GRAND CANYON UNIVERSITY™

CST-320 Technical Report - Interactions

Period (9/27/21-10/10/21)

Student Name: Diego Guerra, Ryan Scott, Andrew Esch

Faculty Name: Isac Artzi

Project Topic: 3

Current task(s) (refer to the tasks listed in LoudCloud and/or instructor directions; detailed bullet list):

- Students will focus on the interaction between user and objects, as well as the interaction among objects. Students will plan, design, and implement interactions between objects in the scene, using mathematical and scientific concepts. In addition, students will plan, design, and implement user-driven interactions, in which the user affects the conditions and characteristics of objects in a scene, and in response, objects exhibit different behaviors, driven by mathematical and scientific principles.
- From this list, here is current task deliverables for this CLC assignment:
    - Decide on a draft list of interactions that will be useful for our final project
        - Each must utilize some type of mathematical or scientific concept
    - Create assets, scripts, scenes, and etc. to implement these interactions
    - Create screenshots and demonstrate interactions in Unity

Activities performed this week (bullet points with explanations):

- Created a single team Unity Project with Unity Collaboration enabled.
    - Using the 25 GB plan, Andrew created and shared a working project (on Unity 2021.1.20f1) that has full-VR functionality.
    - This project will be the single source for collaborating on the final VR project, as this has all VR settings enabled w/ the OVRPlayerController that can grab objects.
- Created grabbable objects in Unity that function with the Oculus Quest 2.
    - Previously, this was an issue listed (from last CLC's technical report). However, Andrew successfully implemented a working OVRPlayerController that allows this functionality. This was then shared in the Unity Collaboration project, which successfully ran on all of our instances of Unity.
    - Following the Medium tutorial (posted on T2 Padlet), Andrew discovered that there is no "Preferred Height" or "Grab Volumes" option. To fix this, he enabled and customized other options to allow this functionality.
- Added a new scene that procedurally generated different types of GameObjects (e.g. fruit, cactus, etc), which has interaction functionality and a UI menu that has text and buttons.
    - From the Topic 3, Week 1 activity.
- Manipulated the LineRenderer example to create types of interactions, scenes, and features
- Reconstructed most of the Maze Example in Padlet 2, toying with and adding features such:
    - Removable ceiling + super jump to get a birds-eye view and traversal of the maze
    - Altering the RNG of the maze construction

Overall progress (describe new knowledge acquired, successes, ideas generated, etc.):

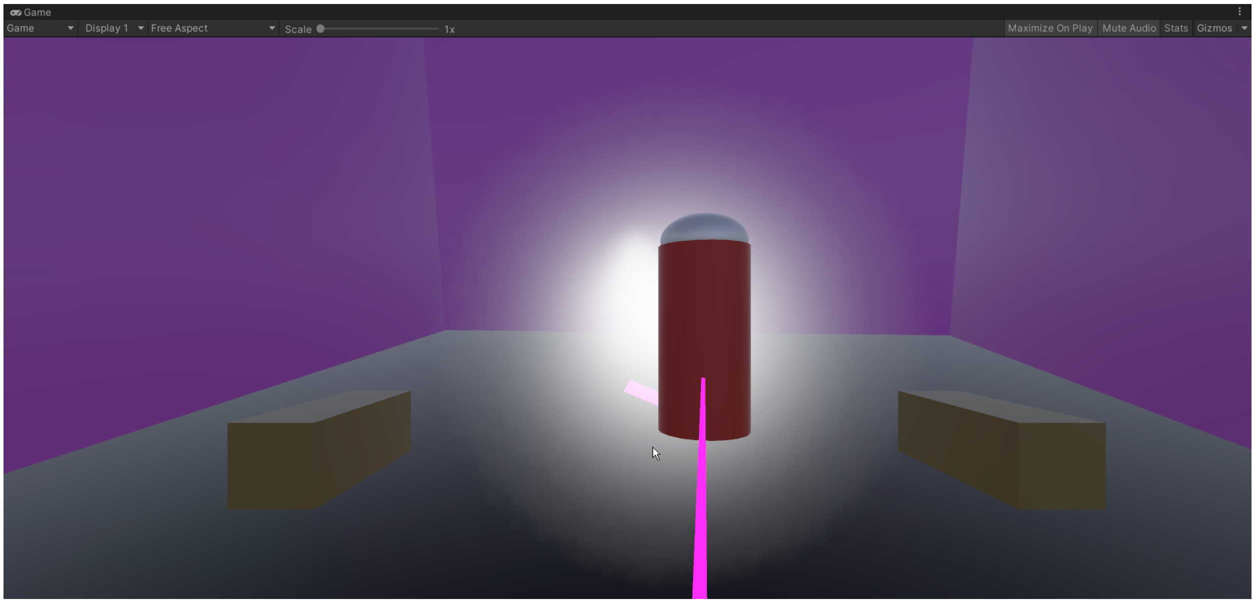
- Skill Inventory Table

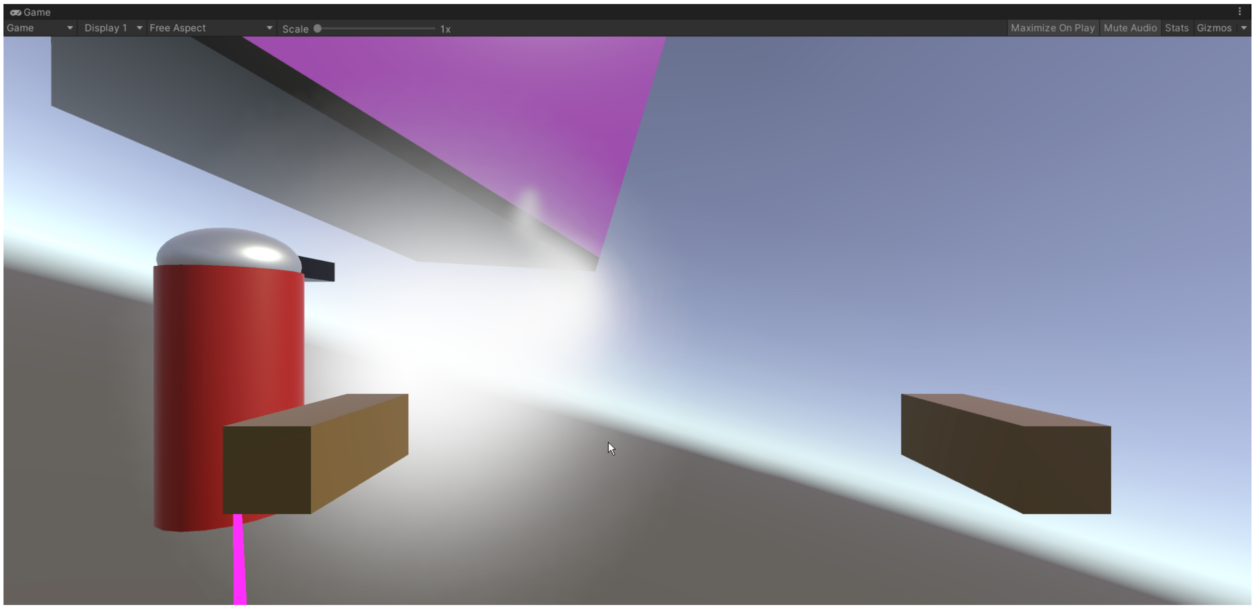
Item #	Who gained the skill/knowledge?	Explain Skill/Knowledge Gained
1	Diego	Successfully build & run applications on the Oculus Quest 2
2	Andrew	Implement a OVRPlayer with functioning hands that can grab objects
3	Andrew, Diego, and Ryan	Create a Unity Collaboration Project
4	Ryan	Procedurally generate mazes and manipulate maze properties
5	Andrew, Diego, Ryan	Raycasting, Line Rendering, and Basic Object Interactions
6	Andrew	Create Complex Particle Systems Using Unity Assets, Settings, and Scripts
7	Andrew	Scripting Orbital Mechanics and Teleport functionality in Unity
8	Andrew, Ryan	Utilize Input Fields, Toggle, Sliders, and other UI elements in Unity C#
9	Andrew, Diego, Ryan	Navigate Between Scenes Using Menus and Box/Sphere Colliders (in C#)

Project Interaction Ideas for Implementation

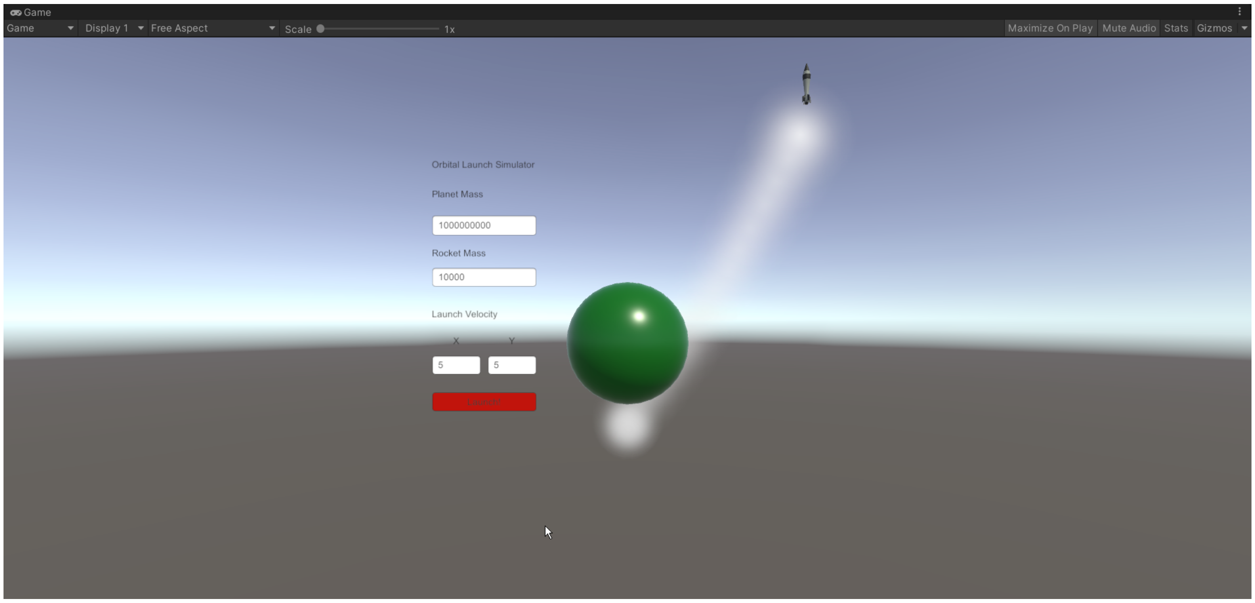
- Space Station Hallway Scene (Diego)
  - Assets Deliverables
    - Door, Floor, and Ceiling 3D Prefab
    - Other Materials for Scene
    - Space Skybox (Ryan)
  - Scripts Deliverables
    - Change Scene with Door Script
    - Return to Main Menu GUI Script
  - Scene Generation Deliverables
    - Space Station Hallway Scene
- Slicing Meshes and Exploding GameObjects using player-wielded lightsaber (Ryan)
  - Assets Deliverables
    - Lightsaber Prefab or other weapon Prefab
    - Various Prefabs (for experimenting with slicing meshes)
  - Scripts Deliverables

        - Script to Handle Trigonometry/Geometry Implementation
    - Scene Generation Deliverables
        - Target Practiced Test Scene
- Fire-Extinguisher Test Room (Andrew)
    - Assets Deliverables
        - Fire Extinguisher 3D Prefab
        - Smoke Particle Effect
    - Scripts Deliverables
        - Fire Extinguisher Script (Physics)
    - Scene Generation Deliverables
        - Fire Extinguisher Test Room Scene





- Frying Pan and Pancake Game Interaction (Ryan & Diego)
    - Assets Deliverables
        - Frying Pan 3D Prefab
        - Pancake Prefab
    - Scripts Deliverables
        - Frying Script
            - (Relate it to Biology, maybe use a biology source to figure out cook times)
        - Stove Script
            - Sphere hitbox
            - Fire Effect
            - Cook when in sphere hitbox
    - Scene Generation Deliverables
        - Space Kitchen Scene
- Rocket Orbit Simulator (Andrew)
    - Assets Deliverables
        - Rocket Prefab
        - Planet Prefab
        - Smoke Particle Effect
    - Scripts Deliverables
        - Rocket Orbit Manipulator Menu
            - GUI shown on in the screenshot
        - Orbital Mechanics Script
            - Utilizes the Gravitational Force formula to implement a rocket object orbiting a planet using user-inputted variables
    - Scene Generation Deliverables
        - Simulation Scene



Orbital Launch Simulator

Planet Mass

1000000000

Rocket Mass

10000

Launch Velocity

X 5

Y 5

Issues that need to be resolved (bullet list):

Note: Our team currently has no Unity-related or VR-related issues to resolve. Instead, the items listed below are ideas we would like to learn about when starting our project.

- Create Oculus Quest 2 long-distance pointer in Unity
- Launching projectiles using raycast
- Advanced particle generation (using custom assets)

Next steps (how will you mitigate the issues listed above; bullet list):

- Research an Oculus Integration tutorial to implement a long-distance pointer
- Research Unity Documentation and related tutorials to implement the feature of launching projectiles using raycast
- Research Blender and Adobe Photoshop tutorials to create custom assets for Particle Systems in Unity

Other comments:

- GitHub Link: https://github.com/drewesch/CST-320
    - Our Unity Scripts are located under /Scripts/Topic-3
- Loom Link: https://www.loom.com/share/0bd8a9fbe37e4b3994b51eeb834eea9b