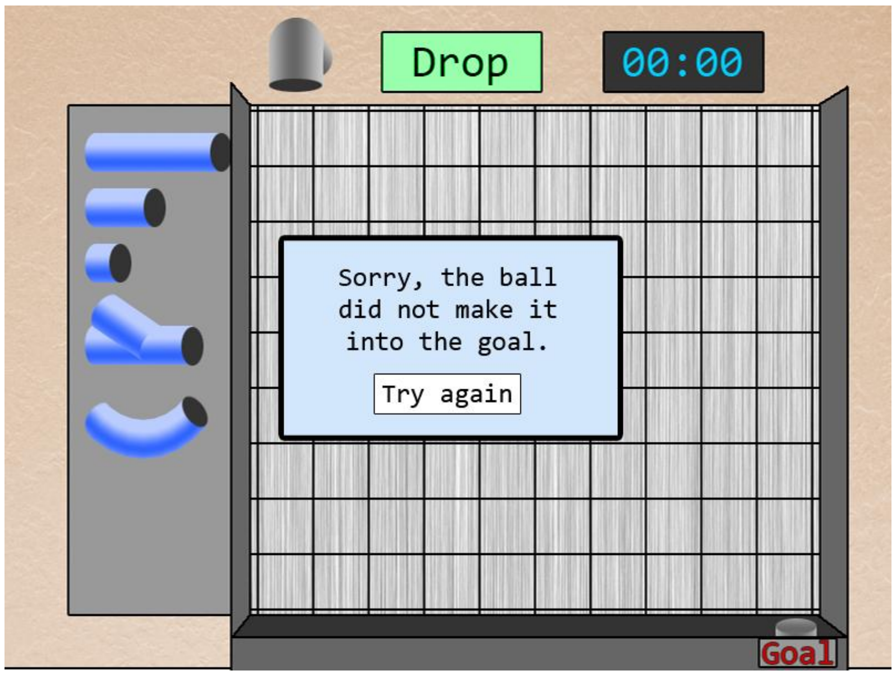
After the Drop button event is initiated, the camera perspective should be switched to the Ball Camera Mode and dynamically follow the Ball as it traverses through its course (whether by reaching the Goal Cup or touching the Gutter).

Figure 4.5



If the Ball stops at any location that is not within the Goal Cup, then a message should pop up (Figure 4.6) informing the player of the failed result and disable the Ball until the Drop button is triggered again.

Figure 4.6



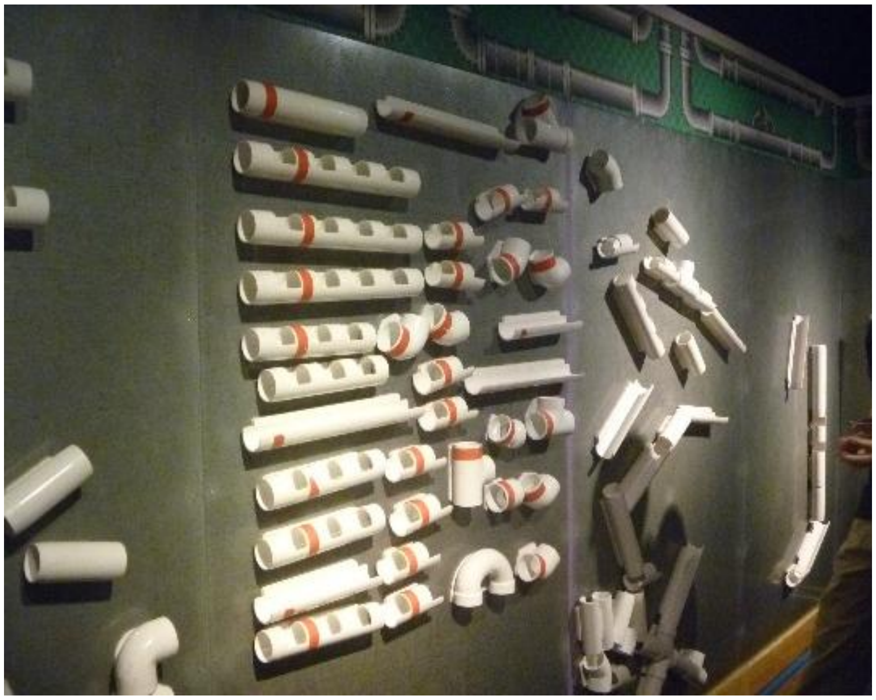
Sorry, the ball did not make it into the goal.

4.3.5 Prototype Object Models

The objects in the Gravitron Virtual Exhibit were modeled after the real version of the exhibit. We took pictures of the exhibit and used them as reference materials when creating their virtual counterparts in Blender.

Tube Objects

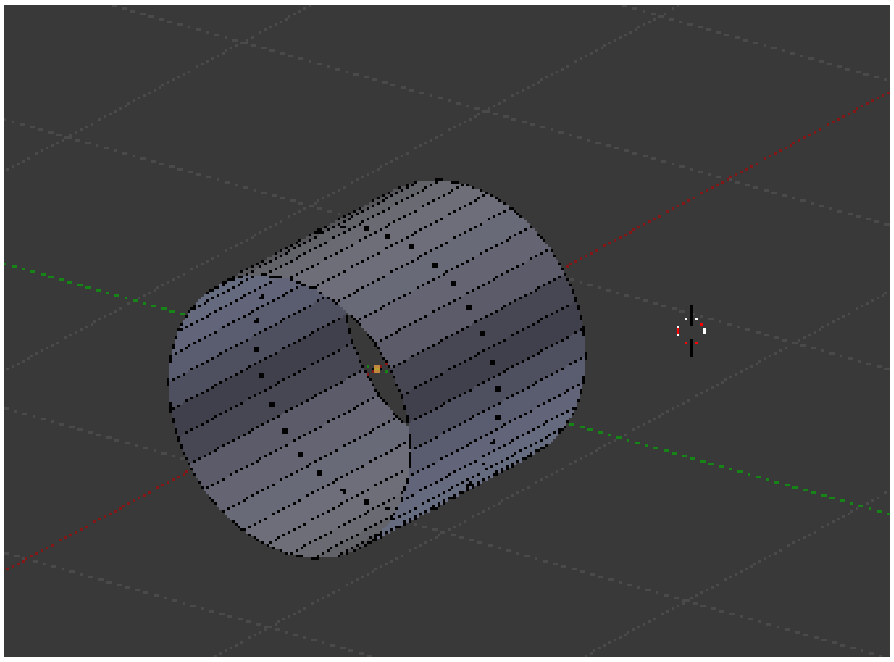
The picture below shows the Gravitron exhibit Tubes that we are referencing when modeling their virtual counterparts in Blender.



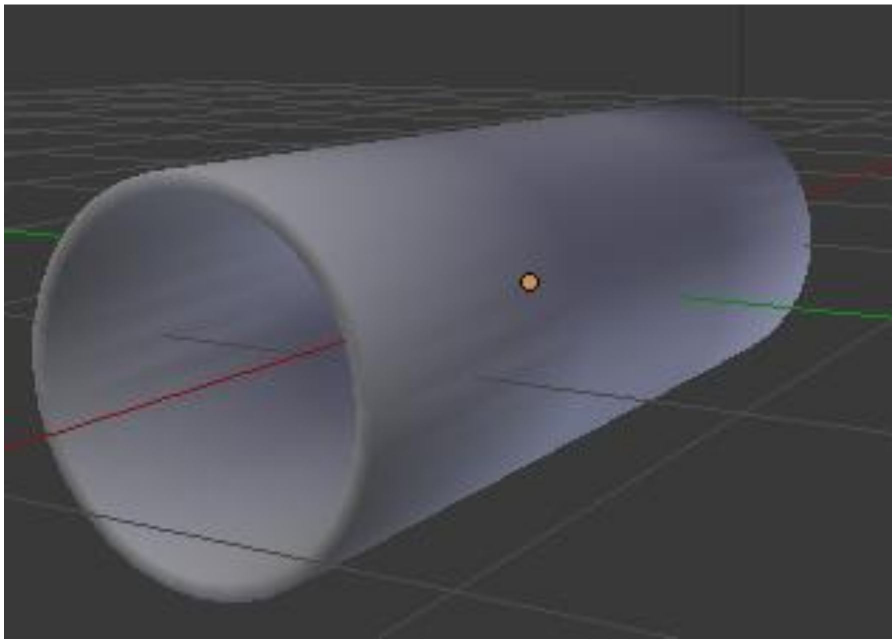
The Tubes in the Gravitron Exhibit are not complex shapes, so we are able to easily duplicate the different Tubes' shape in Blender such that they are able to interact with the Ball like they would in real life. Here are some sample Tubes we made and how to model them in Blender:

The hollow Tube in the picture below is simply a cylinder object that Blender provides. After adjusting the Cylinder that was initially in an upright position to sideway position, we then delete the face at the end of each cylinder as shown in the picture. When the faces are deleted, the Cylinder becomes a hollow Tube.

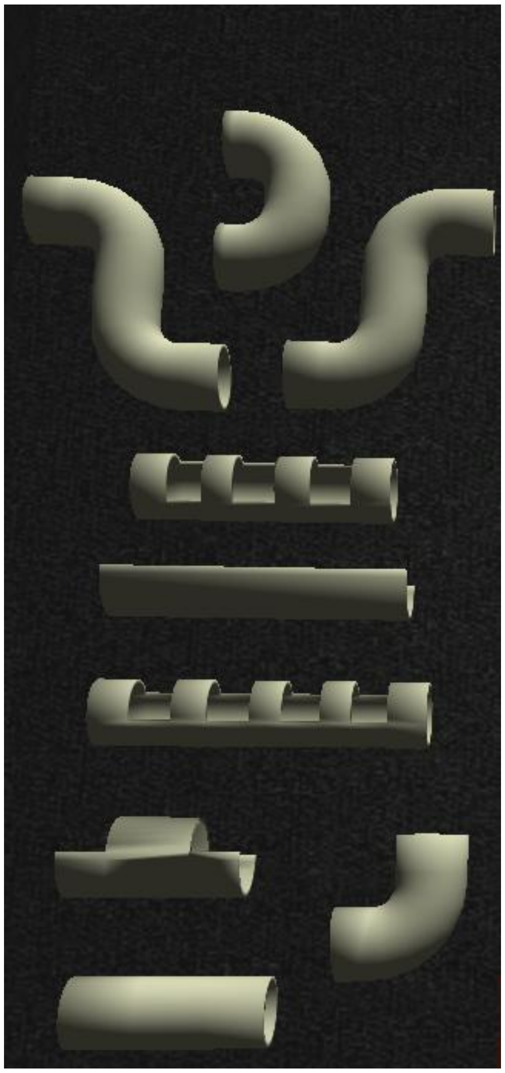
(Keyword - Face: each polygon is called a face consisting of vertices in each corner, which is what gives an object that surface look.)



As depicted in the picture above, the cylinder looks too thin to be utilized in the virtual exhibit since the user will not be able to see the objects when viewing in certain angles. So the next step is to adjust the solidity (which gives it more density) to the value 0.40. The solidity value is based on trial and error to decide what number looks best and becomes completely visible to the user from all angles, so it's not a calculated value, but it is best for all Tube objects to have the same solidity since it ensures the Ball object will pass through each Tube without getting stuck. Once the Tube looks solid enough, we finally smoothed the whole object and the result is the picture below.

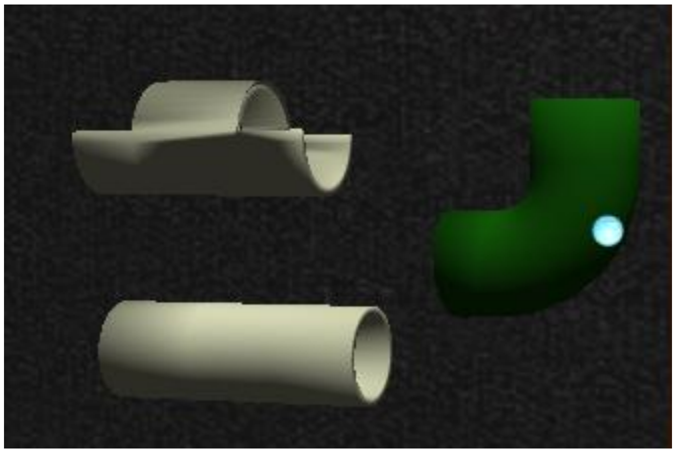


Prototype of overall Tubes that are implemented:

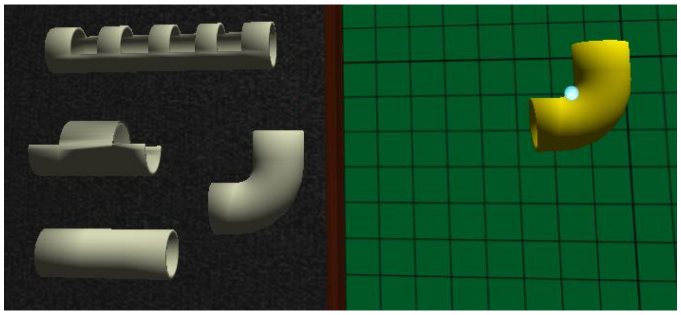


The user should be able to distinguish which Tubes they have selected, thus we came up with a color code to visually point out the selected Tube as well as collisions with another Tube on the Magnetic Wall:

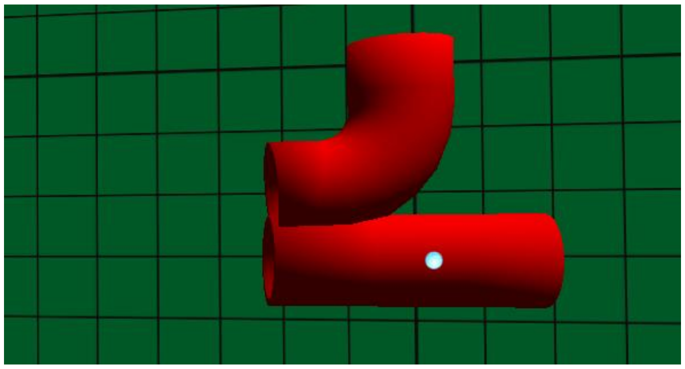
User selects a Tube (highlighting - Green)



User's selection on the Magnetic Wall (cloned - Yellow)

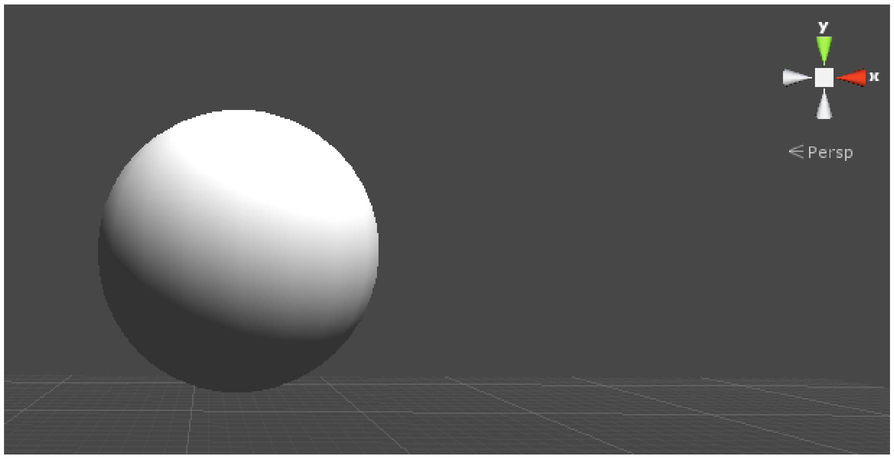


Tube collision with another Tube (collision - Red)



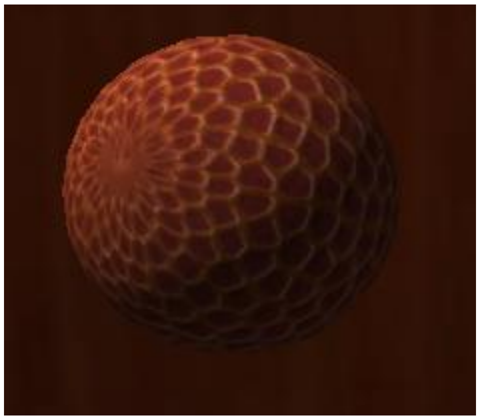
Ball Object

The Ball object is a simple spherical object that is created in Unity. It is much more efficient to use this perfectly rounded sphere than to make our own. We simply attached a rigid body component (which allows the parent object to act under the control of physics, such as gravity) to the Ball object.

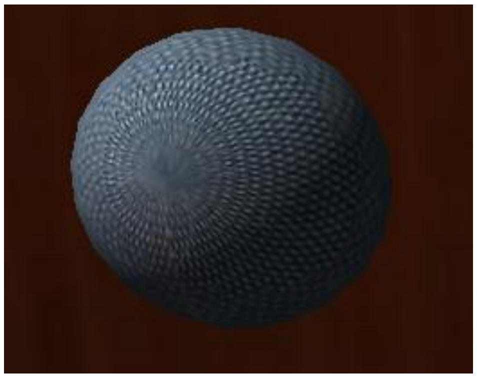


Pertaining to the idea of physics in the Gravitron Exhibit, we decided to implement the ability for the user to change the Ball object types to further be creative and experiment with. Currently, the types of Balls that is in effect are the regular, metal and rubber. Each type has its own physics material that determines for example, the bounciness of the rubber material. Aside from the physics, textures are equally important for the user to distinguish between their selected Ball types when they click on the Change Ball Button that alternates between the three Balls.

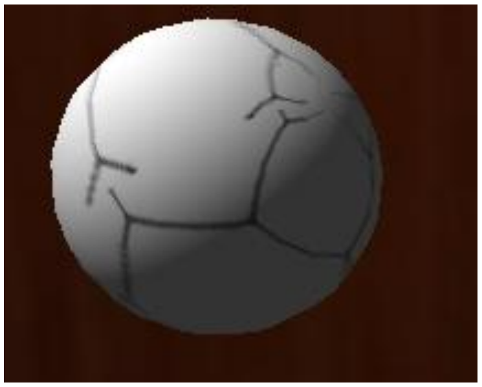
Rubber Ball:



Metal Ball:

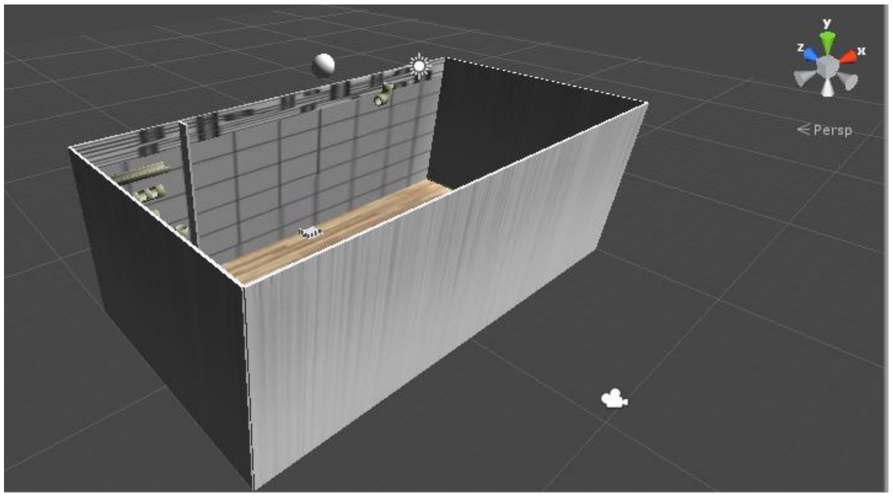


Regular Ball:

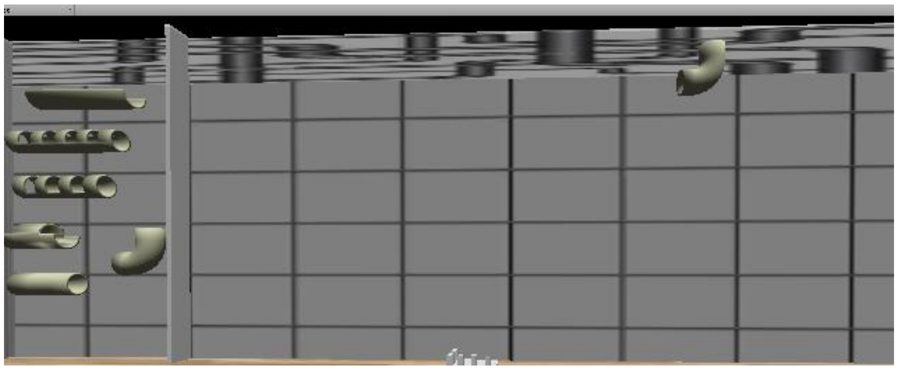


Wall Objects

The Wall objects are simple elongated cube objects that has been thinned and scaled accordingly to a room and are acting as boundary lines between the user and the abyss (referring to the area outside of the Gravitron Virtual Exhibit that goes on infinitely); without it, the user would easily fall over the edges. The first exhibit layout (called prototype 1) is a simple room that encloses the Gravitron Virtual Exhibit.

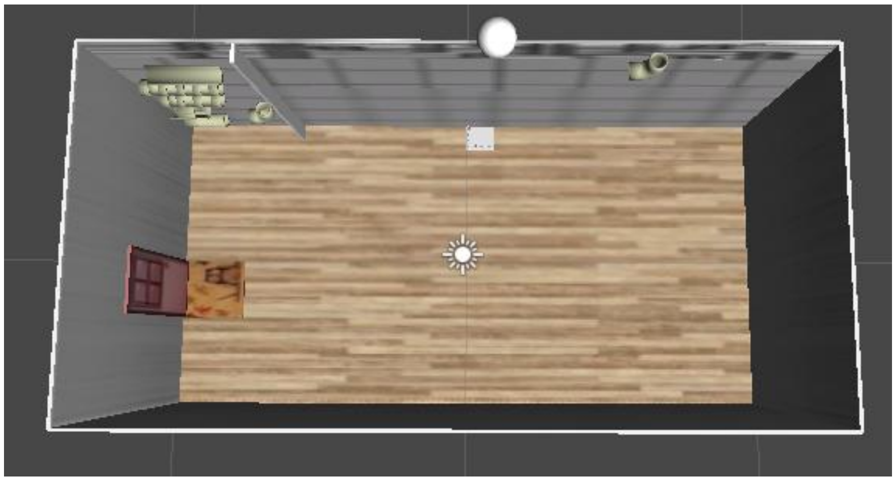


The grid background that all the Tubes are attached to is called the Magnetic Wall and is the central component of this exhibit. The wall is where the user will assemble and test their Tubes design layout.



Floor and Ceiling Objects

The Floor and Ceiling objects are both plane objects (like a flat sheet of paper). However, only the Floor object has collision checking with the user so that the user will not fall through the surface. The Floor object also checks for collisions with the Ball object in case the Ball passes through a gap between tubes, or through or around the Gutter object.



The Floor object in the picture above has a wooden texture attached to it to distinguish the difference between it and the walls. The texture will change before the final product once we obtain a texture that more closely matches the floor at the real version of the Gravitron exhibit. The Ceiling object does not need to be interacted with since the user can never jump high enough to escape the enclosed room. This also saves some space since a script is not needed to handle the interaction.

Door Object

The Door object serves as an exit portal to return the user back to the Virtual Science Center Lobby. It will be located on one of the Walls.

The Gravitron Virtual Exhibit prototype 1 has a red cartoonish Door as a placeholder until we can obtain or design a more realistic one.

Camera Objects

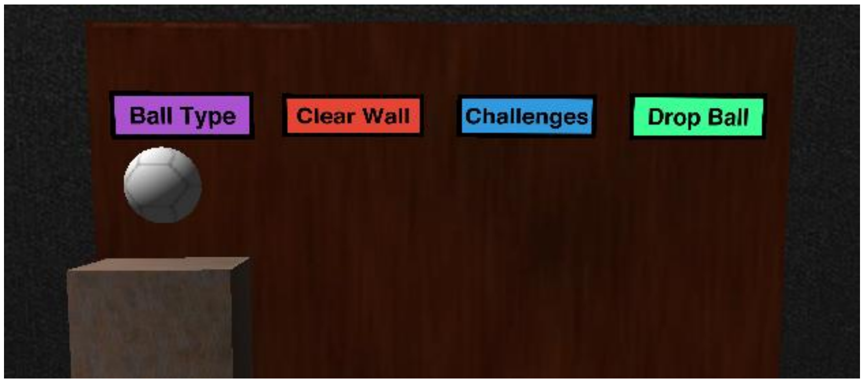
There are two different Camera objects that do the same basic function of displaying the screen to the user. However, the way each Camera is displaying the screen is different depending on their scripts.

- Ball Camera - *this Camera object will act upon the Ball object only.*
- Player Camera - *this Camera object is integrated into the first-person controller (or first-person perspective) in which it follows the player or user object.*

Button Objects

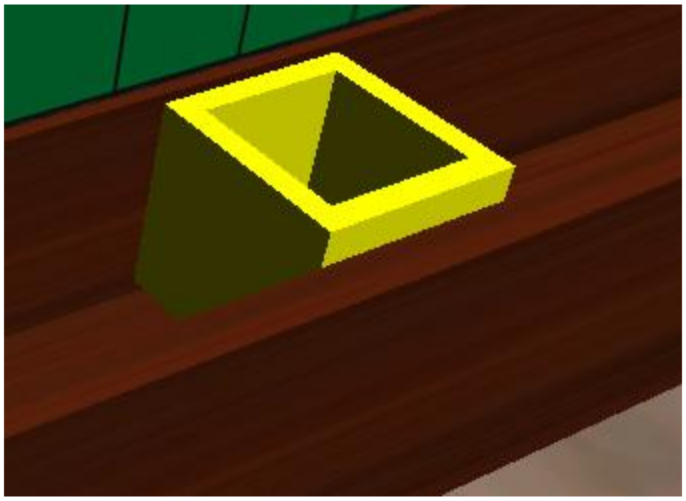
The Button objects (Reset, Drop, Challenge Modes, and Ball Type) will each be comprised of a small and thin rectangular object that will have a text on each of them and the labels will be self-explanatory to the user.
For example, the Reset Button will have the word Reset on it and the Drop Button will have the word Drop on it. The Button objects will be attached by the Magnetic Wall lower section to give the illusion that they are functional.

General information about each Button objects: the Reset Button will reset the Gravitron Virtual Exhibit room, the Drop Button will create a Ball object and follows it, the Challenge Modes Button will trigger a pop-up window for the user, and the Ball Type will alternate between different types of Ball objects for the user to use in the simulation.



Goal Cup Object

The Goal Cup serves as the main objective for the user. In order for a challenge to be complete, the Ball must hit the bottom part of the inside of the Goal Cup.

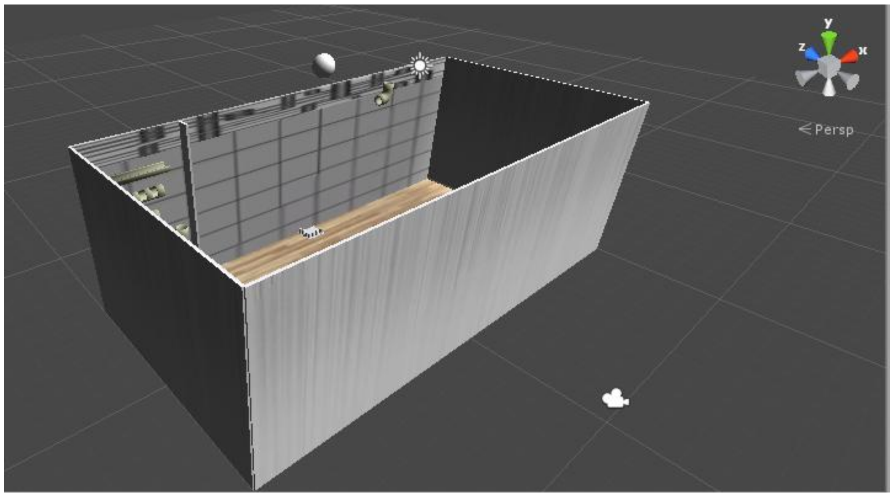


Gutter Object

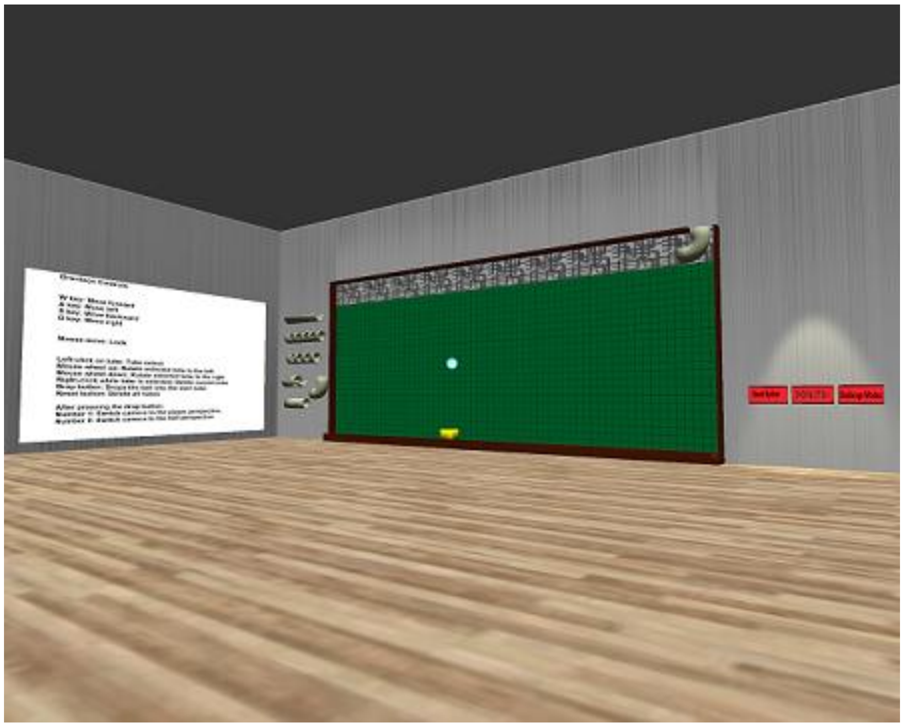
The Gutter object will serve as one of the final destination of the Ball object in case the Ball does not roll into the Goal Cup object. The Gutter will be a long, horizontal, cylinder object with the top portion chopped off. The location of the Gutter object will be right below the Goal Cup object on the Magnetic Wall.

Gravitron Room Prototypes

The first prototype of the Gravitron Exhibit is a simple rectangular room with no colors to allow convenient debugging and setting up other functionalities first. The room contains a set of Tubes, the Magnetic Wall, and the Ball object to ensure that these three main components work as intended before the implementation of other components.



The second prototype builds on the improvement of the first prototype, with the addition of the Button objects (Reset, Drop, and Challenge Modes). The Magnetic Wall is revamped to look more appealing to the eyes as well as a visible ceiling. The look of the room is still dull and grey, but for the purpose of this prototype, each new implementations will have to work with the other components first before going to the next stage. This Gravitron Exhibit prototype is also the version that was utilized for public testing in Otronicon event at the Orlando Science Center, where feedbacks and suggestions were taken for the next update.



The final version of the virtual Gravitron Exhibit. The aesthetics of the whole room was re modified to match the colors of the physical Gravitron Exhibit at the OSC. New Buttons were also implemented (Clear Wall replacing Reset and Ball Type). Each Buttons are color coded to easily distinguish between them while mounted on a wooden wall to let the user know where the Buttons are as soon as the room is loaded. New type of Tubes are added to allow diversity and creativity for the use

4.3.6 Prototype Scripts

Scripts are basically coded instructions (in C# language for preference) that perform actions behind the scenes. There are some scripts that can act independently and some scripts that activates another script and etc. Unity initiate scripts in such a way that they do not need to be attached to any object as long as the programmer knows which object needs to be passed as parameters into the corresponding scripts. The section below will explain the initial prototypes of each script in more details and whether or not they are attached to any objects.

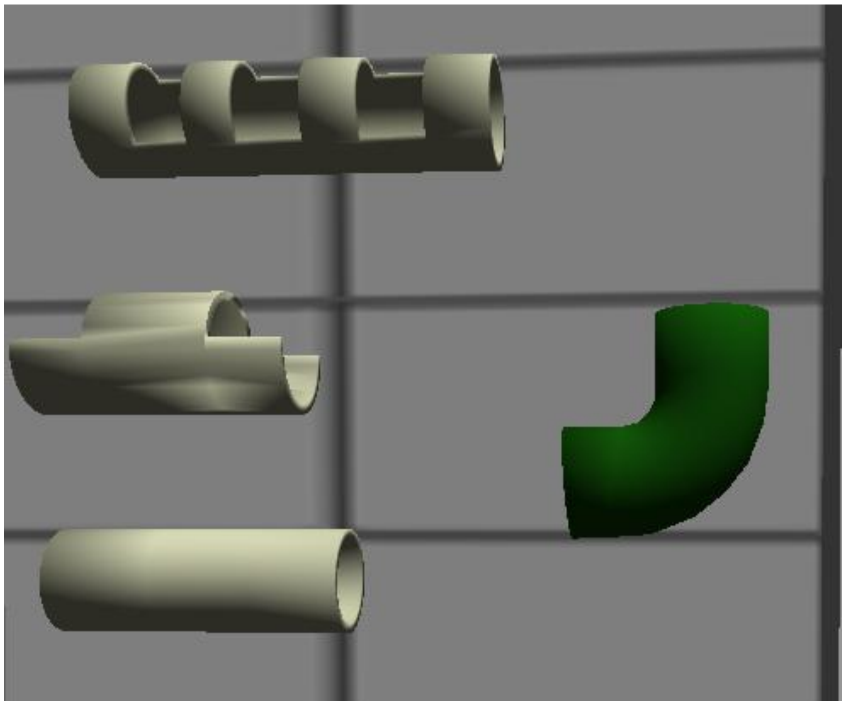
Follow Ball Script
This script is attached to the Ball object. The main purpose of this script is to dynamically follow the Ball object during the Ball Camera Mode until the end of the event.

TurnOn_Off Script
This script is not attached to any object but will be activated by the Drop button event. The purpose of this script is to instantiate a Ball object at a fixed initial starting position at the top of the Magnetic Wall.

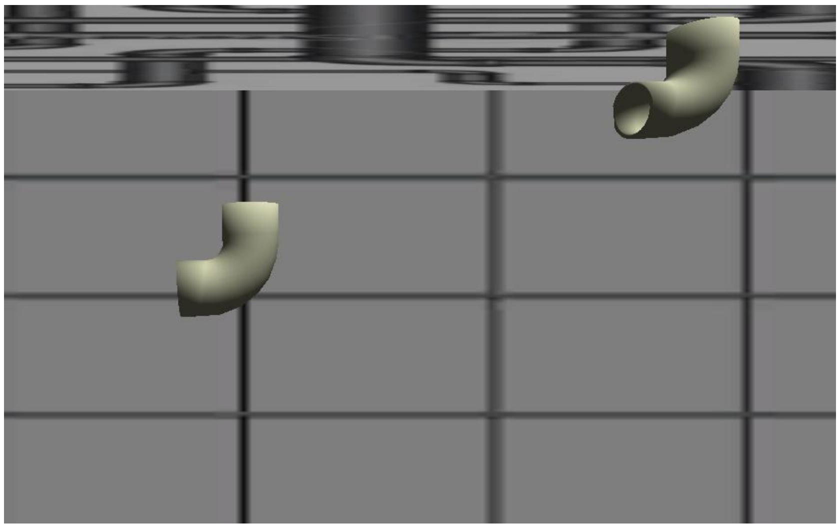
Switch Camera Modes Script
This script is not attached to any object and will handle the cycles of switching between different camera modes (Ball Camera, Player Camera and Tube Camera). By having a script that handles only one functional purpose, it allows organization as well as avoiding any accidental overlapping of Camera objects.

Tube Actions Script

This script is not attached to any objects and will handle everything that is related to the Tube objects. Through this script, user can perform actions such as selecting a Tube object, moving the Tube object along the x-y plane, rotating the Tube object, canceling out of the selected Tube object, and snapping the selected Tube object onto the Magnetic Wall. The Update function will consistently check if any object is in view of the user's screen by following the mouse cursor. If the mouse cursor is hovering over an Tube object, the Update function will call the HighlightTube function to highlight the selected Tube object and then check for further user actions (as seen in the picture below).

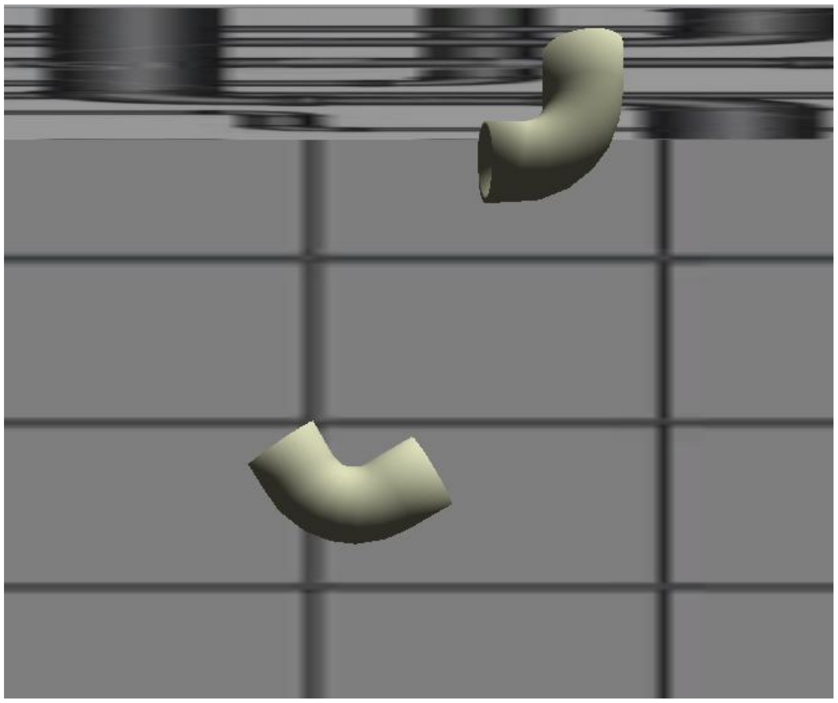


The picture above is missing the mouse cursor, but in the actual footage, the mouse cursor is hovering right over the highlighted green Tube.
If the user's next action is to left-mouse click on the now highlighted Tube object, then the Update function will switch to the Tube Camera Mode and call the GetCurrentTube function to clone the selected Tube object and set it to the clonedTube variable. With the clonedTube now set to true for usability, the script will now switch the Tube Camera Mode from the original highlighted Tube object to the cloned Tube object. The cloned Tube object will be instantiated in the 3D vector position set by createPos values.

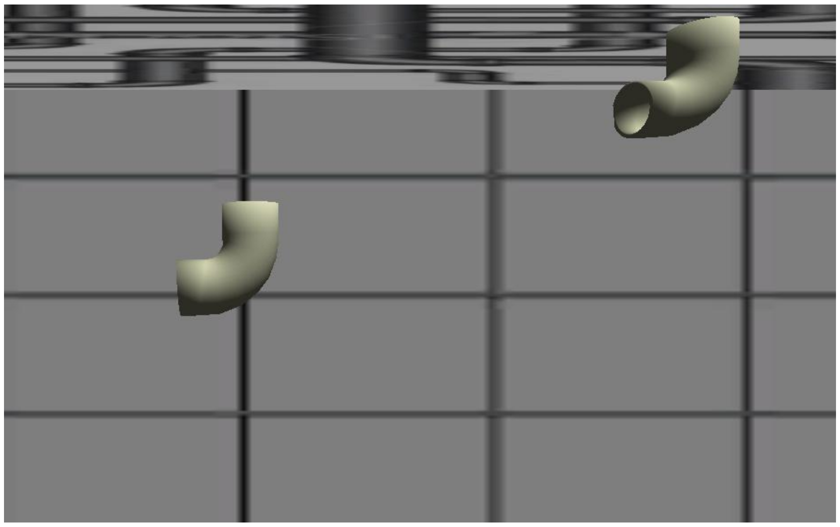


The Tube on the left is the cloned Tube of the highlighted Tube object. The user can now perform new actions (translational and rotational along the x-y plane).

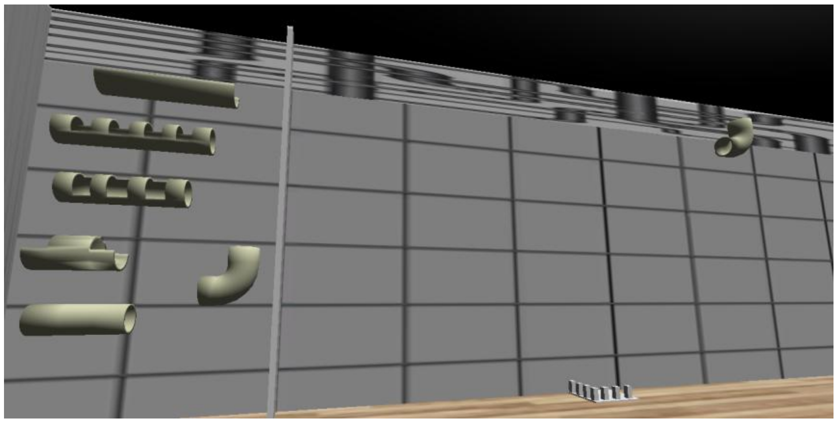
Rotational:



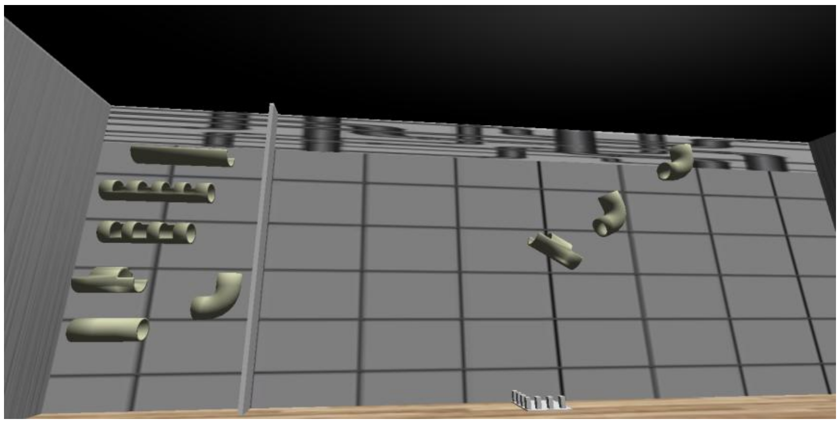
Translational:



If the user does not want to use this cloned Tube object anymore, they can exit from it without leaving it on the Magnetic Wall by simply pressing the ESCAPE key on the keyboard and the Update function will return the camera mode back to the player (in the picture below).



However, if the user wants to keep a Tube on the Magnetic Wall, they can simply press the ENTER key on the keyboard to “snap” the cloned Tube object onto there and the Update function will return back to the Player Camera Mode. However, in the Gravitron Virtual Exhibit version 1.0, the user is unable to interact further with the cloned Tube objects that are already on the Magnetic Wall, since the clonedTube variable is instantiated dynamically to the next highlighted or selected Tube object, thus no longer containing any reference to the previously cloned Tube object.



Reset Button Script
This script is not attached to any object. The main purpose of this script is to reset the Gravitron Virtual Exhibit by clearing any cloned Tubes on the Magnetic Wall.

Drop Button Script
This script is not attached to any object. The main purpose of this script is to instantiate the Ball object at the fixed starting position (right below the Drop Tube object).

Challenge Modes Button Script
This script is not attached to any object. The main purpose of this script is to bring up a Menu that will display a selection of different challenges that the user can click on. Based on the selection, the Gravitron Virtual Exhibit system will then change the layout of the Tubes accordingly.

4.3.7 Current Scripts

Ball Type Button Script
Alternate the Ball object materials and physics each time the user clicks on the Ball Type Button.

Challenge Button Script
Allows the user to select different challenges for the Gravitron, setting the scores based on the mode selected.

Drop Button Script
Create the Ball object in a fixed location on the Magnetic Wall. Also switches the camera to the Ball.

Reset Button Script

Clear (if any) Tubes on the Magnetic Wall.

Ball Class Script
Handles the behavior of the Ball object, such as determining when it has stopped (speed is 0) and return back to the Player Controller Camera.

Setting Variables Script
Containing majority of some general variables and game objects in the Gravitron simulation.

Timer Script
Maintaining the Timer (which determines the score) in the format 00:00, where the first two zeros representing minutes and last two zeros representing seconds.

Tube Manager Script
Maintaining the Tube objects and keep track of the amount and their locations on the Magnetic Wall.

Tube Object Script
Handles the color of selected Tube object depending on the event (e.g. highlighting on Tube is green, able to place on Magnetic Wall is gold). As well as determining what a Tube object can do; rotating, shifting, scaling, etc.

Gravitron Pause Script
Using the generic Pause Menu, the Gravitron Pause script extends to include additional functionalities (such as the ability to view the High School or selecting different Challenges).

4.3.8 Gravitron Pause Menu

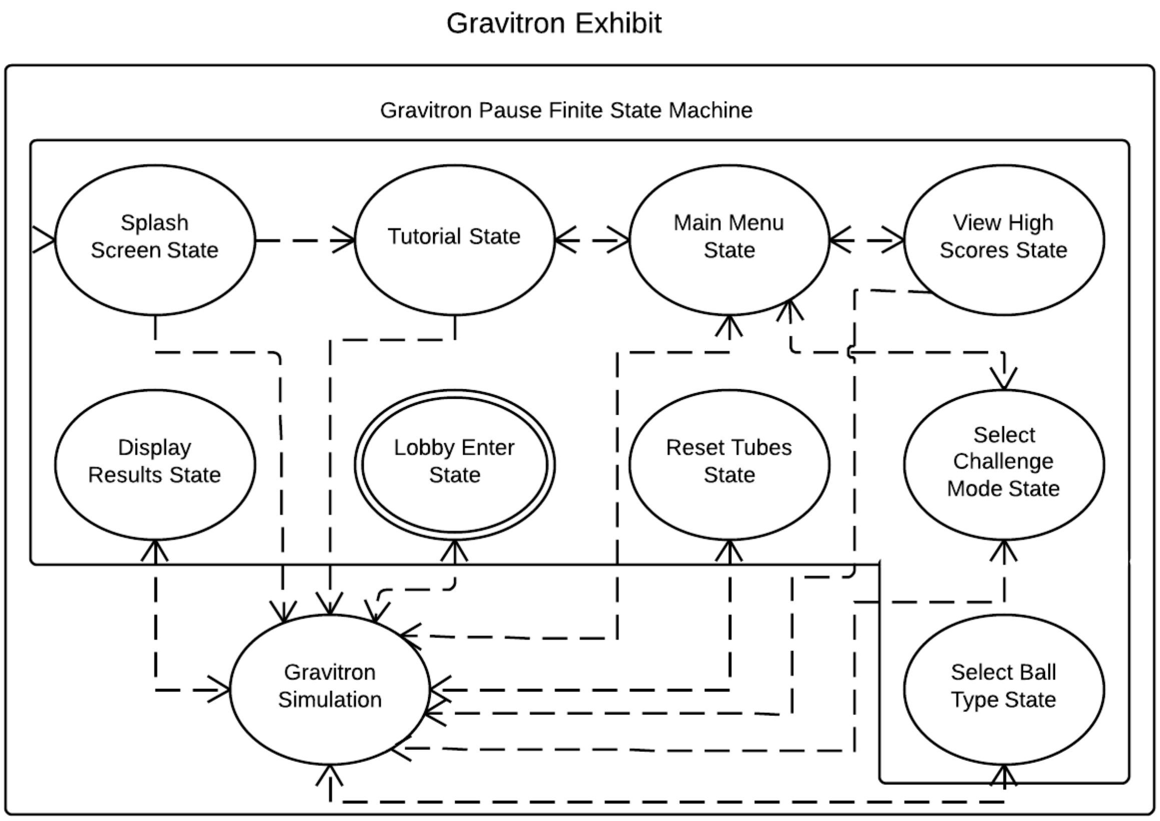
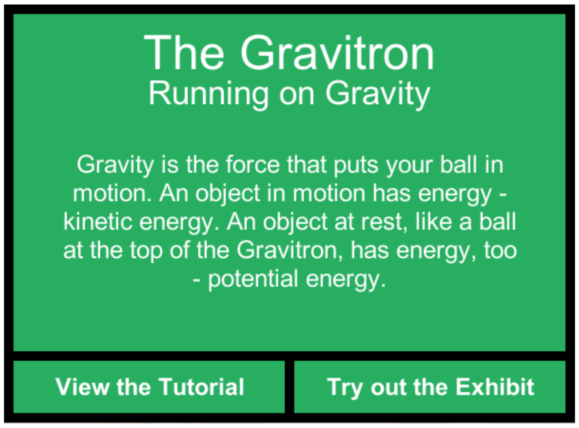


Diagram of the Gravitron Exhibit's GUI Finite State Machine.

The Gravitron Exhibit finite state machine has seven main states that can be transitioned to and an additional state for the Lobby Enter State.

Splash Screen State

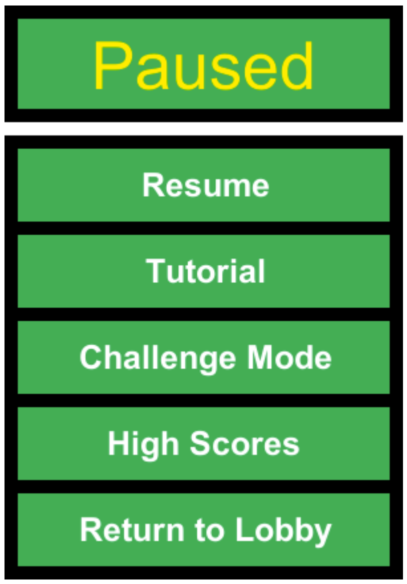


The Gravitron
Running on Gravity

Gravity is the force that puts your ball in motion. An object in motion has energy - kinetic energy. An object at rest, like a ball at the top of the Gravitron, has energy, too - potential energy.

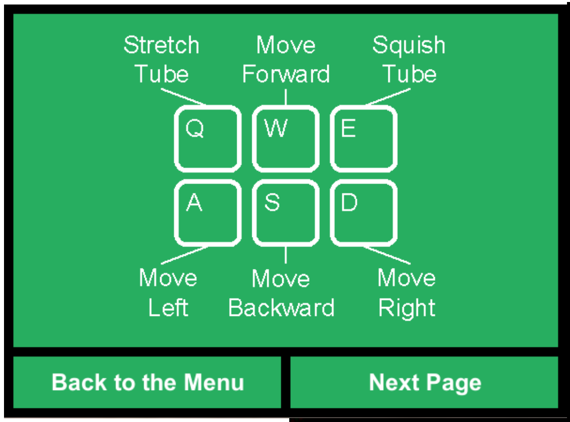
The Splash Screen State is the state that is displayed when the Gravitron Pause Menu is first loaded. This is the only way to reach this state. The Splash Screen State displays a welcome message to the user and gives the user two options: “View the Tutorial” and “Try out the Exhibit”. When the “View the Tutorial” option is selected, the finite state machine transitions to the Tutorial State. When the “Try out the Exhibit” option is selected, the simulation resumes giving the user access to the exhibit.

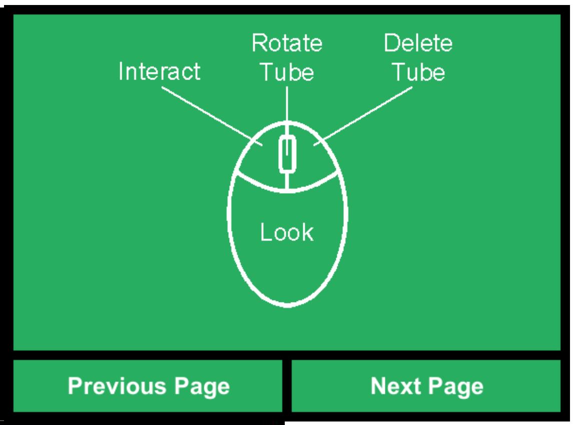
Main Menu State

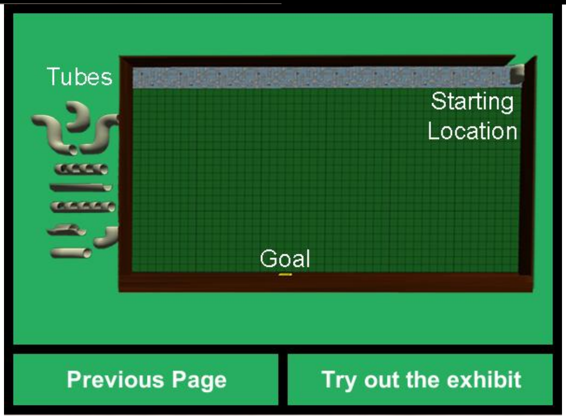


The Main Menu State is the state the default state for the Gravitron Pause Menu and is loaded whenever the simulation is paused. There are five ways to enter the Main Menu State: transition from the Tutorial State, transition from the Select Challenge Mode State, transition from the View High Scores State, transition from the Lobby Enter State, or by pressing the escape key during the simulation. The Main Menu State gives a list of options to the user. This list includes: “Resume”, “Tutorial”, “Challenge Mode”, “High Scores”, and “Return to Lobby”. When the “Resume” option is selected, the simulation resumes. When the “Tutorial” option is selected, the finite state machine transitions to the Tutorial State. When the “Challenge Mode” option is selected, the finite state machine transitions to the Select Challenge Mode State. When the “High Scores” option is selected, the finite state machine transitions to the View High Scores State. When the “Return to Lbby” option is selected, the finite state machine transitions to the Lobby Enter State.

Tutorial State

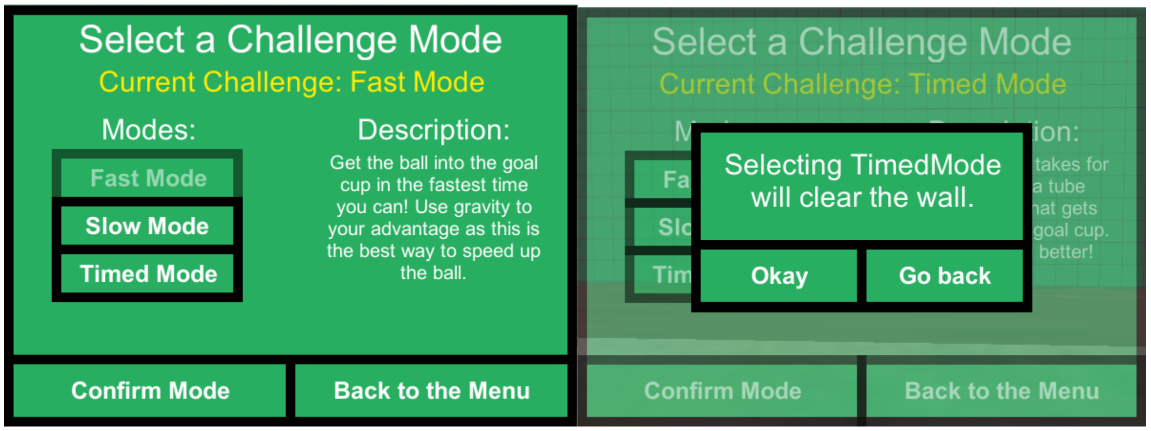






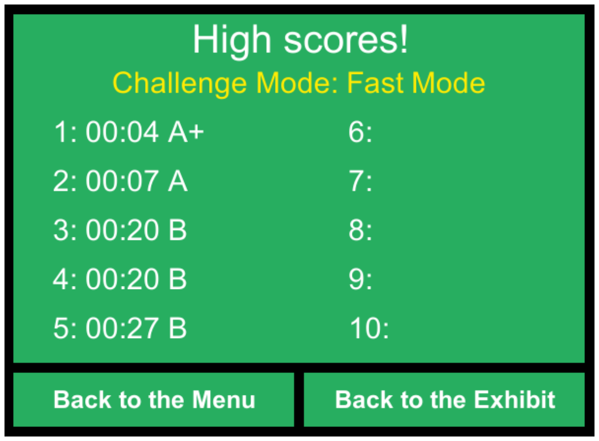
The Tutorial State is a state that displays a list of images that help describe how to move around and interact in the Gravitron Exhibit. The Gravitron Tutorial has three images to inform the user. The first image lists the keys used in the Gravitron Exhibit, which include the “W”, “A”, “S”, and “D” keys for movement and the “Q” and “E” keys for stretching and squishing the currently selected tube. The second image displays how the mouse interacts in the Gravitron Exhibit. Moving the mouse moves the view angle within the exhibit, left clicking will interact within the exhibit, right clicking will delete the currently selected tube, and moving the mouse wheel will rotate the currently selected tube. There are two ways to enter the Tutorial State: transition from the Main Menu State or transition from the Splash Screen State. For each Tutorial page two options are given to the user. These options include “Previous Page” and “Next Page”. If the tutorial page is the first page, the “Previous Page” option is changed with the “Back to the Menu” Option. If the tutorial page is the first page, the “Next Page” option is changed with the “Try out the Exhibit” Option. When the “Back to the Menu” option is selected, the finite state machine transitions to the Main Menu State. When the “Try out the Exhibit” option is selected, the simulation resumes. When the “Next Page” option is selected, the next tutorial image is displayed. When the “Previous Page” option is selected, the previous tutorial image is displayed.

Select Challenge Mode State


Select a Challenge Mode
Current Challenge: Fast Mode
Modes:
Description:
Get the ball into the goal cup in the fastest time you can! Use gravity to your advantage as this is the best way to speed up the ball.
Fast Mode
Slow Mode
Timed Mode
Confirm Mode
Back to the Menu
Select a Challenge Mode
Current Challenge: Timed Mode
Selecting TimedMode will clear the wall.
Okay
Go back
Confirm Mode
Back to the Menu

The Select Challenge Mode State is the state that allows the user to select the current Challenge Mode for the Gravitron Exhibit. The Select Challenge Mode State displays the current Challenge Mode, the Challenge Mode's description, and buttons for each of the other Challenge Modes. There are two ways to enter the Select Challenge Mode State: transition from the Main Menu State or by clicking the Challenge Mode button in the Gravitron Exhibit. The Select Challenge Mode State give the user five options: "Fast Mode", "Slow Mode", "Timed Mode", "Confirm Mode", and "Back to the Menu". When the "Fast Mode" option is selected, the current Challenge Mode is changed to Fast Mode. When the "Slow Mode" option is selected, the current Challenge Mode is changed to Slow Mode. When the "Timed Mode" option is selected, the current Challenge Mode is changed to Timed Mode. When the "Confirm Mode" option is selected, the selected Challenge Mode is applied and the simulation resumes. When the "Back to the Menu" option is selected, the selected Challenge Mode is not applied and the simulation resumes. If the selected mode is Timed Mode and the "Confirm Mode" option is selected, a confirmation box is displayed informing the user that selecting timed mode will clear the wall. Two options are then given to the user: "Okay" and "Go back". When the "Okay" option is selected, the Timed Mode Challenge Mode is applied, the Magnetic Wall is cleared, and the simulation resumes. When the "Go back" option is selected, the confirmation box is removed and the user is allowed to select a Challenge Mode again.

View High Scores State

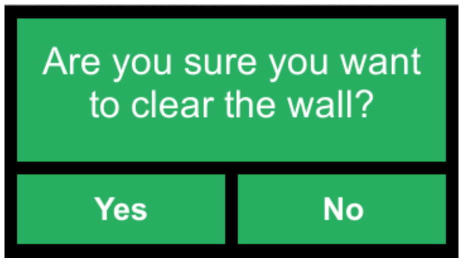

High scores!

Challenge Mode: Fast Mode

```
1: 00:04 A+        6:
2: 00:07 A         7:
3: 00:20 B         8:
4: 00:20 B         9:
5: 00:27 B         10:
```

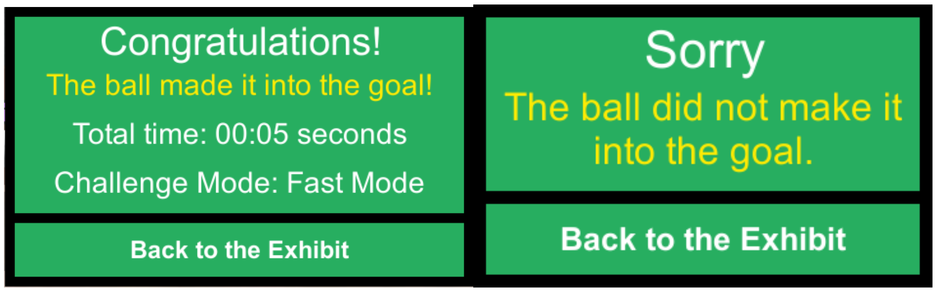
The View High Scores State is the state where the user can view all the high scores saved for each Challenge Mode. The specific high scores displayed is dependent on the current Challenge Mode. The View High Scores State displays the current Challenge Mode and the top 10 scores that the user achieved for that Challenge Mode. There is only one way of transitioning to the View High Scores State and that is from the Main Menu State. The View High Scores State gives the user two options: “Back to the Menu” and “Back to the Exhibit”. When the “Back to the Menu” option is selected, the finite state machine transitions to the Main Menu State. When the “Back to the Exhibit” option is selected, the simulation resumes.

Reset Tubes State


Are you sure you want to clear the wall?

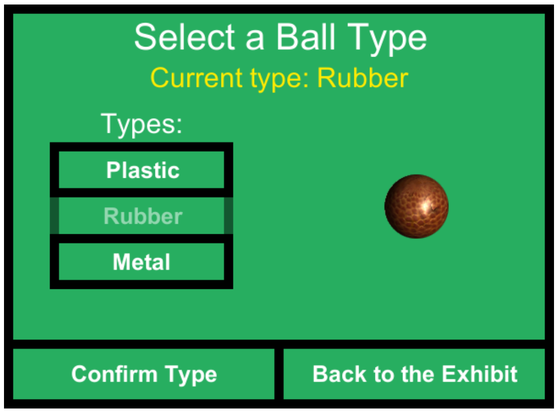
The Reset Tubes State is the state that asks whether the user wants to clear the Magnetic Wall or remove all tubes on the Magnetic Wall. The only way to enter the Reset Tubes State by clicking the Reset Button within the exhibit. The Reset Tubes State gives the user two options: “Yes” to clear the Magnetic Wall, and “No” to keep all the Tube objects on the Magnetic Wall. When the “Yes” option is selected, all the Tube objects on the Magnetic Wall are removed and the simulation resumes. When the “No” option is selected, the simulation resumes, keeping all the Tube objects on the Magnetic Wall.

Display Results State


Congratulations!
The ball made it into the goal!
Total time: 00:05 seconds
Challenge Mode: Fast Mode
Back to the Exhibit
Sorry
The ball did not make it into the goal.
Back to the Exhibit

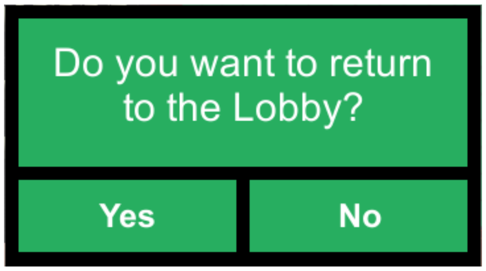
The Display Results State is the state that informs the user of the results of the Ball object dropping through the Tube configuration. There are two types of results that can be displayed: good results for when the Ball object makes it into the Goal Cup object and bad results for when the Ball object stops before reaching the Goal Cup object. The only way to enter the Display Results State is when the Ball object receives the stopped notification. The only option given to the user is “Back to the Exhibit” that when selected will resume the simulation.

Select Ball Type State



The Select Ball Type State is the state that allows the user to select the current Ball Type for the Gravitron Exhibit. The Select Ball Type State displays the current Ball Type, a visual of how the ball will look, and buttons for each of the other Ball Types. The only way to enter the Select Ball Type State by clicking the Ball Type Button within the exhibit. The Select Ball Type State give the user five options: “Plastic”, “Rubber”, “Metal”, “Confirm Type”, and “Back to the exhibit”. When the “Plastic” option is selected, the current Ball Type is changed to Plastic. When the “Rubber” option is selected, the current Ball Type is changed to Rubber. When the “Metal” option is selected, the current Ball Type is changed to Metal. When the “Confirm Type” option is selected, the selected Ball Type is applied and the simulation resumes. When the “Back to the Exhibit” option is selected, the selected Ball Type is not applied and the simulation resumes.

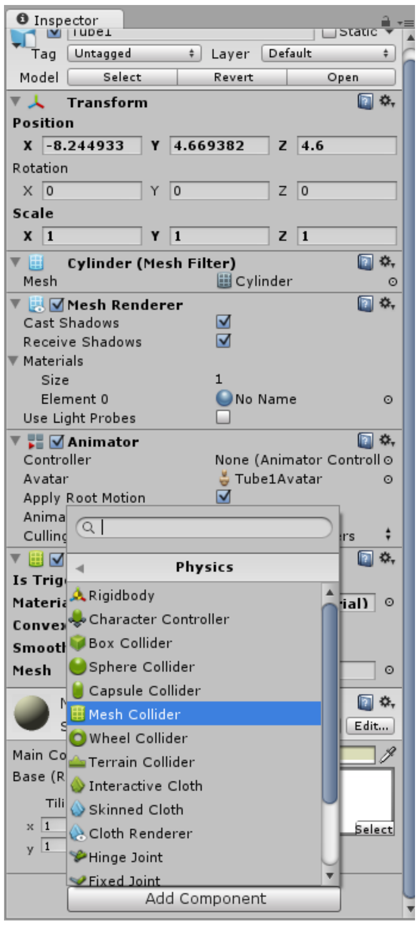
Lobby Enter State



The Lobby Enter State is the state that asks the user if they would like to enter the Lobby Center. There are three ways to enter the Lobby Enter State: clicking on the door placed in the Gravitron Exhibit, by running into the door placed in the Gravitron Exhibit, or by transition from the Main Menu State. The Lobby Enter State gives the user two options: "Yes" to entering the Lobby Center, and "No" to not enter the Lobby Center. When the "Yes" option is selected, the user is then transported to the Lobby Center. When the "No" option is selected, the simulation resumes if the Lobby Enter State was transitioned from the door in the gravitron or the finite state machine will transition to the Main Menu State if transitioned from the Main Menu State.

4.3.9 Objects Collision Checking

Components of the Gravitron Virtual Exhibit such as the Ball object, Floor object and the Wall objects already have a Mesh Collider checkbox that determines whether it is a solid type of material that will not pass through other objects (like the player object). This allows the user to walk on the Floor object without falling through. Components that are created outside of the Unity software, such as in Blender, require a bit more work to achieve the same result. By default, the objects will be just images displayed on the screen. To add the Mesh Collider component to each object created outside of Unity or any objects that do not have it (but require it for collision checking), there is a button in the Inspector window that allows the additional components (as shown in the picture below):


Inspector
Tube1 Static
Tag Untagged Layer Default
Model Select Revert Open
Transform
Position X -8.244933 Y 4.669382 Z 4.6
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Cylinder (Mesh Filter)
Mesh Cylinder
Mesh Renderer
Cast Shadows ✓
Receive Shadows ✓
Materials
Size 1
Element 0 No Name
Use Light Probes
Animator
Controller None (Animator Controll
Avatar Tube1Avatar
Apply Root Motion ✓

Note that there are different types of colliders, the section below will discuss the ones that we will use in the Gravitron Virtual Exhibit along with the advantages and disadvantages of each one.

Mesh Collider

The Mesh Collider provides a collider around an object based on its mesh, thus allowing more precise collision checking on an object if the object is a complex shape.

The advantages of having a mesh collider on an object:

- No need to adjust to the alignment of the object since this option covers each aspect of the object.
- Ensures a very smooth and precise reaction to collisions from other objects.
- Easy to determine where the collision occurs.

The disadvantages of having a mesh collider on an object:

- Extremely expensive to implement, in term of computations required, if the object is complex.
- Cannot have more than one mesh collider per object, thus limited to one.

Sphere Collider

A primitive collider that provides sphere objects to detect collisions much more efficiently, thus this is the best option for spherical objects such as the Ball object. This collider will not work well with objects that are not rounded.

Box Collider

A primitive collider as well that provides a 3 dimensional cube shape that can be shaped into a flattened cube or rectangular prism. It is also scaleable.

The advantages of having a box collider on an object:

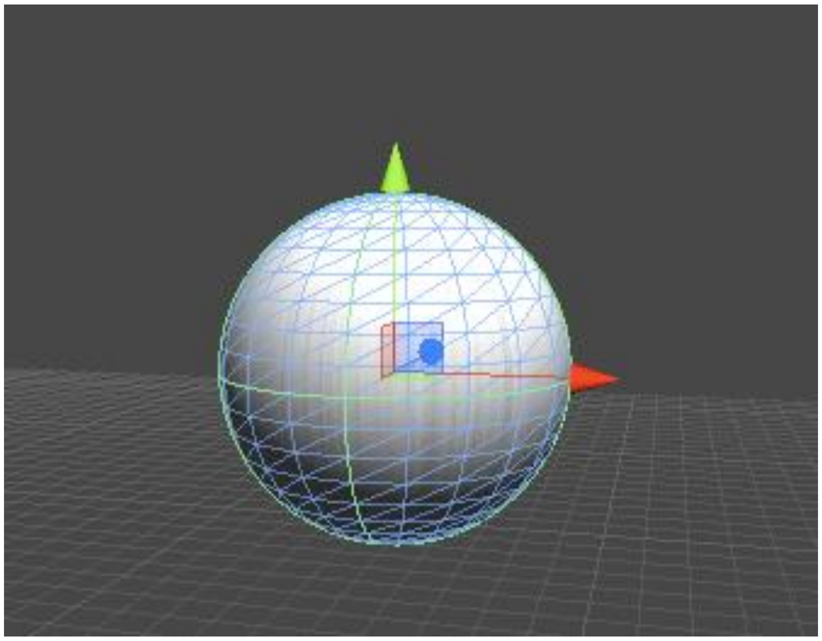
- Cheaper to implement on an object, in terms of computations required.
- For complex shapes, you can have more than one box collider on an object to achieve a similar collision checking and reaction. To do so, each object can only have one type of collider on at a time, create a child object of the current object and add another box collider to that one; to add more box colliders, create another child of the same object. Unity will treat all the box colliders in the child objects as the same collision check with the parent object.
- Easy to utilize

The disadvantages of having a box collider on an object:

- For complex shapes, can also be difficult to implement. Having too many box colliders on an object can cause some confusion.
- Requires more manual manipulation of the collider than the mesh collider option.
- Have to be as precise as possible for the correct physics response, otherwise the collisions with other objects will not have the same result as in real world physics.

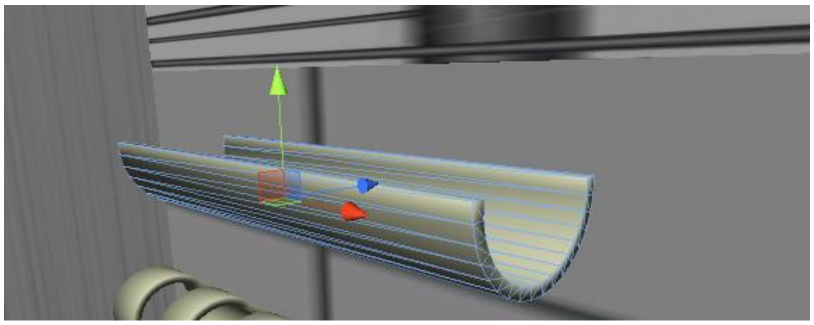
Collision Checking (Ball Object)

As discussed earlier, the Ball object is just a sphere created in Unity, thus the best option is to have the sphere collider component on it. The picture below will represent the sphere collider outlined onto the Ball object, each line will be able to detect collisions from any objects interacting with the Ball object:



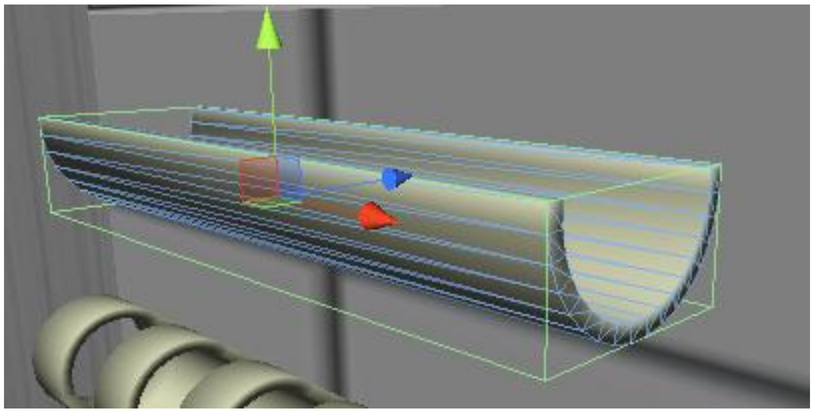
Collision Checking (Tube Objects)

Since the Tube objects can only move and rotate in the x-y plane, it only makes sense that the best option would be to have the box collider component on instead of the mesh collider. But to show the difference between the mesh collider and the box collider, the picture below will explain the reason as to why the mesh collider is much more expensive and the box collider is the better alternative:

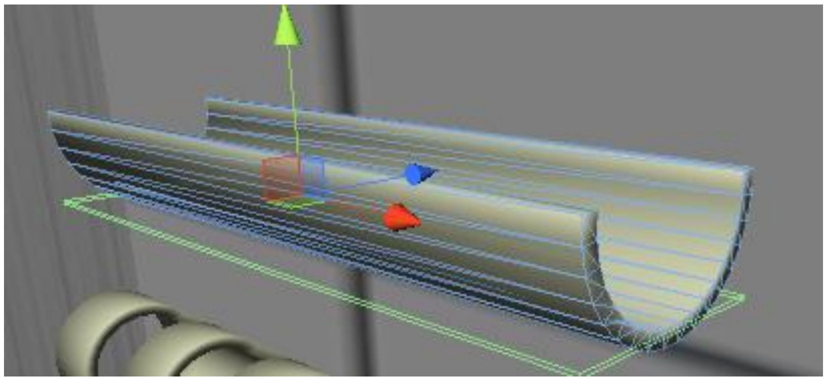


(Picture above) - Notice how the outlines are covering the entire mesh of the object, thus providing very precise collision checking but it also requires a lot more computation.

Now to compare the mesh collider with the box collider component on the same Tube object:



The box collider covers the whole Tube object in a rectangular prism. However, we only need the collision checking to be at the bottom surface of the Tube object rather than the whole object else the Ball object will roll on top of the open Tube (which would be very awkward). Unity allows us to adjust the box collider accordingly as shown in the picture below:



This method will be sufficient enough for the purpose of the Tube objects. And if there are scenarios that requires the system to check for the Ball object if it collides on the top portion of the Tube object, then we can simply create a new empty game object and set it as the child of the collided Tube object (parent) and add another box collider to that child object, in which case, Unity would then count those two (child and parent) as a whole object for a collision checking. Reason for that is because Unity does not allow the same object to contain the same component (meaning a Tube object cannot have two box colliders at once) , but Unity allows us to override it if we create a separate entity that react the same way as the parent object and give it the same collider, the collisions will be counted as one object. However, as mentioned above, the mesh collider cannot collide with another mesh collider even if the component is on a separate entity, that is why Unity only allows one mesh collider per object as a whole.

4.3.10 Development Risks

There will be some bugs or glitches that the user might come across if certain conditions are not met properly. Here is a list of some known situations that will be resolved before the final product as well as a list of possible risk situations corresponding to the operational scenarios.

Operational Scenarios (Tube Objects)
There are different lengths of Tubes, but they are all the same size and diameter for the Ball object to pass through successfully. Because of the different lengths, there are certain conditions in which the user may be able to place a Tube through another Tube on the Magnetic Wall. This type of collision will undoubtedly cause issues during execution time. The Tube collision risk is more of an annoyance to the user than the system, since the Ball will most likely get stuck between the Tubes that are colliding and the user would have to reset the room and redo their design. There are three possible solutions in mind:

> Tube Solution 1 (Object Collision Checking)
> *No Tube objects can pass through one another. The advantage of this method is that it ensures the user cannot move a Tube object through another Tube object that is in the way, instead they have to move around it. The disadvantage is that it makes the user spend more time adjusting the Tube object and can cause aggravation (and possibly quit the Gravitron Virtual Exhibit altogether!).*
>
> Tube Solution 2 (Object Invalid Action Checking)
> *Each Tube object will check if it's colliding with another Tube object. The advantage of this method is that it informs the user if two or more Tubes are colliding with each other. If the tubes are colliding, the system will not allow the user to set the selected Tube in that position on the Magnetic Wall. But, they are still able to move through the Tube that is in the way to get to the other side rather than going around it. The disadvantage is that it will cost more in terms of processing time to check for this larger set of collision points.*
>
> Tube Solution 3 (User Self-Correctness Action)
> *The user will change the Tubes' positions themselves if the Ball object is caught between two Tubes and is unable to move further. A message will pop up to inform the user that the Ball cannot move due to a Tube collision. If the Ball is not moving for certain amount of time (in seconds), the Tube Camera Mode will then simply return back to the Player Camera Mode and the Ball object is then deleted until the user presses the Drop Button again. The advantage of this method is*

that it doesn’t cost any additional computing time to continually check if the Tubes are colliding. The disadvantage is that it depends on whether the user will see the issue and be able to correct it. The message displayed to the user should be precise and informative on the next step of action to ensure that the user understands the situation.

Operational Scenarios (Ball Object)

The Ball object is associated with the Ball Camera Mode and contains a rigid body that gives it realistic physical properties which are affected by the simulated gravity of the system. However, there are some additional scripts that can cause an infinite loop when checking for the Ball’s position in the room. The main purpose of the Ball object is to either check itself if it lands in the Goal Cup or if it comes into direct contact with the Gutter or Floor object. If either of the statements are true, then the Ball Camera Mode will switch back to the Player Camera Mode. If the returned values of the Goal Cup object and the Gutter or Floor object are both false, then this creates an infinite loop in which the user is stuck in the Ball Camera Mode.

Solution 1 (Goal-Cup Checking by Y-Axis)

A possible method of only checking whether the Ball is in collision with respect to the Goal Cup object.

- *If the Ball object is encased in the Goal Cup object, then return back to the Player Camera Mode with a message informing the user of the successful attempt.*
- *Else check if the Ball object has a y-value less than the y-value of the Goal Cup object (meaning that the Ball is lower than the Goal Cup). If it does, then that means the Ball will definitely not able to reach the objective anymore. Thus return back to the Player Camera Mode with a message informing the user of the failed attempt.*
- *Else if the Ball object does not have a y-value less than the Goal Cup object, then that assumes the Ball is still going through its course has a chance of landing in the Goal Cup. This condition will continue to let the user stay in the Ball Camera Mode.*

The advantage of this method is that it ensures a simple collision checking algorithm that only checks in the y-axis value. The disadvantage is that, in situations such as when the Ball object is stuck somewhere on the Magnetic Wall that is still above the Goal Cup object in the y-axis, then the Ball object will remain in the Ball Camera Mode infinitely as well. Another disadvantage in this method is that the user might intentionally place a Tube somewhere on the y-axis value that is below the Goal Cup object, and the system will still check that as a *failed attempt (even if the user arranges the Tubes in such a way to utilize momentum to roll the Ball object back up in the positive y-axis direction).*

Solution 2 (Time Checking Method)
This solution involves checking whether the Ball object is inactive for more than 5 seconds, if so then return back to the Player Camera Mode. This method derives from one of the requirements in which the system will consistently check the Ball object to return false if no movement (change in axis values in the x-y plane) is detected. The advantage of this method is that the system will only check the Ball object and the Goal Cup object. The disadvantage is that in certain cases the user may want the Ball to move as slowly as possible and the system might accidentally register the slow movement as inactivity.

Operational Scenarios (Button Objects)
The three main functional buttons in the Gravitron Virtual Exhibit are the Reset Button, the Drop Button, and the Challenge Modes Button. The function of the Reset Button is to reset the room if the user is not pleased with their tube layout. The function of the Drop Button is to initiate the test of the implemented design layout by creating the Ball object and dropping it from the top of the Magnetic Wall, where the fixed starting point will be. Another important aspect of the Button objects is that they implicitly switch between different camera modes. For example, the Drop Button will disable the Player Camera Mode and then turn on the Ball Camera Mode after instantiating the Ball object in the starting position. Thus, if the Button objects' scripts are not called correctly then user will experience strange behavior from the system.

Solution 1 (One-Script Per Call)
This method will only allow one script to be called when the user clicks on a Button object, thus ensuring no other script will be able to change the camera modes until the corresponding event is finished. It is a solid method that guarantees the correct return values in case the user accidentally stumbled onto some glitch that allows them to turn on the other camera modes in addition to the current one. The advantage of this method is that it prevents any accidental overlapping of the different camera modes, and helps to minimize the chance of users maliciously impacting the system. The disadvantage is that if the camera mode is returned to the wrong perspective, then the user can't do anything about it until the corresponding event is complete, but even so it can get stuck in that event infinitely.

Solution 2 (Solution 1 + ESCAPE Key)

Solution 2 is the same implementation as Solution 1 with the addition of adding an ESCAPE key when calling the script that the Button (which is left-mouse clicked by the user) is associated with. This method would involve adding a message at the top left corner of the camera screen which will inform the user that they can exit the current event at any time by simply pressing the ESCAPE key. The advantage of this method is that it will still prevent any accidental overlapping of the different camera modes as well as unintentional user actions that may cause an infinite loop in the Gravitron Virtual Exhibit system. The disadvantage is that this method relies on the users' action if they are caught in an infinite loop or the wrong camera mode, but the user might or might not be aware of which camera mode they are supposed to be in if it is their first time utilizing this virtual exhibit experience.

Solution 3 (A Button At A Time)
Even if only one script can be active at a time, there can be a scenario in which the user can accidentally press another Button which turns off the current script while calling the script associated with the other Button object. This method will ensure that only one Button can be activated at a time, preventing accidental script change. Another true or false mode will be set to the Button objects; the purpose for this mode is to turn off the other Button objects if the current Button is set to true when the user clicks on it. Not only will this method prevent multiple Buttons activating, it will also implicitly prevent multiple scripts initiating as well, thus avoiding the issue of scripts overlapping. So, essentially, once a Button object is disabled, its associated scripts are also set to false (or rather, can't be called) since the system cannot activate a script in an object that is inactive. The advantage of this method is that it provides the same security as the previous solutions to the Button objects but done in a way that disables the other Button objects. The disadvantage of this method being that, what if the user pressed the wrong Button object than their intended one? They either have to restart the Gravitron Virtual Exhibit room or go through the event of the corresponding Button object that they have selected (which may or may not be stuck in an infinite loop).

Operational Scenarios (Timer Objects)
The Timer object is a rectangular box displaying the starting time at 0:00 s (seconds) to the end time (x amount of seconds), where x will be the time in seconds that is frozen when the Drop Button event is complete. The Timer object is completely reliant on the Drop Button event, thus in scenarios that might cause another event to start during the Drop Button scene, the Timer will never stop counting and will give the user the wrong score. However, if Solution 3 for the Button objects is implemented, then the system

won't have to check whether the Timer object is running infinitely. The main purpose of the Timer object is to keep track of the amount of time (in seconds) the Ball object will take to get from the initial starting position to the Goal Cup object or the Gutter or Floor object. The Timer object is only activated during the start of the Ball Camera Mode and ends when the Ball Camera Mode is turned off implicitly. Thus, to ensure the Timer object will reset correctly and store only the successful attempts, the Ball Camera Mode will automatically disable the Timer object before switching to the other camera modes by priority.

4.3.11 Analysis

The Gravitron Virtual Exhibit will go through many versions before the final product, but even so, the final product can still undergo changes if the OSC Sponsors want to modify this virtual exhibit in the future (after us that is). There are some improvements that we would like to implement in the near future that would provide a better learning experience and convey the objective of the Gravitron Exhibit more clearly.

Expected Improvement (Gravitron Virtual Exhibit Room Layout)
Adjust the virtual exhibit room layout to provide a better visual representation since the current design is plain in terms of color and space. Environmental aspect do get taken into account for visitors' interest, so we need to make it visually appealing as well as a functional exhibit to attract variety of audiences.

Disadvantages
The desire for better quality of the environmental objects come with the cost of more computing memory space and can contradict with one of our requirements. Higher quality doesn't necessary mean better performance since it takes a lot more time to compute other parts of the virtual exhibit.

Limitations
One of the system's requirement is that the OSC Virtual Exhibits Hall as a whole, should attempt to be less than 50 megabytes. Our first priority is to make sure the exhibits are fully functional thus having a higher quality of the environment will be very constraint, depending on how much memory space we have left to work with.

Risks
There are few risks that arise when we improved the environmental aspect of the Gravitron Virtual Exhibit:

- *Possibility of going over the maximum memory capacity in our requirements.*
- *Possibility of overly-complex shapes or textures that might be hard to work with.*
- *Possibility of too distracting to the user to pay attention to the functional parts of the Gravitron Virtual Exhibit (though it can be a good thing if the user becomes distracted by the environment in a good way).*
- *Possibility of preference of consistency, meaning that all the sections of the OSC Virtual Exhibits Hall would have to be in the same quality in terms of environmental aspect, thus costing a lot more memory space than expected.*

Alternatives and Tradeoffs
Another method to the Gravitron Virtual Exhibit room layout is to just take out the surrounding Walls that have nothing to do with the main functional parts (which consists of the Tube objects, Ball object, Coal Cup object, Timer object, Button objects, Exit Door object, and the Magnetic Wall object). The user will only have a visual contact with the Magnetic Wall object and by it, the Tube objects. The surrounding environment will be the endless sky encasing the whole room, to make it feel like an outDoor perspective. The benefit of this type of method is that it can create a lot more open space to the user rather than a small enclosed room. Not to mention saving some extra computing memory. The only downside is that the user will not have boundary Walls to prevent them from going off the Floor object and falling into the abyss, but an easy solution to that is to make an invisible checking system on each edge of the Floor object and prevents the user from going past the edges.

Expected Improvement (Tube Interactions)
The translational and rotational commands on each individual Tube objects will be better improved to respond more efficiently and provide smoother transitions. At the current state, the Tube objects are painfully slow to move around and rotate on the x-y plane along the Magnetic Wall object. The future implementation will change the scripts in the main Tube object script. By adjusting the numbers on the x and y axis values, as well as the rotational speed, the Tube objects can move and rotate faster. However, there is still the issue regarding the keyboard command. The current implementation of the keyboard command only allows one action at a time when pressing a key; meaning that holding down a rotational key command on the keyboard will not continuously rotate the Tube in the corresponding direction, rather it only moves once by the initial rotational key that is pressed. Thus, the expected improvement on that is to change the script to read in keyboard commands to be continuously activated while a key is

pressed and not yet released on the keyboard. The keyboard commands are only one of the two choices the user can have in moving the Tubes around. The ability to have the Tubes follow the mouse cursor around on the Magnetic Wall only, will provide a better translational movement for the user.

Disadvantages
It is more difficult and complex to respond correctly to a key pressed and held down than it is to simply read a key pressed on the keyboard once. Reason for that is because it would require us to manually add in a specific key command to do actions such as rotating an object.

Limitations
There are no known limitations so far regarding the implementation of the new key command improvement. However, the knowledge of how to get it to work is a partial limitation at this stage.

Risks
No risks associated with this method as the key command is already defined by Unity, we just have to make it work for the Gravitron Virtual Exhibit system.

Alternatives and Tradeoffs
There's no other alternatives as Unity doesn't have other options in reading in key commands.

Expected Improvement (Camera Control)
As mentioned earlier regarding the three Camera objects that we have in this exhibit, we need to also improve the quality of the camera perspectives. The two Camera objects that need to be improved are the Ball Camera object and the Tube Camera object, since these two Cameras are functioned dynamically. However, we only need to come up with one algorithm rather than two separate algorithms as the functioning parts of the camera objects should work in a similar manner. How the Camera objects dynamically follows the corresponding objects is by changing the x and y axis values while the z axis value remains the same. Thus the user cannot perform actions like zooming in and out as that requires the z values to be adjustable as well. The future implementation for the two Camera objects will be to have the zooming in and out feature, but with certain constraint from the user zooming too far or too close.

Disadvantages

There are no disadvantages to adding in the zooming in and out feature as it helps the user to see the overall design layout than to only see their objects in a limited visual surroundings.

Limitations
The only limitation is understanding how to implement the zooming in and out feature into the camera object scripts.

Risks
No known risks at this moment.

Alternatives and Tradeoffs
Another method for a better camera quality control is to provide a separate key command to allow the user to see the Magnetic Wall object entirely with a new camera object and implement the ESCAPE key command to return back to their corresponding camera mode when they wish to exit the new camera object mode. The tradeoff is that it require less steps in achieving this method as well as for the user to see their design layout quicker.

4.4 Final Assessment

For the final product we presented to our Senior Design committee, our Gravitron exhibit fully satisfied all functional and non-functional requirements, except functional requirement 1.

Functional requirement 1 was mostly satisfied because the major elements from the exhibit are included. The tubes, balls, magnetic wall, and challenge modes are all present in the simulation. However, a difference between the real life version and our simulation, is that users cannot construct their design across both walls. We felt this feature deserved a lower priority than the core functionality of properly building a layout and simulating the ball. Several users did request the feature to build on both walls while we demonstrated the simulation at Otronicon but other issues took priority.

5. Pinewood Derby Exhibit

5.1 Introduction

The Pinewood Derby Exhibit at the Orlando Science Center allows visitors to construct derby cars and race them against each other. The exhibit teaches visitors about the relationship between gravity, energy, and motion and how those principles can be applied to construct a winning derby car.

Time at the exhibit is limited, so visitors can only choose from pre-made parts to construct their derby car. A standard chassis is used as the starting point for each car, and from there visitors can choose which wheels, axles and weights to place on their car. Axles can be positioned on the car in a few different places, allowing for an adjustable wheelbase. Wheels come in three different sizes, and different sizes can be used for the front and back wheels. Finally, weights are provided to adjust the center of mass of the car, or to simply make the car heavier (up to a given maximum weight).

Visitors are allowed two runs each to determine what makes the fastest derby car. Afterwards, a score will be tallied together from all the attempts and compared with the other competitors to determine whose derby car is the fastest. The Pinewood Derby exhibit is a great opportunity for visitors to learn about the physics of gravity and how they can be used to design a faster derby car--the satisfaction of winning is part of the fun!

5.2 Pinewood Derby High Level Requirements

5.2.1 Introduction and Goals

The online exhibit for the Pinewood Derby will aim to have the same design challenges and provide a similar exposure of physics at work as the real exhibit. However, some aspects of the online version will differ from the real exhibit, largely in terms of creative freedom with track layouts and car design. Instead of manually assembling a derby car, users will pick and choose the parts that make up the car, which the simulation will then automatically install.

To keep things simple, users must still choose from a set list of parts, but the online exhibit can offer a greater variety of parts and a more variable placement of those parts. The second main difference is in providing a greater variety of derby tracks than the single track offered at the Orlando Science Center, including tracks that would be longer or contain a variety of features, such as slopes, hills, and loops. Each track will offer a new scenario in which users can see the effects of gravity on their derby car, and experiment to find the best derby car setup for each track. This is a major benefit over the real life exhibit, as physical derby tracks are very large, inflexible, and expensive to build.

5.2.2 High Level Requirements and Features

1. Functional:

1. The exhibit shall take place in its own room, which shows the track and table of parts.
2. The exhibit shall instruct the user on how to assemble their car from the list of parts, including any design limitations.
3. The exhibit shall allow the user to construct a derby car from a given list of parts.
4. A standard chassis shall be provided for all derby cars, with at least one given shape.
5. At least three varieties of wheels shall be provided, the primary difference being the diameter of each wheel.
6. Each kind of wheel shall be set per axle, meaning that each axle must have the same kind of wheel on each end, but the wheels on the front axle may differ from those on the back.
7. Axles can be placed on the front and rear ends of the chassis, with their position adjustable from front to back, but not so much that it would cause the wheels to intersect.
8. Different sized weights shall be available to place on various points on the chassis.
9. The user’s car shall be within a given weight limit. If the car weighs more than this, the program should not allow the user to continue, and instruct the user to modify their car such that it does not exceed the limit.
10. At least one track shall be provided which will emulate the track at OSC.
11. The exhibit shall implement a reasonably accurate physical simulation of a running derby car, taking into account the type and placement of axles, wheels, and weights on the chassis.
12. Proper collision detection of the car against the track shall be implemented; the car should not fall through the track.

13. If the car stops moving or is otherwise unable to complete the track, the simulation shall end.
14. The time it takes for the car to complete the course shall be recorded, and displayed properly to the user.
15. Times for a few of the previous runs shall be kept for the user to compare, in order to track the user's progress in finding the fastest car. These times do not need to be stored for later visits to the exhibit, only until the user leaves the exhibit.
16. The system will record the position the cars finish in (1st, 2nd, 3rd, etc)

2. Non Functional:

1. The system shall play appropriate sound effects when certain events occur (such as footstep sounds while walking).
2. The system shall provide appropriate textures for each object that requires them.
3. Cars shall look similar to the real life ones at the OSC.

3. Desired Features

1. A variety of chassis should be provided, with different shapes leading to different aerodynamic properties.
2. Customizable appearance of the chassis, such as customizable colors.
3. Wheels with properties other than size, such as grip or different tread widths.
4. Particle effects for the derby car as it runs (effects such as fire, wind, etc).
5. Cool track changing animations; tracks lower into the floor and the new one raises up.
6. Multiple tracks of varying shapes

5.2.3 Exhibit Components

This section gives a listing and brief description of all the major components of the Pinewood Derby exhibit. These represent either actual objects inside the simulation or components needed to program the exhibit's core functionality.

Chassis

The chassis is the base of the derby car. All weights and axles will be attached directly to the chassis. The chassis is adjustable for length, and comes in three fixed sizes. Each size has a different weight. Axles are placed at fixed locations near the end of the chassis. The chassis must not be small enough such that the wheels do not collide when the smallest chassis is chosen.

Wheels

These attach to axles on each end of the chassis. The wheels can be different sizes on each end, but must be the same size on each axle. Larger wheels have greater mass than smaller wheels.

Axles

Axles control the placement of wheels on the derby car. There are two axles per car, one in the front and back. The placement of these axles can vary, in order to adjust the wheelbase of the car. The axles cannot be placed close enough to each other such that their wheels would intersect.

Weights

Weights can be attached to a few places on the chassis. Weights are only used to increase the weight of the derby car and/or to shift the center of mass of the car. If attaching a weight causes the car to go over the weight limit, then the weight cannot be attached.

Track

At least one track will be provided for the user to test their derby car. Each should be designed in such a way that a different solution will be required to find the fastest car. There will only be one empty slot on each track since there will only be one user in the simulation at a time. The user’s assembled car will run in this empty slot.

Finish Line

At the end of each track should be a finish line. When the user’s derby car passes this line, the timer should stop and the time for the run should be displayed, along with some of the user’s previous times. The run is considered finished when the front of the car first crosses the finish line. When a car crosses the finish line, the scoreboard will display the position it finished in (1st place, 2nd place, etc).

Reset Button

If at any time the user wants to stop the derby car simulation, the user can click on a reset button located at the top of the screen. This will reset the cars to the top of the track and the recorded values on the scoreboard.

Sound (Non-functional component)

In the Pinewood Derby Virtual Exhibit, several sound effects will be used to further immerse the player in the exhibit.

The following events will trigger certain sound effects:

- Car finish - crowd cheering

5.3 System Design

5.3.1 Operational Scenarios

When a user enters the exhibit room, they will be presented with a dialog box that gives a brief description of the Pinewood Derby. They will be given the option to either start the tutorial or explore the exhibit.

If the user chooses to explore the exhibit, they will be able to move freely around the room. They should see the track in front of them with stairs leading up to the top where the workbench is located. Eventually they should make their way over to the rear of the track and up the ramp. After going up the ramp, they should see the workbench and the top of the track. Before racing, the user will have to construct their car at the workbench.

At the workbench, the user will be asked to select a body type for their car. Next they will be asked to select an axel type for the car. Next, they will be asked to select a wheel type for the car. Next, they will be asked to optionally add weights to their car. Finally, they will be moved to the scale. At any point in the process, the user will be allowed to move to a previous or next workbench step. If the car is too heavy when weighed, they will be forced to make the car lighter before racing. Once the car is at or under the weight limit, they will be asked if they wish to continue editing or proceed to the track.

Once the user selects to proceed to the track, the system will copy the car into the empty track slot. The user will be free to press the start button on the track to release their car or they can continue to explore the room.

When the user presses the start button on the track, the user’s car and the NPC cars will all be given a slight initial acceleration. The cars will make their way down the track and once all cars have crossed the finish line or after the timer runs out, the time the cars took to reach the finish line will be displayed in order from shortest to longest in a message box.

The user can then close the message box. From there they can edit their car, run the race again, or go back to the lobby.

If the user wishes to run the race again, they will need to press the reset button to return the cars to the top of the track and then press the start button. The user can also return to the workbench and they will be presented with their current car configuration which they can modify as needed.

5.3.2 Camera Functions and Modes

There are three main cameras in the Pinewood Derby. The Player camera, Workbench camera, and Follow Car camera.

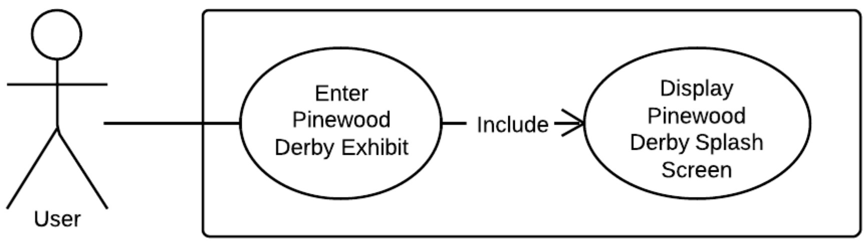
Player
This camera view is from the player's point of view. It moves with the player, and the direction it faces is controllable with the mouse. The player can look up and down and side to side.

Follow Car
This camera is located slightly behind and above the car. This provides a view that is commonly found in racing video games.

5.3.3 Pinewood Derby Use Cases

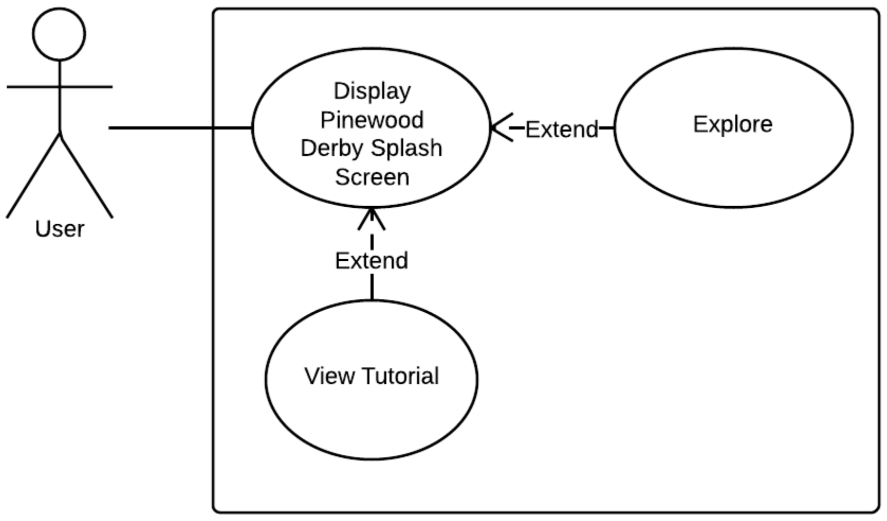
UC38. Enter Pinewood Derby Exhibit – Upon clicking on the Pinewood Derby Exhibit Door within the Lobby Center, the user is then taken to the Pinewood Derby Exhibit Room. From this room, the Pinewood Derby Splash Screen is then displayed. See Use Case 38 below.

Enter Pinewood Derby Exhibit (UC38)


User
Enter Pinewood Derby Exhibit
Include
Display Pinewood Derby Splash Screen

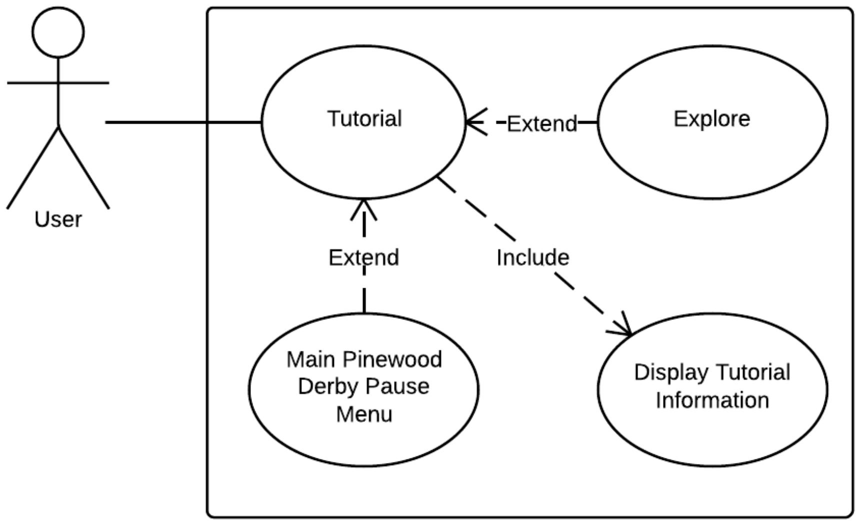
UC39. Pinewood Derby Splash Screen – Information on the Pinewood Derby Exhibit is displayed on the Splash Screen. Information includes a welcome message, what the Pinewood Derby Exhibit is about, and instructions on what to do next. The user will also be given two options: "Explore the Pinewood Derby Exhibit" and "View tutorial". Each of these options are selectable and will display a different interface upon selection. See Use Case 39 below.

Pinewood Derby Splash Screen (UC39)



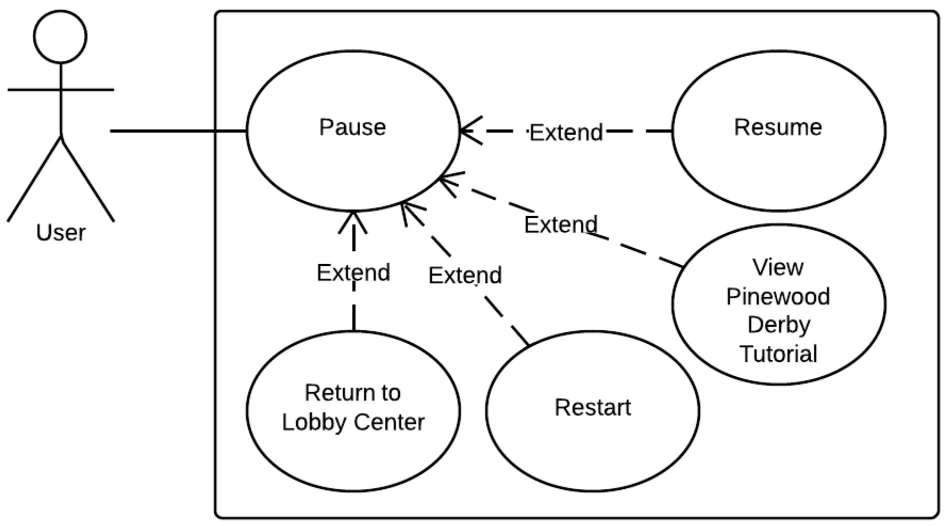
UC40. Pinewood Derby Tutorial – When the "View Tutorial" button is selected from the Pinewood Derby Exhibit Splash Screen or from the Main Pinewood Derby Pause Menu, a new interface is brought up with information on how to use the Pinewood Derby Exhibit. Information will include controls for within the exhibit and what to do next. The Pinewood Derby Tutorial gives the user two options: "Explore" and "Main Menu". See Use Case Diagram 40 below.

Pinewood Derby Tutorial (UC40)



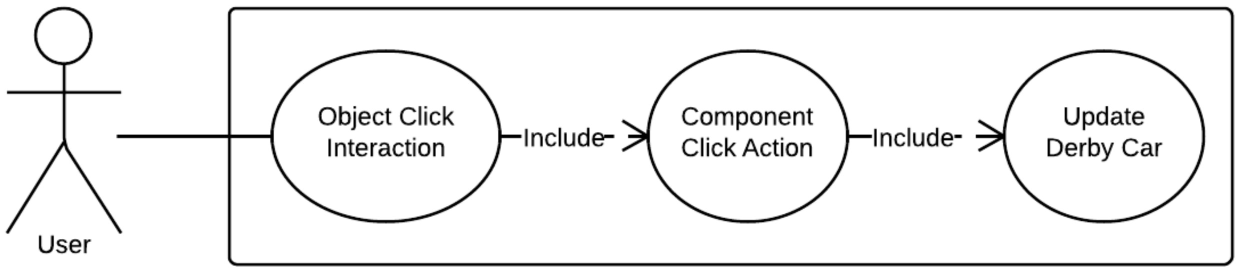
UC41. Pause Exploring of Pinewood Derby Exhibit – When the Pause Menu is brought up, the user is no longer able to control the player. Four options are displayed while the user is within the Pinewood Derby Exhibit. These options are "Resume", "View Tutorial", "Restart", and "Return to Lobby Center". See Use Case Diagram 41 below.

Pause Pinewood Derby Exhibit (UC41)



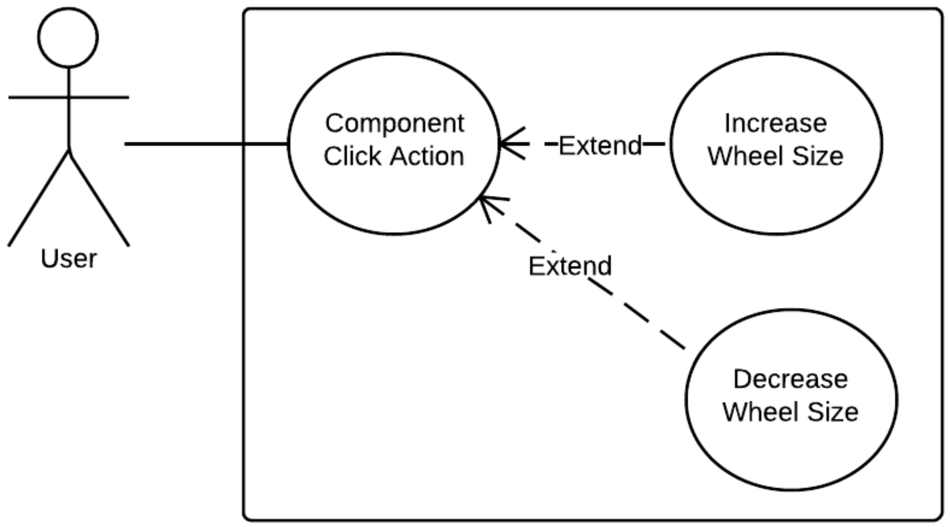
UC42. Pinewood Derby Workbench – Within the Pinewood Derby Exhibit Room, there is a Workbench. The Workbench is used to create and customize a Derby Car. When the Workbench is clicked, it checks to see if a Workbench component was clicked. When a Workbench component is clicked, the Component's click action method is called. After the Component Click Action finishes, the display Derby Car and the Derby Car at the top of the ramp are updated. See Use Case Diagram 42 below.

Pinewood Derby Workbench (UC42)



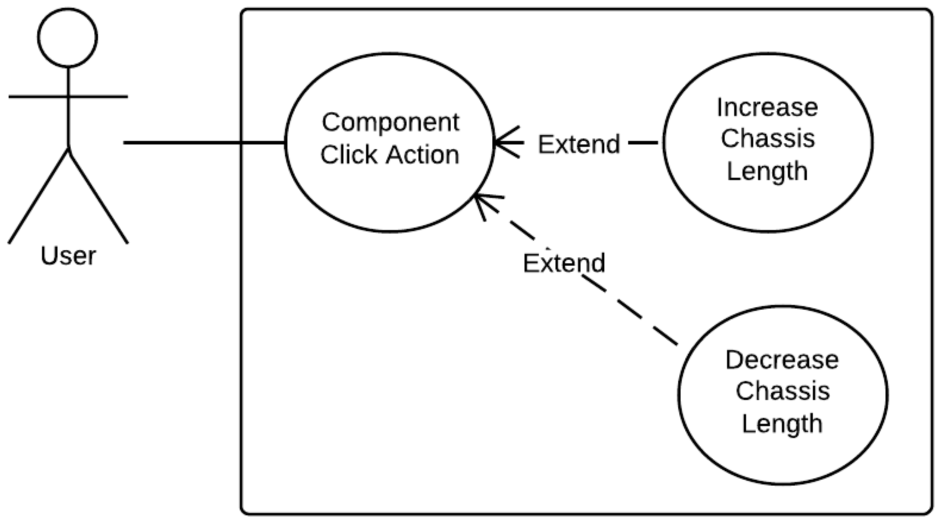
UC43. Workbench Wheel Component – One of the set of components on the Workbench is the Wheel Components. There are two sets of Wheel components. One set is for changing the front wheel, while the other set is for changing the back wheel. All the Wheel Components have two options: increase wheel size and decrease wheel size. See Use Case Diagram 43 below.

Workbench Wheel Component (UC43)



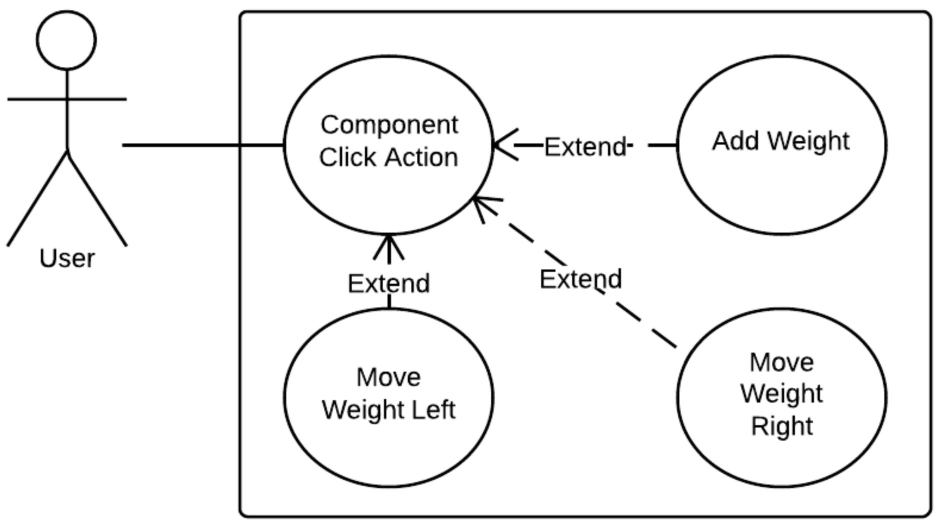
UC44. Workbench Chassis Component – Another of the set of components on the Workbench is the Chassis Components. There is one set of components for changing the Chassis. All the Chassis Components have two options: increase chassis length and decrease chassis length. See Use Case Diagram 44 below.

Workbench Chassis Component (UC44)



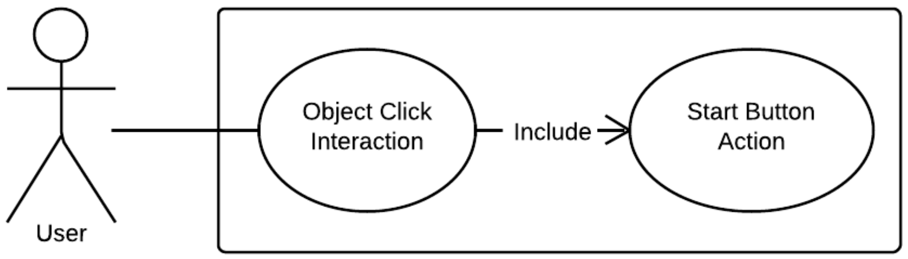
UC45. Workbench Weight Component – The last set of components on the Workbench is the Weight Components. There are three options for the Weight Components: add weight to Derby Car, move weight left, and move weight right. See Use Case Diagram 45 below.

Workbench Weight Component (UC45)



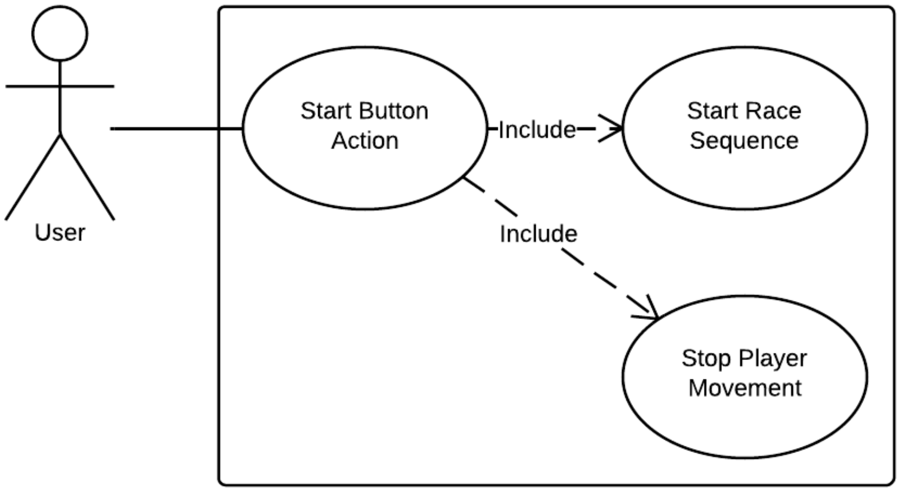
UC46. Pinewood Derby Start Button – Within the Pinewood Derby Exhibit Room, there is a Start Button placed on the Track. The Start Button is used to test the user created derby car by releasing it down the derby track. When the Start Button is clicked, the Start Button Action is called. See Use Case Diagram 46 below.

Pinewood Derby Start Button (UC46)



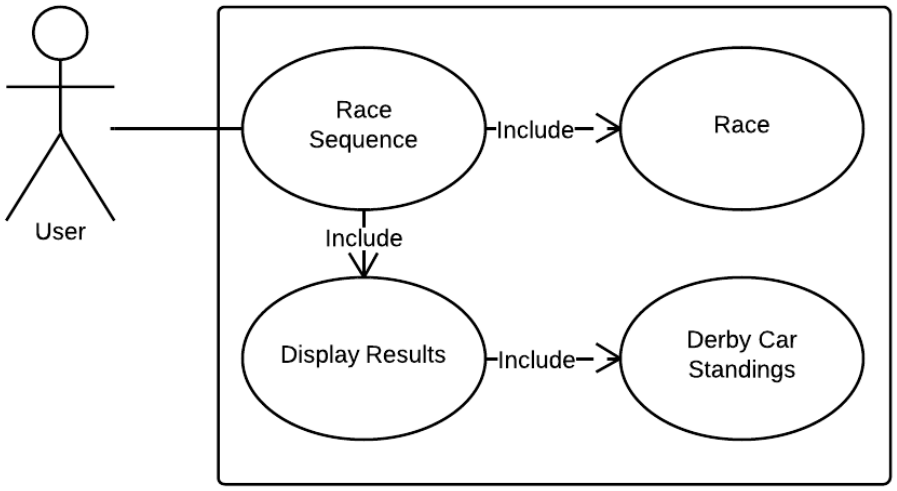
UC47. Pinewood Derby Start Button Action – When the Start Button is clicked, the Start Button Action is called. Within this action, two other actions occur. First, the user loses control over the Player. The user will no longer be able to move the player or look around. Instead the camera will focus on the derby car so that the user can see the race. Second, the Race Sequence is started. See Use Case Diagram 47 below.

Pinewood Derby Start Button Action (UC47)



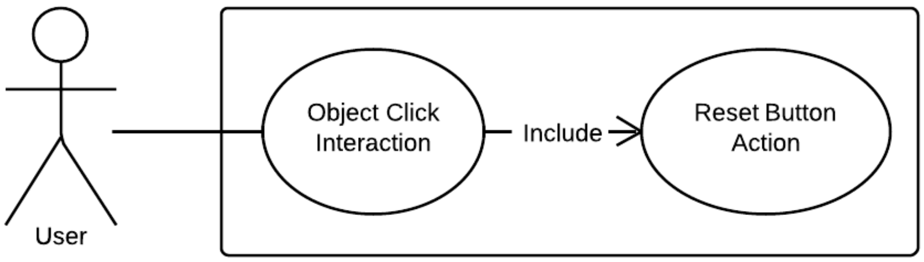
UC48. Pinewood Derby Race Sequence – The Race Sequence is the sequence for releasing the derby car from the top of the track and watching it move along the track. When the derby car passes the finish line, the race is over and each Derby Car's standing is displayed at the finish line. See Use Case Diagram 48 below.

Pinewood Derby Race Sequence (UC48)



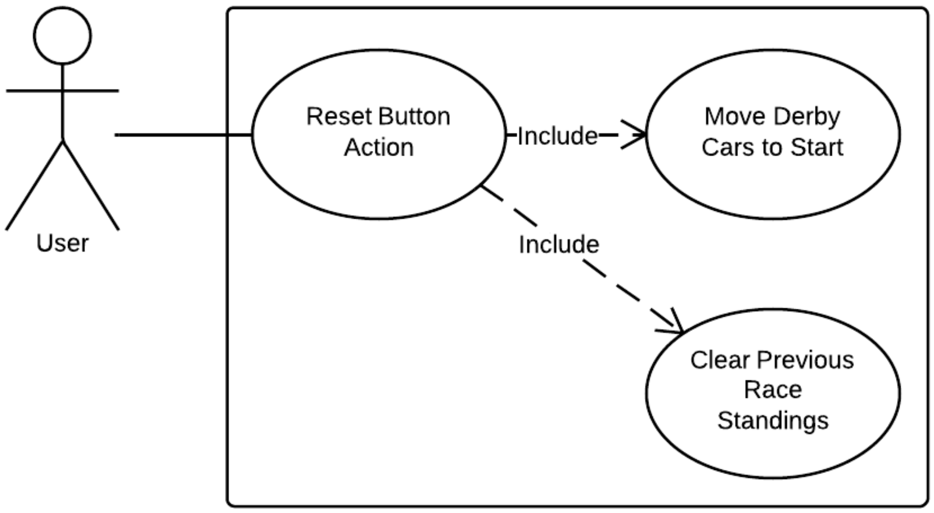
UC49. Pinewood Derby Reset Button – Within the Pinewood Derby Exhibit Room, there is a Reset Button placed on the Track. The Reset Button is used to reset all the Derby Cars back to the starting position after a race has been completed. When the Reset Button is clicked, the Reset Button Action is called. See Use Case Diagram 49 below.

Pinewood Derby Reset Button (UC49)



UC50. Pinewood Derby Reset Button Action – When the Reset Button is clicked, the Reset Button Action is called. Within this action, two other actions occur. First, all the Derby Cars on the track are moved back to the starting positions. Second, the race standings are cleared in preparation for the next race. See Use Case Diagram 50 below.

Pinewood Derby Reset Button Action (UC50)



5.3.3 Prototype objects

Derby Car

The prototype derby car object only includes the rigid body objects used by the physics engine to simulate the derby car. The car is made up of three objects: a back wheel, a front wheel, and a connecting piece.

The two wheels are joined to the connecting piece by Unity's HingeJoint physics component. This component binds an object to a point, and allows it to rotate around a vector leading from that point. The HingeJoints are connected to each end of the connecting piece at a fixed distance away from the center, with the rotation axis aligned with where an axle would be on a derby car.

The wheels use sphere collider components, which provide smooth collision detection against the surface of the track. To keep the derby cars from falling over, the motion of all three objects is locked to the XY plane (aligned with a slot in the track), and the rotation is locked on the Z axis.

Each object contains a RigidBody component which links the object to Unity's physics engine. Each component has a mass component, which is updated when the parts of the car are modified.

Finally, one car is given a FollowCamera script, which allows the camera to follow behind their car when they start the race.

Ramp / Track
The ramp is where the derby cars are lined up and raced. It uses an invisible gate object to keep cars aligned at the top of the ramp before each race.

Ramp Scoreboard
The scoreboard is located at the end of the ramp, and is used to determine the winner of each race. Eight colliders are placed on the finish line, one for each slot in the track, and are used to determine when a car reaches the finish line. Once a car reaches the finish, its place in the race is displayed as a number above the corresponding slot in the track.

Workbench
The workbench is where the user goes to modify their derby car. It contains all the parts the user can select from. By clicking on a part, the part will be added to the user's car. A duplicate of the user's car is displayed on the workbench, so they can see the changes made to their car as they apply them. The workbench displays the current weight of the derby car, as well as the maximum weight, so the user can know how heavy their car is as they put it together. The pieces on the workbench work through the parent workbench object to issue commands to the car on display and the car on the track at the same time.

Wheels
Different sized wheels are provided for the front and back of the car. Each sized wheel uses the same script to update the size of the wheels on the derby car.

Chassis
Chassis of different lengths are also provided for the user to use with their car. These buttons scale the connecting piece of the derby car.

Extra Weights
We planned for extra weights to be added to the derby car, but we did not complete this feature in time. The buttons to configure the position and size of the weight are present on the workbench, however.

5.3.4 Prototype scripts

Derby Car Script
This script serves as the main script for interacting with a derby car. It contains functions for adjusting the parts of the derby car, and for resetting the car to the top of the ramp.

Derby Launch Button
This script disables the starting gate and switches the camera view to follow the user's derby car.

Derby Reset Button
This script moves the derby cars back to the top of the ramp, enables the gate to keep them from going down the ramp again, and resets the score display on the scoreboard at the end of the ramp.

Derby Workbench Script
This script serves as a link between the buttons on the workbench and the derby cars themselves. All buttons on the workbench use functions in this script to modify the derby car. The derby car to edit is chosen through the editor and passed to this script.

Finish Line
When a Derby Car crosses the finish line, this script will update the car's lane's standing on the Scoreboard.

Follow Derby
The camera follows behind the user's Derby Car at a fixed angle, updating every frame.

Pinewood Derby Pause Script

Using the generic Pause Menu, the Pinewood Derby Pause script extends to include additional functionalities (such as the ability to view the view a tutorial and restart the room).

5.3.5 Pinewood Derby Pause Menu

The Pinewood Derby Pause Menu was not completed but here are the reachable states within the prototype:

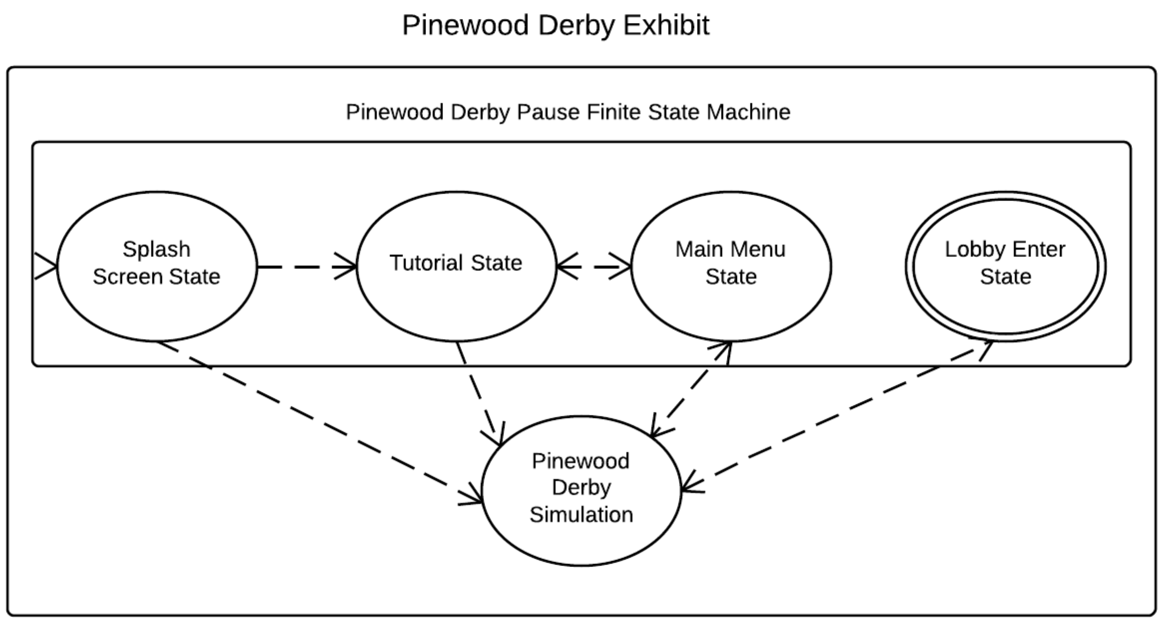
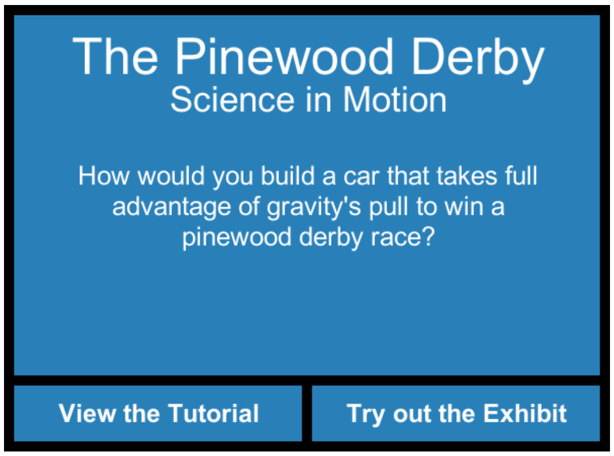


Diagram of the Pinewood Derby Exhibit’s GUI Finite State Machine.

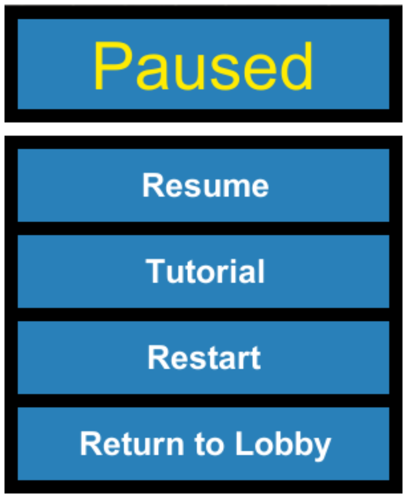
The Pinewood Derby Exhibit finite state machine has three main states that can be transitioned to and an additional state for the Lobby Enter State.

Splash Screen State


The Pinewood Derby
Science in Motion
How would you build a car that takes full advantage of gravity's pull to win a pinewood derby race?
View the Tutorial
Try out the Exhibit

The Splash Screen State is the state that is displayed when the Pinewood Derby Pause Menu is first loaded. This is the only way to reach this state. The Splash Screen State displays a welcome message to the user and gives the user two options: “View the Tutorial” and “Try out the Exhibit”. When the “View the Tutorial” option is selected, the finite state machine transitions to the Tutorial State. When the “Try out the Exhibit” option is selected, the simulation resumes giving the user access to the exhibit.

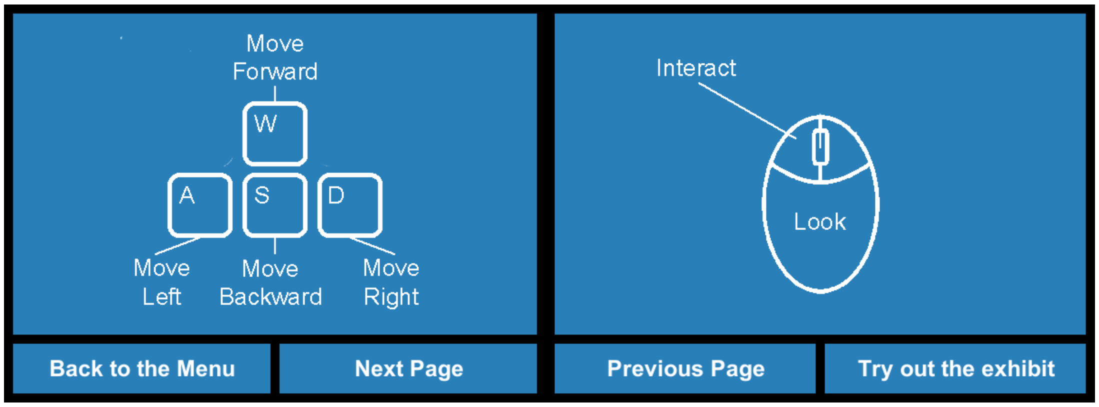
Main Menu State



The Main Menu State is the state the default state for the Pinewood Derby Pause Menu and is loaded whenever the simulation is paused. There are three ways to enter the Main Menu State: transition from the Tutorial State, transition from the Lobby Enter State, or by pressing the escape key during the simulation. The Main Menu State gives a list of options to the user. This list includes: “Resume”, “Tutorial”, “Restart”, and “Return to Lobby”. When the “Resume” option is selected, the simulation resumes. When the “Tutorial” option is selected, the finite state machine transitions to the Tutorial

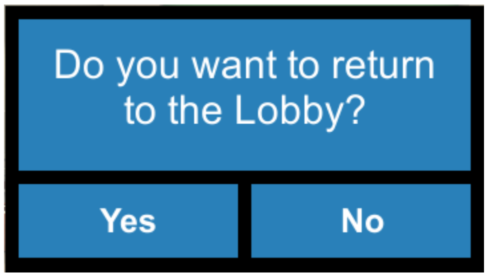
State. When the “Restart” option is selected, the Unity scene is reloaded. When the “Return to Lbby” option is selected, the finite state machine transitions to the Lobby Enter State.

Tutorial State



The Tutorial State is a state that displays a list of images that help describe how to move around and interact in the Pinewood Derby Exhibit. The Pinewood Derby Tutorial has two images to inform the user. The first image lists the keys used in the Pinewood Derby Exhibit, which include the “W”, “A”, “S”, and “D” keys for movement. The second image displays how the mouse interacts in the Pinewood Derby Exhibit. Moving the mouse moves the view angle within the exhibit and left clicking will interact within the exhibit. There are two ways to enter the Tutorial State: transition from the Main Menu State or transition from the Splash Screen State. For each Tutorial page two options are given to the user. These options include “Previous Page” and “Next Page”. If the tutorial page is the first page, the “Previous Page” option is changed with the “Back to the Menu” Option. If the tutorial page is the first page, the “Next Page” option is changed with the “Try out the Exhibit” Option. When the “Back to the Menu” option is selected, the finite state machine transitions to the Main Menu State. When the “Try out the Exhibit” option is selected, the simulation resumes. When the “Next Page” option is selected, the next tutorial image is displayed. When the “Previous Page” option is selected, the previous tutorial image is displayed.

Lobby Enter State



The Lobby Enter State is the state that asks the user if they would like to enter the Lobby Center. There are three ways to enter the Lobby Enter State: clicking on the door placed in the Gravitron Exhibit, by running into the door placed in the Gravitron Exhibit, or by transition from the Main Menu State. The Lobby Enter State gives the user two options: "Yes" to entering the Lobby Center, and "No" to not enter the Lobby Center. When the "Yes" option is selected, the user is then transported to the Lobby Center. When the "No" option is selected, the simulation resumes if the Lobby Enter State was transitioned from the door in the gravitron or the finite state machine will transition to the Main Menu State if transitioned from the Main Menu State.

5.4 Simulating Derby Car Physics

5.4.1 Introduction

There are many physical principles which must be understood to get the most out of a derby car. Where should weight be added? How does one reduce friction from the wheels on the axles? Does the size of the axle matter in relation to the bore of the wheel? What shape chassis is the most aerodynamic? To build a fast derby car, one must know these principles well. To create a reasonably accurate simulation of a running derby car, one must know these principles even better.

Fortunately for us, Unity comes with a physics engine built-in which does all the heavy lifting for us. This does not make it easy, however. It took us many iterations and models before we found one that would work well for simulating the movement of a pinewood derby car. We also had to take into account the ways in which the car could be modified by the available parts. We kept three physical principles in mind when designing the physical car model and the ways in which it could be adjusted: inertia, center of mass, and height.

Even with the part restrictions in place, we believe the simulation will not be a disappointment! Many restrictions are no greater than those found at the real life exhibit, and the virtual environment will still give us greater freedom to add more parts and challenges. We believe that as long as the core lesson of the exhibit is implemented--how gravity affects the speed of a derby car--the simulation will succeed.

In the next few sections, the physics of derby cars and how we plan to simulate them will be explained, mostly in terms of the three main physical principles listed above, and how each kind of part affects the speed of the derby car. An apology to physics majors: you may cringe a bit at some of the following explanations, but you will survive.

5.4.2 Inertia

The weight of a derby car is a very important factor in how fast it finishes the race. Generally, adding more weight makes for quicker times, and so many participants try to get their cars as close to the weight limit as possible. But it is not the simple act of adding weight which makes the car faster--after all, heavier objects don't fall faster than lighter ones (in a vacuum). The following equations will explain:

$F = Ma$	*Force = Mass * Acceleration*
$f = mg$	*force = mass * gravity*
$Ma = mg$	*No difference between gravitational mass (m) and inertial mass (M)*
$a = g$	*Acceleration is the force of gravity for all objects (on Earth)*

The force of gravity is greater on objects which have more mass. However, the increased force on heavier objects is canceled out by the object's inertia. Because there is effectively no difference between gravitational mass and inertial mass, they are treated as the same thing in the above equations. The downward force of gravity is the same on any derby car, heavy or light. Inertia still makes a difference in speed however, especially when we move outside of the frictionless vacuum of which physicists are enamored. Greater inertia allows a moving object to better overcome wind resistance, and this is why heavier cars are generally faster than lighter ones.

The result of this is a maximum weight limit imposed on each derby car. The challenge is to pick the right combination of parts which will lead to the heaviest car that does not exceed the weight limit. However, care must be taken as to where the weight is located on the car.

5.4.3 Center of Mass and Height

For derby cars with the same weight, the placement of the weight can make a difference. At the start of a race, the potential energy for a derby car is given with the following equation:

$E_p = M * g * h$

E_p	*= Potential energy*
M	*= Mass*
g	*= Gravitational Force*
h	*= Height of the center of mass, relative to its resting height on level ground*

When the center of mass of the derby car is placed further back, the center of mass is higher up on the starting ramp, giving the derby car a higher amount of potential energy.

Using the equation for kinetic energy, we can determine the velocity of the derby car at the end of the track:

$E_k = (½) * M * v^2$

E_k	*= Kinetic energy*
M	*= Mass*
v	*= Velocity*

$(½) * M * v^2 = M * g * h$ *Masses cancel out*
$(½) * v^2 = g * h$ *Solve for velocity*
$v^2 = 2 * g * h$
$v = \sqrt{(2*g*h)}$

This equation ignores wind resistance, but that is effectively a constant applied to all derby cars. Thus, we can use the above equation to show that the speed of the derby car at the end of the race is greater when the center of mass is further back on the car (and thus higher up on the ramp, meaning *h* is greater).

Of course, the wheelbase of the car must be taken into consideration. If the weight is placed on or behind the rear axle, the car may “wheelie” on the way down, and that will greatly slow down the car, assuming it can complete the course without flipping over.

Treating each axle as the fulcrum of class one lever (where the fulcrum is in the middle of the effort and resistance forces, like a seesaw), we can determine if the car will flip over from the weights placed on either side of each axle by counting up the weight of the items on either side of each axle, and determining the proportional weight of the chassis on each side.

Each aspect of adjustability on the derby car allows the user to shift the center of gravity one way or the other. Lengthening the chassis places the CoG higher up on the ramp, increasing potential energy. Adjusting the size of the wheels on either end of the car can shift the center of gravity toward the front or back of the car. Finally, placing a weight on the top of the car is the most direct way of affecting the CoG.

5.6: Final Assessment

For the final product we presented to our Senior Design committee, our Pinewood Derby exhibit fully satisfied the following functional requirements: 1, 3, 4, 5, 6, 10, 12, 16. We partially satisfied requirements 11 and 14, and we did not satisfy 2, 7, 8, 9, 13, and 15.

In short, users are currently able to change the front and rear wheels between 3 sizes and then race against 3 other preset cars. The positions that the cars finish in are recorded and displayed to the user. The user can then reset the cars, modify their design, and retry the race.

Requirement 11 is only partially satisfied because the wheels and the length of the chassis are the only parts of the car that can be modified. Requirement 14 is partially satisfied because we were tracking the time of the user’s car but weren’t displaying that to the user or recording it.

For the requirements we did not satisfy, most involved the customization features that we were almost finished with. The rest focused on a tutorial for constructing the car, ending failed races automatically, and listing the fastest times achieved by the user.

We also fully satisfied non-functional requirements 1 and 2. For the third non-functional requirement, we did not satisfy it since our current cars appear just as two wheels connected by an axle. However, we believe that this two would have been a relatively quick feature to implement since we would just need to render the proper objects onto the existing physics model

The bulk of our time working on Pinewood Derby was spent on trying to figure out the right way to build a car within Unity. Although it seemed like it would be quite simple, the actual physics are not so intuitive. These unforeseen delays set us back and led to the incomplete requirements. Fortunately, we had stated in our requirements that we would have at least one functional exhibit which was satisfied by the Gravitron.

6. Administrative Content

6.1 Budget and Financing

All software used for the development of this project is free so no budget is required. The Orlando Science Center will be providing the web hosting for the final system.

Software(s)	Cost
Unity3D Game Engine	$0
MonoDevelop	$0
Blender	$0
Gimp	$0
Web Hosting costs	$0

7. Project Summary

7.1 Summary

The goal of this project was to provide an online application which simulates at least one exhibit from the Orlando Science Center by using Unity and the Unity Web Player. Multiple users can run a standalone version of the simulation through their preferred web browser with the Unity Web Player. By putting this simulation on the web for free, the OSC can be experienced by anyone who desires, even if they don't live near Orlando. The exhibits aim to educate the users about the Engineering Design Cycle which is an implementation of the Scientific Method.

7.2 Constraints and Risks

Certain constraints do come to mind; such as the scope of the project in terms of practicality or accomplishing what we desire with our lack of experience in Unity. Some creative constraints are evident already, such as which exhibits will be suitable for online viewing, and how interactive exhibits will translate to an online environment. Listed below are some constraints/risks with possible solutions:

- Limited time to work in the Presagis building as we can only visit when our sponsors are available.
  - Solution: By creating an API or Interface for the functions we require, we can work outside of the Presagis office more often.
- Presagis' Vega Worlds is in development still, thus dramatic changes and code breaking will be common.
  - Solution: We can inform Presagis of the incompatibility with the recent revision and revert to a previous revision until issues are addressed.
- Non-code resources: Art, models of exhibits, avatar models, etc.
  - Presagis has told us that they will supply us with an artist to help with these assets, but if that falls through then we may need to create some of the art ourselves.
- Failing to fulfill one or more of our requirements
  - In the worst case, we will only produce one exhibit. But, we will take all precautions to avoid this by planning ahead and managing our time.

- Bugs in the program may prevent progress
    - We intend to seek out Presagis for assistance when we encounter issues that we cannot quickly resolve.
- Breaking the Non Disclosure Agreement (NDA) by disclosing too much information in our documentation about proprietary aspects of Presagis software
    - Send documents and presentations to Presagis for review before turning them in.

8. Appendices

8.1 Sources

Jobe, John D. *Pinewood Derby Physics Lectures*. Web. 12-3-2013. <http://www.pinewoodderbyphysics.com/lectures.shtml>

"Unity3D." *Unity*. Unity Technologies, 2005. Web. 04 Dec. 2013. <http://unity3d.com/>

United Arts Campaign, and State of Florida. "Orlando Science Center." *Osc.org - Home*. N.p., n.d. Web. 04 Dec. 2013.
<http://www.osc.org/>