Rollover Unlimited

**Game Concept**

In the game, the player must roll a ball around an obstacle course and try to reach the end of the course. The player’s aim is to try and get the highest score possible, with a high amount of collectibles and a quick time contributing to a higher score. Falling off the course will cause you to respawn at the beginning where you are forced to retry the stage again.

**Features**

* Player can move around the environment in all directions
* Player movement will have a drag force added to it.
* Camera will follow player movement but not rotate along with the player.
* Level will have slopes which will cause the player to slow down.
* Level will feature moving platforms.
* Player will have the ability to move with the arrow or WASD keys on PC.
* Player will have the ability to move with joysticks on all other consoles.

**Technical Risks**

This project is on such a small scale that there should be no technical risks, as there is no need from 3rd party libraries and all content is only edited by those in charge of that given content so nothing is overridden.

**Game Flow**

Scenes in the game include:

* Menu - The menu scene will have a button which will either take you to the leader board scene or the game scene (depending on what button the player clicks).
* Game - The game scene will have a portal which takes you to leaderboard and a button to return to menu such as esc key.
* Leaderboard - The leaderboard will have a button to return to the menu where the player can choose to try and get a higher score.

**GameObjects, Scripts and Systems**

Player:

* The player will have one script which controls movement and collision.
* The player movement will be done with to float values moveHorizontal and moveVertical which get the input for horizontal and vertical. The game object’s rigidbody’s add force function is then called and multiples a vector3 called movement and speed.
* The player script will also have a onCollisionEnter function which will dictate what happens when it collides with:
  + Anything tagged portal: Loads the leaderboard scene through a scene manager
  + Anything named DeathZone: respawns the player to the starting location of 0, 0, 0.

Main Camera:

* Camera Movement script which updates the cameras movement with the players.
* It sets an offset and then sets the camera position to the player’s position and adds the offset.

**Input Method**

PC:

* The input method for movement will be the WASD keys or the arrow keys.
* The mouse will be used in menu scenes.

Other Platforms.

* The other platforms will use joy sticks in order to move. Android and iOS will have a controller pad on screen that the player can touch and use.
* Navigating menus will be done by using the joy stick and X button on both PS4 and PS Vita. Android and iOS will feature touch controls to select each option in a menu.

**Deliverables**

At the end of the project the game will be able to run on PC, iOS, Android, PS Vita and PS4.

**System Requirements**

The plan is to get a build of the game running on the following platforms:

* PS4
* PS Vita
* PC
* iOS
* Android

On mobile platforms such as iOS and Android, the games will be landscape with the joy stick appearing in the bottom right corner.

**Third Party Tools**

In unity, the plan is to use Yughues Free Ground Materials and Foxxee’s Spheres packs, which can both be found in the Unity 2017 Assets store.

**Coding Conventions**

The plan is to use Hungarian notation as our primary naming convention. As well as this, the plan is to capitalize the first letter of all function names.

**Source Control**

We will be using SVN as our source control and only asset folders and project settings should be committed with a detailed list of changes included.

Only those assigned to certain content should be changing them and committing them.

**Team Members**

Tasks for Matt:

* UI design
* Menu and Leaderboard layout
* Applying texture art

Tasks for Nick:

* Player and camera movement
* Collision detection
* Scene transition

Tasks for Jeremy:

* Level design
* Applying texture art