**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** |  | Here, then the mouse is released, I get the direction from the cat to the current mouse position by subtracting the cat’s position from the mouse position and normalizing it.  I then set the magnitude to the power level by multiplying the direction vector by the power level.  This creates a force that I can send over to the cat for it to be launched in the direction of the mouse. |
|  | |  | 1. a) | Identify an example of inter-object communication within the Unity project. Explain why this inter-object communication was desired or necessary. |
|  | | **2** |  | (in the controls/game manager)    (in CatController)    I use inter-object communication to send forces to the cat so that it launches itself.  This ensures that the cat’s script is in full control of its own motion, and to have tidy private function that can be called from anywhere outside of the script. |
|  | |  | b) | List the steps and describe the requirements (in C# and in the Unity Editor) for enabling inter-object communication in this case. |
|  | | **2** |  | First, in the controller script I creates a serialized field ‘cat’ so that I could draw the cat object into it    Then, I created the function ‘Launch’ within the CatController script.  And then I simply called cat.sendMessage and passed “Launch” and the force as the parameters |
|  | |  | 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 cat animation I have three parameters:  bool Airborn (I just noticed it’s spelt wrong),  bool Stunned,  and float YVelocity |
|  | |  | b) | Provide a rationale for why these types were chosen for creating the desired transitions between animation states. |
|  | | **2** |  | A picture containing text  Description automatically generated  Airborn is a boolean because I have an idle animation that only plays while the cat is stationary and on the ground, so I needed a flag that was either true or false  Stunned is also a Boolean because while airborne once the cat has hit a wall, it’s animation turns from a graceful jump and roll to a sad out of control cat, so I also needed it to be a flag that could be true or false.  And lastly I needed to know the YVelocity to know when to switch from the jump to the roll animation |
|  | |  | 1. a) | Identify an example of linear interpolation (Mathf.Lerp, Vector3.Lerp or Color.Lerp) within your project. |
|  | | **1** |  | Here, I use lerp to determine the power level while the mouse is pressed |
|  | |  | 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** |  | I used it here because I wanted the rate that the power level increases to be dynamic,  When you’re tossing your cat rarely would you need the accuracy when going for short throws, and that’s the same if you’re going for more power, if you’ve held it for that long you’re probably going for full force, but in the middle is where you want a bit more accuracy so the power increases fast at first, slows down, and then speeds up right at the end    Something like this would have been very difficult to do with math alone so an animation curve is a perfect use |
|  | |  |  | 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** |  | During the game I have a slider at the bottom of the screen that shows the power level while the mouse is pressed.  I used a slider, and to update it in the Controller script I used a serialized field for a slider    In start I set its min and max values to be the min and max power values (which are also serialized fields)    And to update it, when the mouse is pressed, I set its value to be the current power level |
|  | | | | |
|  | | **14** |  |  |