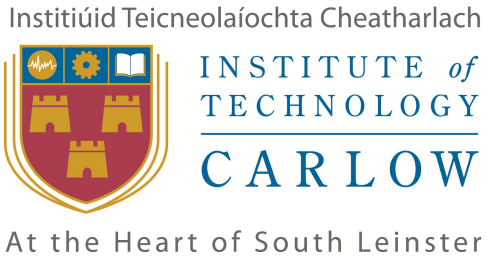

Computer Games Development CW208
Technical Design Document
Year III

Aoife Powders
C00218374

3rd May 2020

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Faculty of Science

Open-Book and Remote Assessment Cover Page

Student Name: Aoife Powders

Student Number: C00218374

Lecturer Name: Phillip Burke

Module: Final Year Project

Stage/Year: 4th Year

Date: 03/05/2020

Declaration

This examination/assessment will be submitted using Turnitin as the online submission tool. By submitting my examination/assessment to Turnitin, I am declaring that this examination/assessment is my own work. I understand that I may be required to orally defend any of my answers, to the lecturer, at a given time after the examination/assessment has been completed, as outlined in the student regulations.

Contents

Game Architecture

Classes

Baseball bat
BatCapsule
BatCapsuleFollower
Velocity Debugger

Button
ButtonPush

Button Game
ButtonCollisionWithPressurePoint
ButtonGameTimer
ButtonRandomColorChange
Button Score
CanvasButtonGameTimer
PlayButtonGame
RandomColorChange

Dial
Dial
DialRotation

Grabbable Stationary Objects
DoorFollow
DoorGrabbable
DrawerFollow
DrawerGrabbable
Follow
FollowRotation

Gun
Ammo
MoveProjectile
Shoot

Important
ClipInPlace
GameManager
GrabbingAndReleasingObjects
Locomotion

UISelect

MazeGame
LeverMoveObjects
MazeGameParticles
MoveMazeObjectCanvas

Rope
Rope
RopeCollisions

Tracking
AxisController
ButtonController
HandleTrackingChanges
SetCorrectCameraHeight
Spawner

Features

Feature: Tracking Controller

Tasks:

- Create a class to handle tracking changes.
- Use Unity XR to get access to input devices list.
- Get the tracked devices for specified roles.
- If the controller was being shown and it was not found disable the controller.
- If the controller was hidden and it's now found, enable the controller.

Feature: Tracking Buttons

Tasks:

- Create a ButtonController class.
- Use Unity XR to get access to the button list.
- Create a dictionary of buttons that will be displayed in the inspector.
- Create a dropdown menu in the inspector to choose which controller is being used, left or right.
- Create a boolean that will keep track of if the button is being pressed or not.
- Get the label selected from the inspector and find it in the dictionary.
- Get the device we want to check (right/left).
- Check if the button is being pressed.
- Trigger OnPress event if pressed.
- Trigger OnRelease event if released.

Feature: Grabbing and Releasing objects

Tasks:

- Create a GrabbingandReleasingObjects class.

- Create game objects to hold the object you're colliding with and the object you're holding.
- Check if you're not holding anything and if the object is grabbed.
- If you're grabbing then the object you are colliding with is moved to the object in your hand and its rigidbody is moved to the fixed joint connected body.
- If the object has been released then the connected body is null, and the object in your hand is null.

Feature: Button

Tasks:

- Create a buttonPush class
- On the button add a spring joint and a collider for how far you want the button to be pressed for something to happen.
- In the class check if the button has collided with the collider.
- Trigger something to happen when the button is pressed.

Feature: UI

Tasks:

- Create a UI canvas.
- Add buttons and text.
- When you point you hand at the canvas draw a ray that extends from your hand.
- To interact with the buttons check if the ray is colliding with the button then check if the trigger button is being pressed.
- Trigger something to happen when you click the button.

Feature: Hitting objects with objects

Tasks:

- Create a bat or something to hit objects with.

- On the bat place multiple cubes with no colliders along where you want the objects to hit.
- Create a batCapsule prefab which will be a cube with colliders.
- Create a batCapsule class that will instantiate the batCapsule prefabs in the same position as the cubes on the bat and set the batCapsules to follow them.
- Create a batCapsuleFollower script to get the velocity of the batCapsules.

Feature: Clip blocks in place

Tasks:

- Create 2 blocks with the same dimensions and names.
- Create a ClipInPlace class.
- Check if both blocks are colliding.
- Check if they have the same name.
- Set them to the same position and rotation.

Feature: Teleportation

Tasks:

- Create WayPoints.
- Create a locomotion class.
- Draw ray when it collides with a waypoint.
- Check if the trackpad button is being pressed.
- If it's being pressed, set the position of the user to the waypoint position.

Feature: Grab from distance

Tasks:

- Create an object to grab.
- Create a grab from distance class.
- Draw ray when it collides with the object you want to grab.

- Check if the grab button is being pressed.
- If it's being pressed, set the position of the object to move towards the users hand.

Feature: Door (also the same for levers and drawers)

Tasks:

- Create a door object.
- Connect the door to a wall using a hinge joint.
- Create a knob object.
- Connect the knob to the door by using a fixed joint.
- Create an object to be placed on the knob.
- Create a follow class.
- Check if the object has been grabbed
- If it has then move the knob towards the object.
- When released return the object to the knob.