EIE3360 Integrated Project
Laboratory 2A – Interactions

1. Objective

In this laboratory exercise, you will use Unity to build a third-person stealth game for Android. You need to develop a game environment, joystick and touchpad for the game. The following topics you will learn in this laboratory.

1. **Camera Movement**
2. **Object Controls (e.g. Key, Single doors, Double doors and Lift)**
3. **HashIDs**

2. Preparation

- A PC with Microsoft Windows
- Unity Hub
- Unity 2022.3.16
- Android SDK
- Microsoft Visual Studio
- Basic knowledge in C# programming, object-oriented programming and 3D graphics

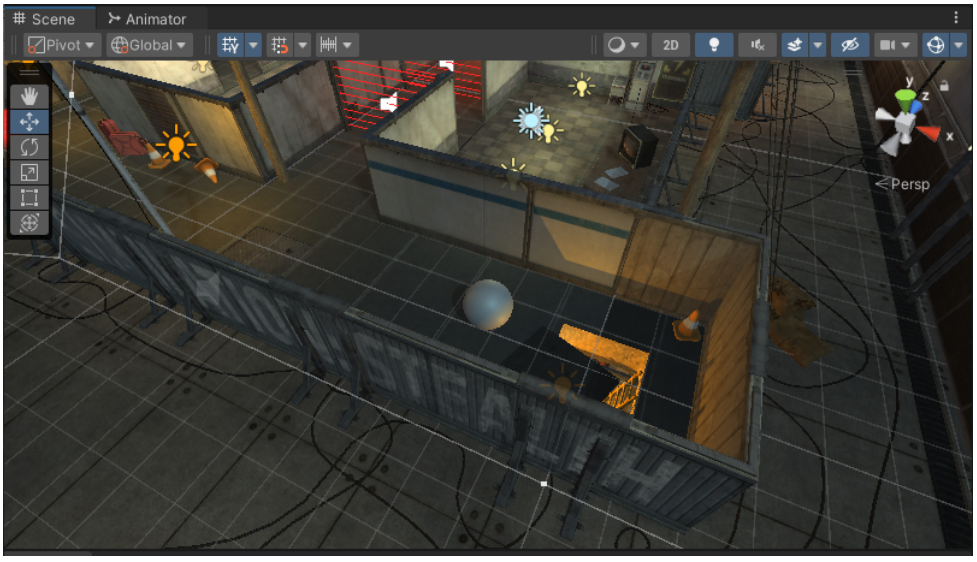
3. Develop a Unity Android Game

This laboratory exercise uses Unity to create a game called "Stealth" for Android. In the game, the player sneaks around to avoid triggering the alarms and tries to escape from a maze where enemies are patrolling.

Continue with the Lab 1A project, save the Stealth_Lab1A scene to Stealth_Lab2A.

3.1 Camera Movement

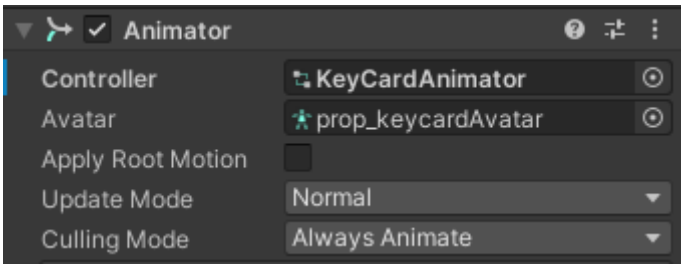
1. Dealing with the camera movement, to ensure the game's controls work correctly, the camera needs to be pointing to the positive world Z axis. This means that when the user presses the up key, the player moves positively in the Z axis. This will translate to an approximately up-direction movement on the screen. Pointing the camera slightly negatively down the X and Y axes in this task.
   a. Adjust the scene view camera to the view shown below.



b. Select “Camera_Main” in the Hierarchy, click GameObject on the top menu → Align With View. (The camera of the game view is the same as the camera of the scene view)
c. Remove the script you created in Lab 1 to set up “Camera_Main” following the game object.
d. Go to Scripts/LevelDynamics in the Project windows and drag the “CameraMovement” to the “Camera_Main” object.
e. Save the scene and the project.
f. Click the Play button to test the camera movement.

3.2 The Key

1. The sliding doors will open when the player or enemies approach, the same system for the lift doors at the end of the level. However, for the lift, the doors only open once when the player has picked up a key. Therefore, storing whether the player has picked up the key.
   a. Go to Scripts/Player in the Project windows and drag the “PlayerInventory” to the “Player” object.
   b. Save the script.
2. **Make the key with a spinning animation for the player to pick it up.**
   a. In the “Models” folder, drag and drop “prop_keycard” into the Hierarchy.
   b. Set the position to X = -22, Y= 0.4, Z = 32.
   c. Double-click the name to focus on it and drag and drop “KeyCardAnimator” from the Animators folder in the project windows to the “Controller” properties of the Animator in the Inspector.



   d. Uncheck “Apply Root Motion”.
   e. Click Add Component → Sphere Collider.
   f. Set the “Radius” to 1 and check “Is Trigger”.
   g. Click Add Component → Light and select “Point” in the Type.
   h. Set the “Range” to 2.3 and the “Color” to R = 0, G = 159, and B = 255.
   i. Set the “Intensity” to 2.5.
   j. Go to Scripts/LevelDynamics in the Project windows and apply the “KeyPickUp” script to the “prop_keycard” object in the hierarchy.
   k. In the “Audio” folder, drag and drop “keycard_pickup” to the “Key Grab” variable of the “KeyPickUp (Script)” in the Inspector.
   l. Rename the original “prop_keycard” from Prefabs to “prop_keycard_bk” and drag and drop the “prop_keycard” from the hierarchy to the “Prefabs” folder to save it as a prefab. Rename the new Prefab to “prop_keycard_lab2a”.
   m. Press the Play button and try to pick up the key.
   n. Save the scene and the project.

If Player goes near the switch, an error will occur.

Fix the error: Discuss with Lab1B groupmate.

ArgumentException: Input Button Switch is not setup. To change the input settings use: Edit -> Settings -> Input

Checkpoint 1: Demo to the instructor or teaching assistant

3.3 Single Doors

A single sliding door will be used for normal rooms on the level. It will open when the player or enemies approach.

1. **To create a single door:**
   a. In the "Models" folder, drag and drop "door_generic_slide" to the hierarchy and set the position to X = -6, Y = 0, and Z = 7.
   b. Double-click the name in the Hierarchy to focus on it.
   c. Rotate it Y = 90.
   d. Click Add Component → Sphere Collider.
   e. Check "Is Trigger".
   f. Set the Center to X =0, Y = 1, Z = 0 and the Radius to 3.
   g. Select "door_generic_slide_panel" under "door_generic_slide", click Add Component → Box Collider.

2. **Apply the sliding animation:**
   a. Go to the "Animators" folder in the Project window and select the "door_generic_slide" in the Hierarchy
   b. Drag and drop "SingleDoorAnimator" from the "Animators" folder to the "Controller" property of the Animator.
   c. Uncheck "Apply Root Motion".
   d. Select "door_generic_slide_panel" in the hierarchy, and click Add Component → Rigidbody.
   e. Uncheck "Use Gravity" and check "Is Kinematic".
   f. Select "door_generic_slide" in the Hierarchy, and click Add Component → Audio Source.
   g. Move "DoorAnimation" script to "Assets\Scripts\LevelDynamics"
   h. Select "door_generic_slide" in the Hierarchy and apply the "DoorAnimation" script to the door "Assets\Scripts\LevelDynamics" folder in the Project windows
   h. In the "DoorAnimation" script, click and select the "door_accessDenied" clip to the "Access Denied Clip" and "the door_open" clip to the "Door Swish Clip".
   i. Rename the original "door_generic_slide" from Prefabs to "door_generic_slide_bk" and drag and drop the "door_generic_slide" from the hierarchy to the "Prefabs" folder to save it as a prefab.
   j. Duplicate 2 "door_generic_slide" game object.
   k. Set both to the positions of X = -15.9, Y = 0, Z = 7 and X = -7.9, Y = 0, Z = 37 respectively.
   l. Save the scene and the project.
   m. Play the game to see whether the doors are workable.

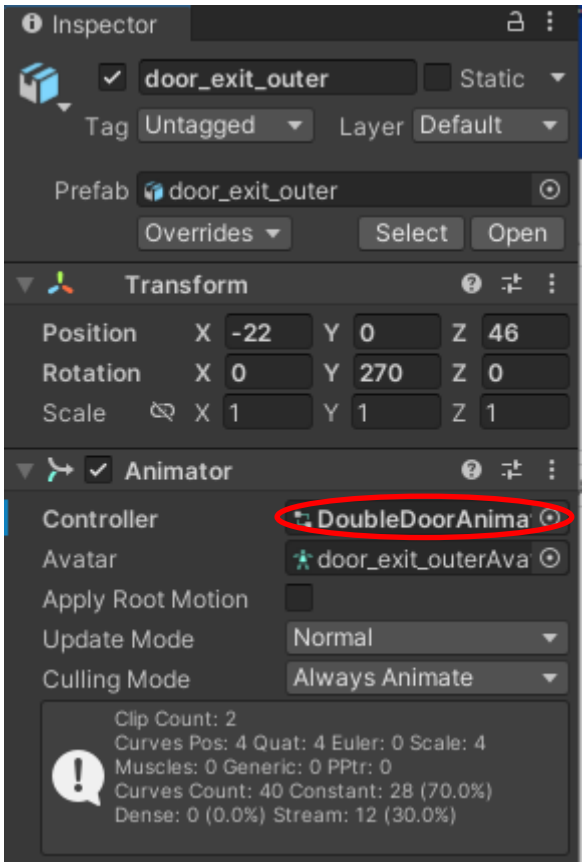
Fix the error: Hints: add hash

NullReferenceException: Object reference not set to an instance of an object DoorAnimation.Update () (at Assets/Scripts/LevelDynamics/DoorAnimation.cs:69)

3.5 Double Doors

Working on the lift by creating a set of double doors, the lift has inner and outer gates, and the inner door tracks the motion of the outer door.

1. **To make the double doors**
   a. In the "Models" folder, drag and drop "door_exit_outer" to the Hierarchy and set the position to X = -22, Y = 0, Z = 46.
   b. Double-click the "door_exit_outer" to focus on it.
   c. Rotate it to Y = 270.
   d. Expand "door_exit_outer" in the Hierarchy, select both child objects and click Add Component → Box Collider.
   e. Select the "door_exit_outer", click Add Component → Sphere Collider and check "Is Trigger".
   f. Set the center of the Sphere Collider to X = -1, Y = 1, Z = 0 and the Radius to 2. (It makes the doors shut behind the player character when he is in the lift).
2. **To make the door animation:**
   a. Select "door_exit_outer" in the Hierarchy, then, go to the "Animators" folder, drag and drop "DoubleDoorAnimator" to the "Controller" in the Inspector.



   b. Uncheck "Apply Root Motion".
   c. Select both child objects of "door_exit_outer" and click Add Component → Rigidbody.
   d. Uncheck "Use Gravity" and check "Is Kinematic".
   e. Apply the script "DoorAnimation" from "Assets\Scripts\LevelDynamics" to "door_exit_outer".
   f. In the "DoorAnimation" script, click ⊙ and select the "door_accessDenied" clip to the "Access Denied Clip" and the "door_open" clip to the "Door Swish Clip".
   g. Check the "Require Key" variable.
   h. Click Add Component → Audio Source; uncheck "Play On Awake".
   i. Rename the original "door_exit_outer" from Prefabs to "door_exit_outer_bk" and drag and drop the "door_exit_outer" from the hierarchy to the "Prefabs" folder to save it as a prefab.

j. Save the scene and the project.

3.6 The Lift

Create a lift. When the player enters the lift, it will raise and end the game.

1. **To make the lift**
   a. In the "Models" folder, drag and drop "prop_lift_exit" to the hierarchy, and set the position to X = -21.85, Y = 1.5, Z = 48 and Rotation of Y = 270.
   b. Create an empty GameObject called "prop_lift_collider", drag and drop it to the child object of "prop_lift_exit" in the Hierarchy and reset the position.
   c. Click Add Component → Mesh Collider, search for "prop_lift_exit_collision_001" in the Assets folder and drag and drop it to the "Mesh" properties in the Inspector.
   d. Select "prop_lift_exit", click Add Component → Box Collider and check "Is Trigger".
   e. Set the "Center" to X = 0, Y = -1.1, Z = 0 and the "Size" to X = 3.3, Y = 0.5, Z = 3.5.
   f. Click Add Component → Rigidbody, uncheck "Use Gravity" and check "Is Kinematic".
   g. Click Add Component → Audio Source, uncheck "Play On Awake" and check "Loop".
   h. Click  in the Audio Clip and select "lift_raise" clip.
   i. Search the script "LiftDoorsTracking" and apply it to the "prop_lift_exit" in the Hierarchy.
   j. Select both "door_exit_inner_left_001" and "door_exit_inner_right_001" under the "prop_lift_exit\door_exit_inner" in the Hierarchy, click Add Component → Box Collider.
   k. Go to "Scripts\LevelDynamics" and apply the "LiftTrigger" script to the "prop_lift_exit" object in the Hierarchy, to trigger the lift's movement.
   l. Rename the original "prop_lift_exit" from Prefabs to "prop_lift_exit_bk" and drag and drop the "prop_lift_exit" from the hierarchy to the "Prefabs" folder to save it as a prefab.
   m. Save the scene and the project.
   n. Create an empty GameObject called "SceneFadeInOut", set Tag to "Fader", drag and drop Scripts\SceneFadeInOut\ScreenFadeInOut to the SceneFadeInOut GameObject.
   o. Set the current Scene to Build Settings -> Scenes In Build and index = 0.
   p. Play the game to check the lift works or not.

4. Exercise

Modify the game to fulfil the following requirements:

1. Stop the alarm system when the alarm system loses the sighting of the player for 5 seconds.
2. Add a touch joystick to the left bottom of the screen and a touchpad button to the right bottom of the screen,
   a. Use a joystick to control the player's movement
   b. Use a touchpad button to **control the laser switch unit, i.e. turn off the laser**.

Refer to Lab 1B 3a for Error:

```
ArgumentException: Input Button Switch is not setup.
 To change the input settings use: Edit -> Settings -> Input
UnityEngine.Input.GetButton (System.String buttonName) (at <a05f37d7a0a34fad9547811820ecb744>:0)
LaserSwitchDeactivation.OnTriggerStay (UnityEngine.Collider other) (at
Assets/Scripts/AlarmSystems/LaserSwitchDeactivation.cs:26)
```

Hints: You may download the joystick package from the assets store https://assetstore.unity.com/packages/tools/input-management/joystick-pack-107631 and the tutorial provided the setup procedures, https://www.youtube.com/watch?v=8-X3BmvtXT0

Checkpoint 2: Demo to the instructor or teaching assistant/take a video with explanations and upload it to Blackboard

Describe and explain the methods you used to achieve the results in your lab report. Any necessary scripts you have added or modified should include.

5. References

[1] Unity - Game Engine, http://unity3d.com/

[2] Unity - Stealth tutorial,

https://www.youtube.com/playlist?list=PLX2vGYjWbI0QGyfO8PKY1pC8xcRb0X-nP

[3] Unity Manual, http://docs.unity3d.com/Manual/index.html

[4] Unity - Script API, http://docs.unity3d.com/ScriptReference/index.html

[5] Unity - The Particle System, https://learn.unity.com/project/creative-core-vfx

[6] Unity – Tutorials, https://learn.unity.com/tutorials