**Worksheet – Interactive Lab Project #1**

*Provide concise sentence form answers to the questions below.*

*Include cropped screenshots, diagrams or illustrations to help clearly communicate your answers.*

| **Score** | | **Out of** | | Question |  |
| --- | --- | --- | --- | --- | --- |
|  |  | | 1. a) | | Explain the difference between public and private fields. |
|  | **1** | |  | | A public field can be accessed from other classes. Private fields can only be accessed from within their own class |
|  |  | | b) | | Give an example of where you used a private field in your project. Explain why you chose to make it a private field? |
|  | **1** | |  | | I made jumpForce a private field because nothing else should be able to modify the player’s jump force. |
|  |  | | c) | | Give an example of where you used a public field in your project. Explain why you chose to make it a public field? |
|  | **1** | |  | | I made the playerController’s canJump field public so that a separate component could access it to check if the player was on the ground. |
|  |  | | 1. a) | | Identify a ***Trigger*** event in your Unity Project. |
|  | **1** | |  | | Moving into the buttons causes an OnTriggerEnter2D event. |
|  |  | | b) | | For each of the 2 overlapping game objects, list the requirements to produce the trigger message. Refer to the ***Collision Action Matrix*** in your answer. |
|  | **2** | |  | | For the Player and Button, a Static Trigger Collider and Rigidbody Collider must overlap to produce the trigger message. For Player’s OnGroundDecector collision box and the Ground, the ground’s Static Trigger Collider must overlap with the Player’s Rigidbody trigger collider. |
|  |  | | c) | | There are multiple component configurations for producing a trigger message. Provide a rationale for why you chose the specific configuration in 2b. |
|  | **2** | |  | | I chose that configuration as the player needs a Rigidbody to move around while the ground and buttons are static objects and only need Colliders. |
|  |  | | 1. a) | | Identify a ***Collision*** event in your Project. |
|  | **1** | |  | | When the Player collides with a Box. |
|  |  | | b) | | For each of the 2 colliding game objects, list the requirements to produce the collision message. Refer to the Collision Action Matrix in your answer. |
|  | **2** | |  | | For the player and box, they are both rigidbodies. So for both, they need to collide with another Rigidbody collider or a static collider. |
|  |  | | c) | | There are multiple component configurations for producing a collision message. Provide a rationale for why you chose the specific configuration in 3b. |
|  | **2** | |  | | I chose the configuration as both player and the boxes need rigidbodies to have physics simulations. |
|  |  | |  | | Why and when is it important to use Time.deltaTime() in algorithms within Update() and FixedUpdate()?  Provide an example of where Time.deltaTime() was used in your project and explain why. |
|  | **2** | |  | | So that the framerate changing doesn’t affect the speed of the simulation. |
|  |  | |  | | Provide an example of where you created a reference to another game object in the Scene.  How was the reference created?  Why was this reference necessary? |
|  | **2** | |  | | In the ButtonPress script I created a reference to the objects I instantiated in order to change their size. |
|  |  | | 1. a) | | Provide an example of where a Prefab was instantiated in the project. |
|  | **1** | |  | | I instantiated a prefab in the ButtonPress script. |
|  |  | | b) | | List the steps you took to create the Prefab. |
|  | **1** | |  | | 1. I made a new game object in the scene. 2. I added all the components to it. 3. I dragged it into the project window |
|  |  | | c) | | List the steps required to create an instance of the Prefab at runtime. |
|  | **2** | |  | | 1. Make a GameObject variable to store a reference to the prefab you want to instantiate. 2. Call Instantiate. |
|  |  | |  | | Provide an example of adjoining game objects that were placed using Vertex snapping. |
|  | **1** | |  | | I placed the walls and ground together using vertex snapping |
|  | | | | | |
|  | **22** | |  | |  |