

CS3247 Game Development

Mini Project 5 Submission

- Name: Yang Shiyuan
- Matrix Number: A0214269A

Learning Outcome

- This project explores the possibilities of the **Portal** game mechanics while incorporating knowledge learned from VarLab Lesson 3: Blueprint Communication and Lesson 5: Physics.
 - This project is further enhanced from mini-project 3 to incorporate **physics** mechanics into the game and to make the game more engaging to play.
-

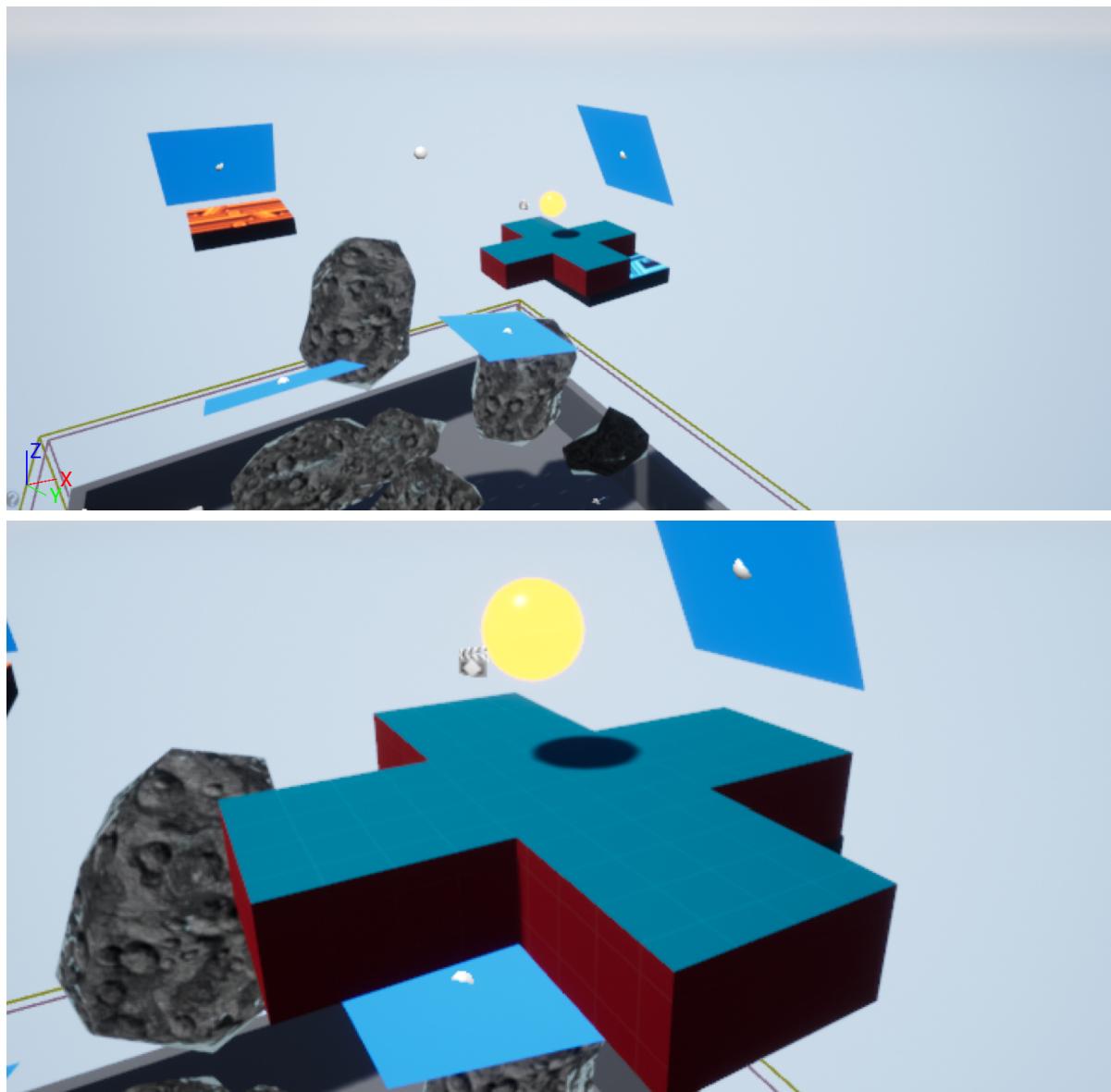
Instructions

1. Download source file from [here](#).
2. Unzip and extract the file.
3. Double-click on `VARlabs_Starter.exe`.
4. Play and evaluate.

Gameplay Guide

Overall

- This project can be summaries to an obstacle course for player to explore and roam while using the famous game mechanics inspired by the game (Portal) [[https://en.wikipedia.org/wiki/Portal_\(video_game\)](https://en.wikipedia.org/wiki/Portal_(video_game))]
- In this mini project, the main goal for the player is to use their creative to reach the **Golden Globe** at the top of the floating platform.



How to play?

- Once pressed **Play**, user will be able to navigate using the common movement keys:
 - **W** = Move forward
 - **S** = Move back
 - **A** = Move left
 - **D** = Move right
- Player is able to use the **Mouse** to orientate their view in the game.

- Player is able to spawn two different colored portals using the following keys:

- **Q** = Spawn **green** portal

- **E** = Spawn **purple** portal



- Do note that player is only able to spawn portal on **walls** that are **blue** in color, similar to the one shown above.
 - When player is trying to spawn a portal, the player will be shooting out a **red** ray trace indicating the possible attempts to spawn a portal. Not all attempts to spawn a portal will be successful, as it will depend on the amount of space left on the wall, available for the player to spawn.
 - At any one time, the player is only able to spawn 2 portals. This sets up a channel of transport for the user to travel. If the player is attempting to spawn more 2 portals, one of the existing portals will be destroyed, allowing the channel of transport to be redirected to the new portal.
- For physics, the player is able to use the slow motion mechanics to allow the player to have more precise control over their movement.
 - **Shift** = Slow motion
- For physics, the game also includes an all mighty push mechanics that allows the player to exert a force push and push him/herself to greater heights.
 - **Left Mouse Click** = All Mighty Push
- There are many ways available for the player to reach the golden globe found on the top floating platforms! Feel free to use your creativity and imagination to roam freely in the existing level to reach your goal.

As there are many mechanics and mathematics used in the creation of the mini-project, this section will serve as a directory for the developers to find the various mechanics implemented. This section will also provide a high-level elaboration on the purpose of the implementation.

Base Asset

Meshes([miniProject5\Portal\Meshes](#))

- [SM_Portal](#):
 - This is a mesh serving as a placeholder for the portal rendered scene.
 - When there is only 1 portal set up, this mesh will be set to a [M_DarkBlack](#) material, indicating that the portal channel is not being set up.
 - When there are 2 portals set up, this mesh in both of the portals will be set to the respective rendered material, showing the scenery projected out of the partnering portal.
- [SM_PortalBorder](#):
 - This is a mesh serving to indicate portal(A|B)
 - It will be set to [MI_PortalRingA](#) in green color or [MI_PortalRingB](#) in purple color.
- [SM_PortalWall](#):
 - This is a mesh serving as a specific surface available for portals to form.

Materials([miniProject5\Portal\Materials](#))

- Color
 - [MI_DarkBlack](#):
 - This is a material instance created to render [SM_Portal](#) to indicate that it is a non-linked up portal
- PortalRenderTargets:
 - [MI_RenderTargetsA](#) & [MI_RenderTargetsB](#):
 - These are material instances that attached to [Portal\(A|B\)](#) to display the scenery that is projected from the partnering portals.
- PortalRing:
 - [MI_PortalRingA](#) & [MI_PortalRingB](#):
 - These are material instances that attached to [Portal\(A|B\)](#) to distinguish the two portals.
- PortalWall:
 - [M_PortalWall](#):
 - This material is attached to the [SM_PortalWall](#) to indicate a wall that is capable to host a portal.

Core Asset

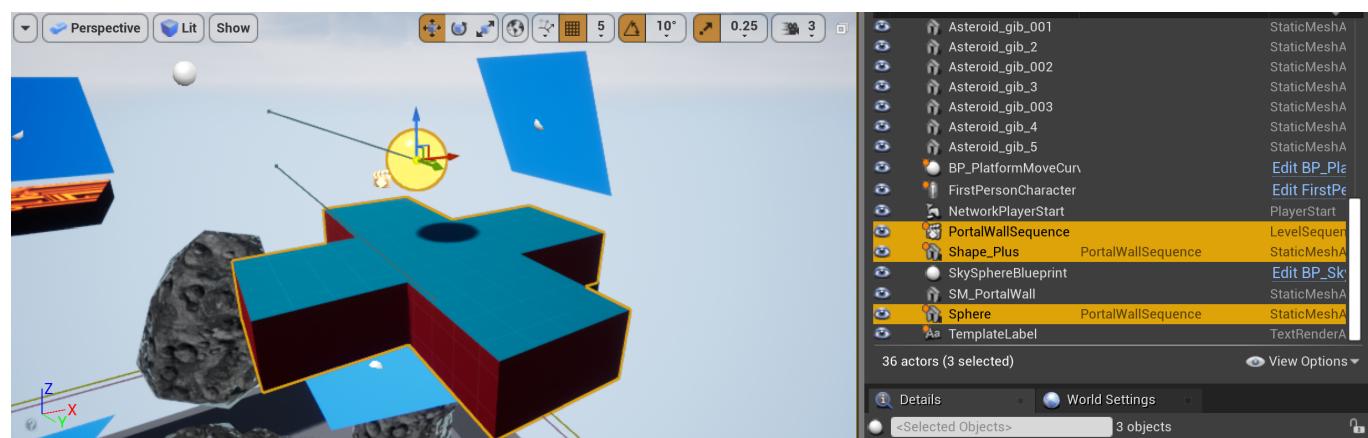
PortalComponent

- Functions:
 - **SpawnPortalAlongVector:**
 - Shoots out a line trace to indicate the attempt to generate a portal
 - Once the line trace hits the **BP_PortalWall**, it will call function **TryAddPortal** to attempt generation of a portal.
 - When there is more than 2 portals, calls function **SwapPortals** to swap out the old **portal(A|B)**.
 - Lastly, using function **sequence** to **Link Portal**
 - **SwapPortals:**
 - Once a portal is being swapped out, this function will destroy that portal.

BP_PlatformMoveCurve & BP_PlatformMoveLinear

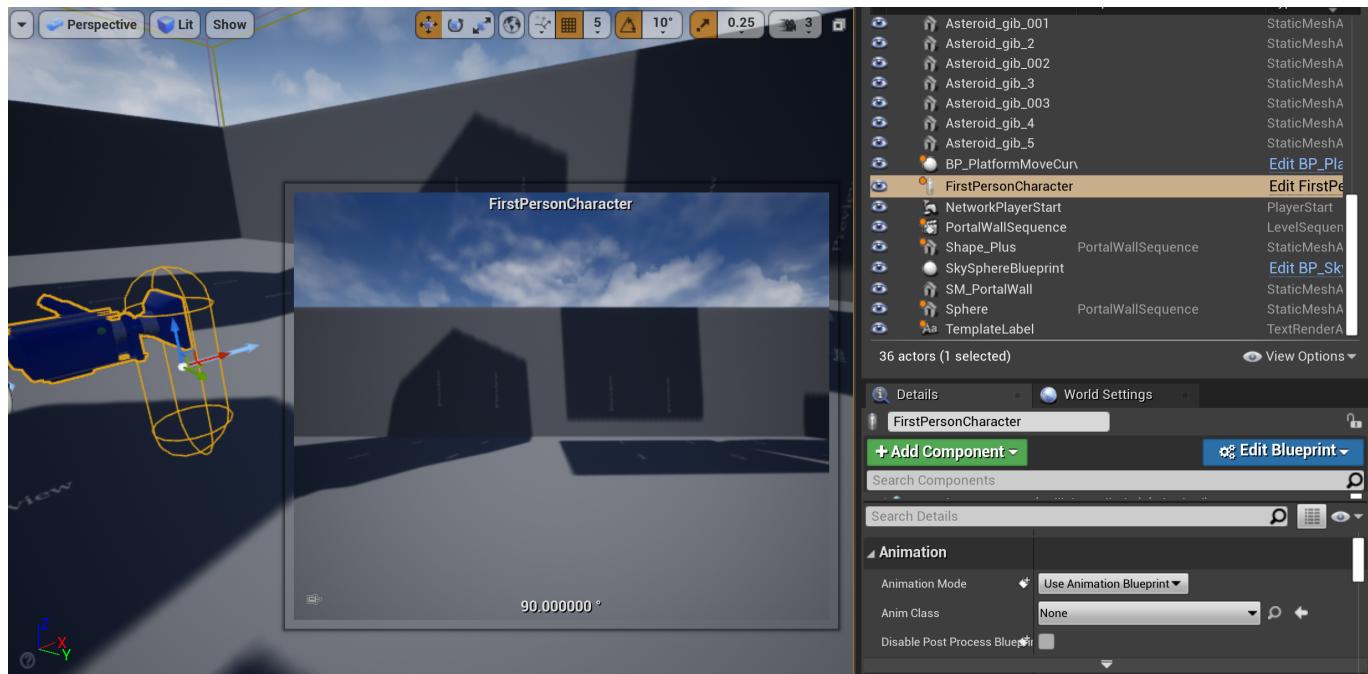
- These assets are modified from the VarLab Lesson 3: Blueprint Communication's **BP_TargetPopup**. They serve as movable platform in the game that helps to bring more challenges to the player when he/she is trying to move towards the **Golden Globe**.
- Event Graph:
 - In here, it contains the **timelines** component which the assessment rubrics required.
 - Since there are **BP_PlatformMoveCurve** & **BP_PlatformMoveLinear**, in total, there are 2 **timelines** components present in this project.

Shape_Plus, Sphere & PortalWallSequence

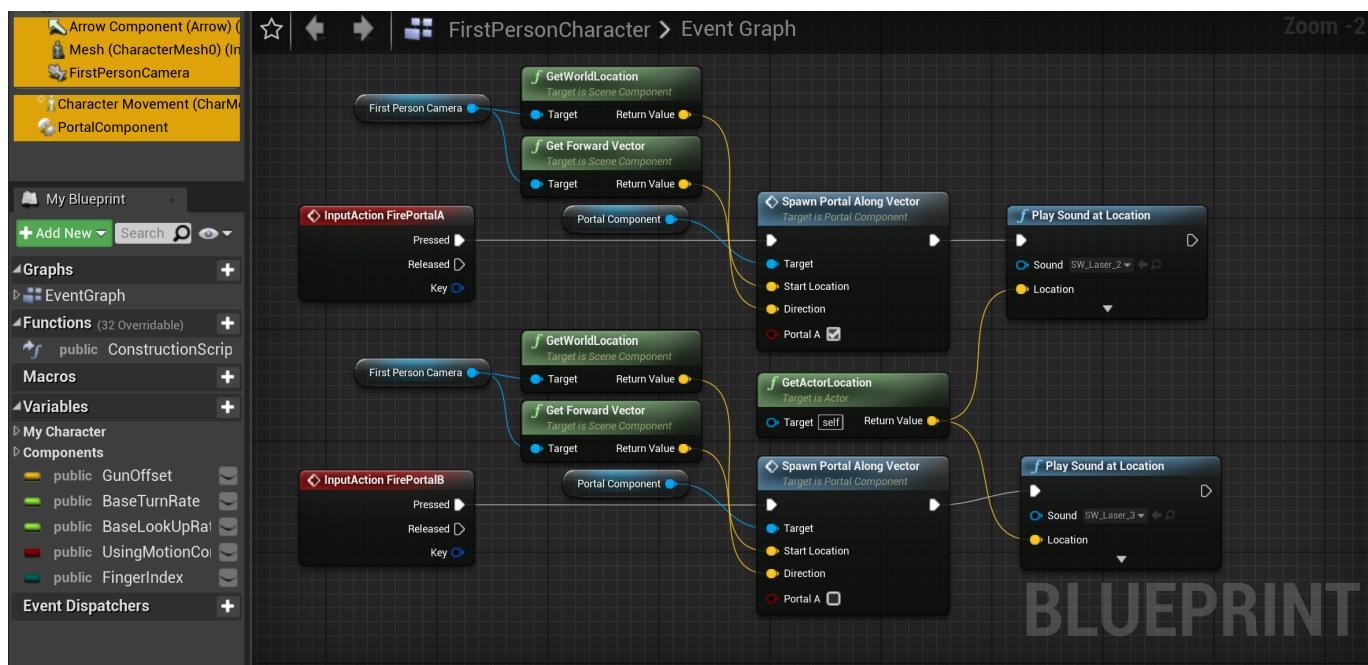


- These 2 game objects utilize the **sequence** as specified by the assessment rubrics.
- In general, these 2 game objects together with **PortalWallSequence**, will cause themselves to move linearly in a straight line manner.
- Their purpose are to offer some challenges for the player to reach the target.

FirstPersonCharacter



- This is a blueprint imported from the Unreal Engine Starter Template.
- The main areas of modifications are shown below:



- Basically, the blueprint implemented takes in the input keys **Q** and **E**. It is tasked to call the function **SpawnPortalAlongVector**, trying to create a portal

BP_PortalWall

- Functions:
 - **ConstructionScripts:**
 - Responsible for setting up the wall in the appropriate size.
 - **TryAddPortal:**
 - Represents a portal creation manager by calling up the relevant functions needed to assess whether a portal can be added

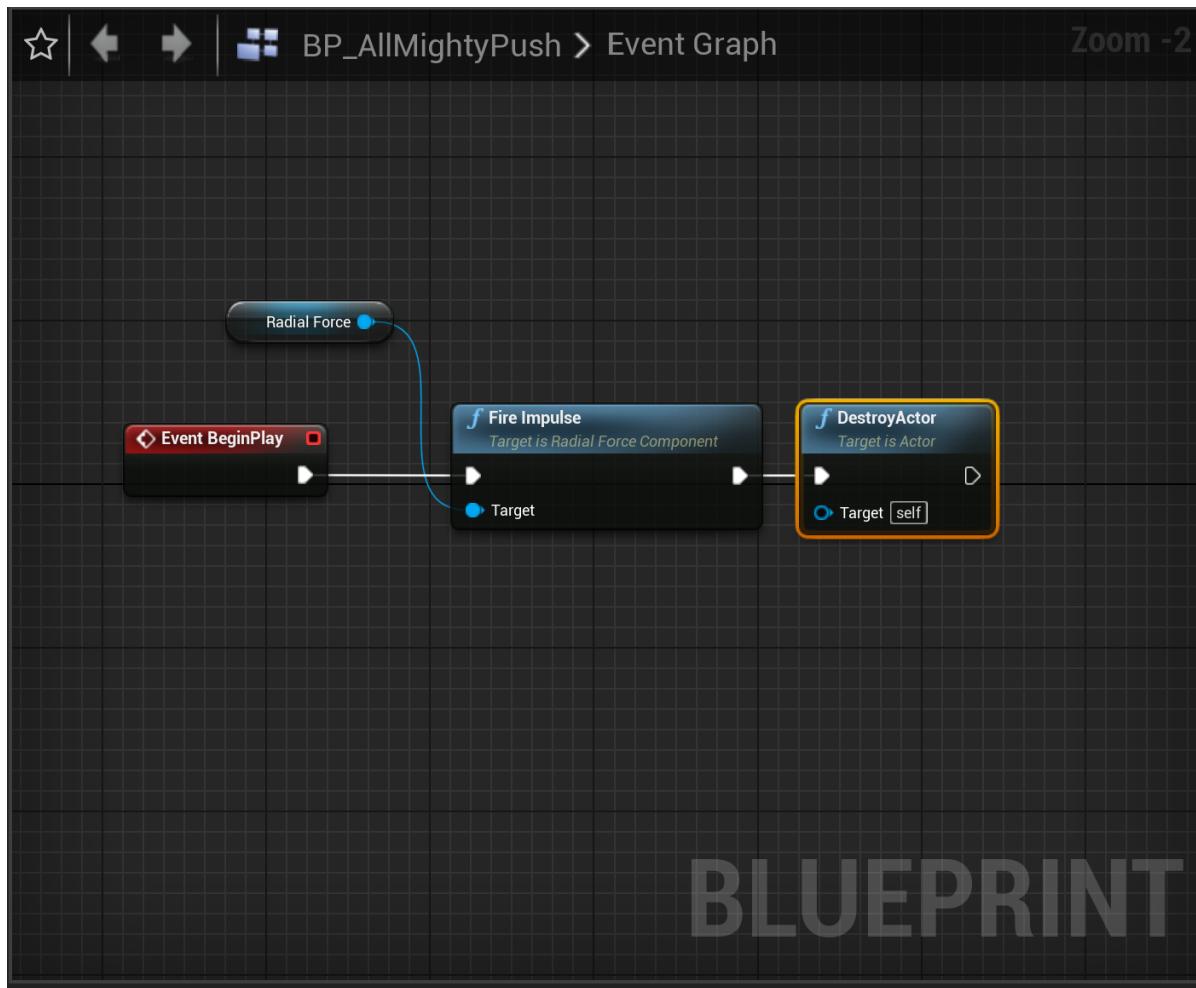
- It also calls the relevant functions to obtain the appropriate geometrical location for the new portal to form.
- **ConstrainPortalToWall:**
 - Helps to keep the portal created within the wall dimensions
- **ClampPointToWall:**
 - Helps to keep the portal created within the wall dimensions
- **HasRoomForNewPortal:**
 - Prevents portals from overlapping each other
- **OnPortalDestroyed:**
 - Remove the oldest **portal(A|B)** when there are 3 portals, such that at any one time, there are only 2 portals.

BP_Portal:

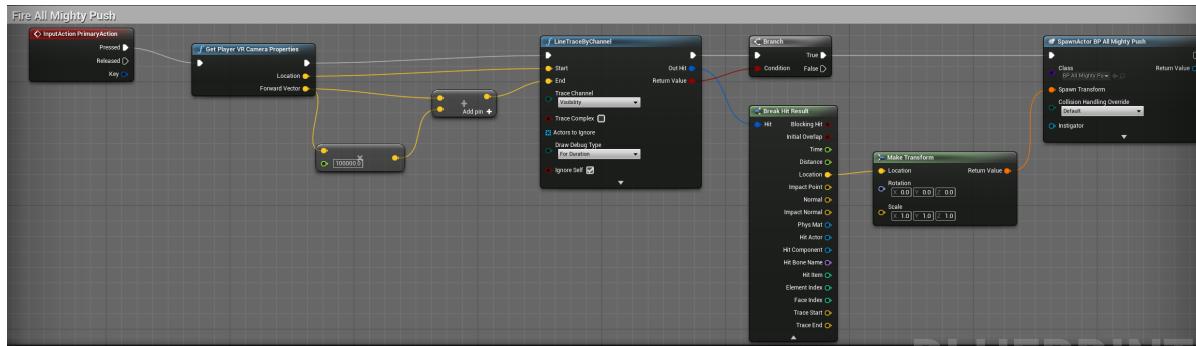
- Functions:
 - **ConstructionScripts:**
 - Sets the relevant materials for the new portal formed
 - **LinkPortal:**
 - Renders the portal to display the scenery projected out from the partnering portal
 - **CheckTeleport:**
 - Checks whether the player is really moving forwards such that he/she wants to cross the portal.
 - **TeleportPlayers:**
 - Teleport the player to the partnering portal

Physics:

- All Mighty Push:
 - Purpose: To provide players more means/methods to reach the goal target.
 - Implemented under **Event Graph:**



- Implemented under **FirsePersonCharacter/Event Graph**:



- Slow Motion:

 - Purpose: To provide players with more precision in their movement.
 - Implemented under **FirsePersonCharacter/Event Graph**:

