Programming Assignment 2: RPG Basics (Preliminary Handout)

Handed out: Thu, Apr 12. Due: (tentative) Mon, Apr 23, 11:00pm. Late policy: up to 6 hours late: 5% of the total; up to 24 hours late: 10%, and then 20% for each additional 24 hours. The submission procedure will be the same as in the first programming assignment (see below).

Overview: The objective of this assignment is to learn more about Unity through the design of the basic structure of an RPG-style game (but not a full game). This includes importing character and object models into Unity, using navigation meshes to control movement, scripted camera control, transitioning between animations using Unity’s animation controller system. To make your task simpler, we have based the principal elements around the Brackeys/Lague RPG Tutorial (Videos E01, E02, E03, and E10 particularly):

https://www.youtube.com/watch?v=nu5nyrB9U0k&list=PLPV2KyIb3jR4KLGCACiWQ5qHudKtYeP7

Caution: Note that it is possible to download the complete project and models from the resources provided with the tutorial. I would caution against this because: (1) you will learn more by following the step-by-step instructions in the video and testing them as you go and (2) there are many things in the final project that you do not need to implement, and figuring out what to delete and what to keep is quite complicated. (Believe me—I tried!) The Brackeys tutorials are very useful because he is quite proficient in Unity scripting, and you can learn a lot by simply following along as he develops his programs.

Here are the major project components:

Environment: The environment consists of two principal elements, a ground surface upon which the characters move and obstacles that the characters should avoid. You can either use the environment model provided in the tutorial or build your own. The credit you receive will depend on which option you choose. (We will provide more specific information soon. For maximum credit, your ground and obstacles should be generated on your own, e.g., using Unity’s terrain editor, procedurally generating your own terrain, or generating your own terrain and obstacles using a modeling tool, like Blender, and importing it. You may use the models provided in the tutorial for partial credit.)

Player: The player character is a humanoid character, of your choosing. (You may use the character provided in the tutorial or import one either from the Unity Asset Store or from some other resource, such as Adobe’s Mixamo Store. For extra credit, build your own, e.g., using Blender. This is a good deal of work though!) The player character must support at least three modes of navigation, idle, walking, and running. (You may select others if you like, but at a minimum you need to support the ability to stand still, move at a standard speed, and move at a rapid speed).

Following the tutorial, the player navigates about the environment using a click-to-move approach. Left-clicking indicates a point on the ground to which the player should start walking (standard speed). Unlike the tutorial, however, we will use a different mechanism for changing speed. While the user holds down the Shift key, the player character switches into running mode (rapid speed).
**Enemy:** The player character is chased by a non-player character, called the *enemy*. (You may have more than one if you like.) The enemy is also a humanoid character, and it should have a different appearance than the player character. It need only support two animations, idle and walking. As in the tutorial, whenever the player character comes sufficiently close to the enemy character, the enemy starts chasing the player character. As with the player, the enemy uses the navigation mesh to move. If the player can move to a sufficient distance, the enemy stops chasing the player.

**Camera Control:** The camera control should be the same as in the tutorial, following the player from a fixed direction, but the user can zoom in and out, can pan around the player character, and (in addition to the tutorial) can crane up and down. (More details will be provided soon.)

**Online Resources and Tutorials:** In addition to the Brackeys/Lague RPG tutorial, there are a number of online resources that were helpful in designing our implementation:

- [https://unity3d.com/learn/tutorials/topics/navigation/navmesh-agent](https://unity3d.com/learn/tutorials/topics/navigation/navmesh-agent) The Unity video tutorial on NavMeshes. Also check out the videos under the section “Live Sessions on Navigation”
- [https://docs.unity3d.com/Manual/Navigation.html](https://docs.unity3d.com/Manual/Navigation.html) The Navigation section from the Unity manual. Check out the subsection on “Navigation How-Tos” especially “Moving an Agent to a Position Clicked by the Mouse” and “Coupling Animation and Navigation.” (Note that with the simple animations that I used, I had some difficulty getting the method described in “Coupling Animation and Navigation” to work
- [https://www.youtube.com/watch?v=...rest of link omitted](https://www.youtube.com/watch?v=...rest of link omitted) A tutorial by Holistic3d entitled “Mechanim & Mixamo in Unity 5”. This explains how to create and download a Mixamo model and transition between animations. (I would recommend using this method for our project, rather than the blend-tree approach that is discussed in the Brackeys/Lague tutorial.)