**Worksheet – Interactive Lab Project #2**

*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 |  |
| --- | --- | --- | --- | --- |
|  | |  |  | Identify an example of a vector math calculation (from the assignment requirements) found within the Unity project. Explain how the results of this calculation were used to accomplish a specific task or result. |
|  | | **2** |  | I used vector math to calculate the distance and angle between the mouse and the player character’s position. I used the angle to control the player’s movement by rotation them towards the mouse, and then applying a force on them relative to the magnitude of the vector to the mouse. I used the same logic for steering enemies towards the player. |
|  | |  | 1. a) | Identify an example of inter-object communication within the Unity project. Explain why this inter-object communication was desired or necessary. |
|  | | **2** |  | I had the PlayerHit script send a message to the GameManager script in order to update the information on the number of hearts left. I decided to do this because I wanted to try keeping individual scripts small. I decided the to make a game manager object to control the hearts and score, and needed a way for it to know when the player was hit. |
|  | |  | b) | List the steps and describe the requirements (in C# and in the Unity Editor) for enabling inter-object communication in this case. |
|  | | **2** |  | I had to get a reference to the GameManager component I wanted to communicate with. I created a serializable GameManager variable and dragged the corresponding component inot the field. I could then use this reference to call a function within the GameManager class. In this case I used the SendMessage function in order to do this. It wasn’t really necessary to use SendMessage in this case as I already had the reference directly to the script, but I didn’t get into a situation where I needed to reference a GameObject instead of a specific Component, so I chose to use it here anyways. |
|  | |  | 1. a) | Choose a Sprite Animation from your project. List the **Parameters** (name and type) that you created in the **Animator** window. |
|  | | **1** |  | For the Player sprite animation, I created the float ‘Speed’, bool ‘IsHit’, and Trigger ‘Died’. |
|  | |  | b) | Provide a rationale for why these types were chosen for creating the desired transitions between animation states. |
|  | | **2** |  | Speed was needed to keep track of the player’s current speed, in order to adjust the rate the moving animation played.  IsHit was used to play the hit animation for a set duration of time based on variables set in the game code, so using a bool to track this was the simplest.  Died only needed to be entered once (upon player’s death), so a Trigger was used. |
|  | |  | 1. a) | Identify an example of linear interpolation (Mathf.Lerp, Vector3.Lerp or Color.Lerp) within your project. |
|  | | **1** |  | I used linear interpolation to animate the hearts in the UI. |
|  | |  | b) | Explain why linear interpolation was an appropriate choice for changing a value over time. (As opposed to other ways of changing values, such as simply adding a constant to a variable). |
|  | | **2** |  | It was appropriate for this type of scaling animation, as it was simple and allowed for easy adjustments using an animation curve. I wanted it to go between two set values at a variable rate. |
|  | |  |  | Choose 1 dynamic UI element within the Unity Project. Describe and list the requirements (in C# and in the Unity Editor) for updating the UI element at runtime. |
|  | | **2** |  | The Hearts were updated when the player was hit. An array of references to the hearts was given to the GameManager script, which determined which heart to play the animation on. If the player was hit, the GameManager could then use this array to deactivate and remove the next heart from the screen, and begin animating the next one after.  For text elements a reference to the text component of the object was given to the necessary scripts. |
|  | | | | |
|  | | **14** |  |  |