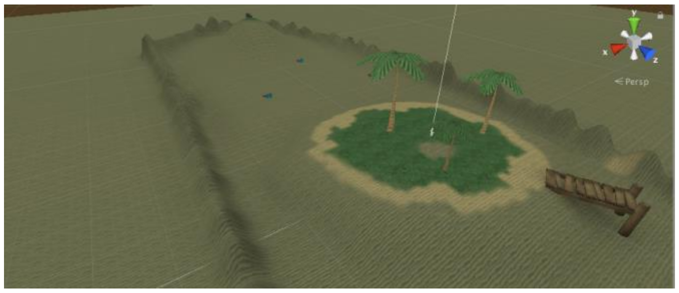
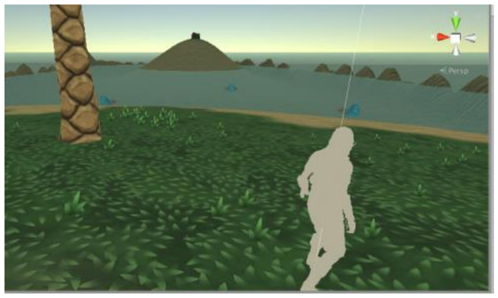
CMSC 425 : Spring 2019

Programming Assignment 2: Crab Soup, Anyone?
Part A: Navigating a Terrain

Due: Monday, April ~~1st~~ **5th** at midnight **Worth:** 50 points total

Project 2 overview: Your main character is a treasure hunter who starts on one island and sees in the distance a second island with a chest full of treasure. You know that the key to the chest is somewhere underwater around this island and you must dive underwater and swim to find it. Unfortunately, the local crab population has mutated and is very hostile. Avoid the crabs, find the key, open the chest, get back safely and receive your reward.





Purpose: We'd like you to use more features of Unity, practice those for your final project, and work with concepts from lecture. There are four major new elements to this assignment:

Part A: Terrain and navigation. The focus of this part is to create a terrain, and agents to navigate it
1. *Terrain* – The Project 1 maze will be replaced by a terrain created by the Unity terrain editor.
2. *NavMesh* – You'll add a NavMesh to your terrain to for the mutant crabs to navigate.

Part B: Character animation. The focus of this part is to animate the player and NPC agents.
1. *Mixamo* – You'll find and import existing character models and animations for your agents.
2. *Animation blending* – You'll blend idle, walking and running (or swimming) animations for the player.

Requirements: In more detail, for Part A you should:

a. Use the Terrain editor to create islands, an outer reef, and other elements as you want.
b. Add wall colliders around the outside to make the reef into an arena that can't be exited.
c. Paint the Terrain with appropriate textures, and include some obstacles that are set unwalkable.
d. Add a NavMesh to the Terrain.
e. Add a Player with a collider and a controller to move around. Can be a ball or fixed model.
f. Add three NPC agents to chase the Player when close, and a script to lose the game when they touch.
g. Add a treasure chest for the player to find.
h. Attach actions to the chest so when the player gets to it without losing, they win.
i. Add the Asset store "water" to create an underwater effect based on a water area.
j. Include a skybox to make the background interesting.

We invite you to improve the game play, to modify elements within the spirit of the project, to add elements if they make the game more interesting and keep the spirit of the project, and otherwise have fun. And work towards your final project (eg, if you want specific style of chasers in your final project, feel free to use this project to explore that.) A specific bonus item is to implement an "air" system so the player has to come to the surface for air after a certain amount of time underwater (eg, in the water region). You may want to add trees, underwater plants, etc. You're welcome to start on the animations for Part B. We'll get instructions out later.

Details and comments:
A number of elements from the earlier programming assignment should be included as well. These include:

- *Camera:* The camera should following the player, making smooth rotational transitions
- *Winning/losing:* There should be some indication of winning/losing, for example, by freezing the game and displaying an appropriate text message
- *Transitions:* You should implement transitional commands, including 'R' for Restart, 'Q' for Quit, 'P' for Pause/Unpause, and 'C' to disable the chaser. You do not need to include a start menu.

New considerations include:

- Part of the gameplay design is picking appropriate speeds for the player and crab characters. If the crabs are too fast they catch the player too often – if they're too slow, the game will be too easy. The player should be faster, but you set the ratio.
- If the terrain is open as in the example implementation, the chase may not be interesting. Water with a fog to obscure things, or obstacles, can help.
- You may use physical assets from the Unity asset store, or elsewhere. You should avoid just copying scripts and try instead to understand and reproduce them. In any case you should document where assets and ideas came from.
- There's multiple ways to implement water, and we'll be happy with any effective approach. We've included a tutorial video below. You can use a water asset. To know a character is in water, consider their y value, or consider how you might add water areas to the navmesh.

Resources: An overall introduction that might help to Project 2, Parts A and B, is the Brackeys/Lague RPG Tutorial (Videos E01, E02, E03, and E10 particularly), although you don't need to create your figures from scratch in Blender. https://www.youtube.com/watch?v=nu5nyrB9U_o

- Terrain
  - https://docs.unity3d.com/Manual/terrain-UsingTerrains.html
  - https://www.youtube.com/watch?v=x40A4TkqsUk

- NavMesh
  - See the Unity video tutorial on NavMeshes. Also check out the videos under the section "Live Sessions on Navigation
  - https://unity3d.com/learn/tutorials/topics/navigation/navmesh-agent
  - The Navigation sections 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."
  - https://docs.unity3d.com/Manual/Navigation.html
  - https://docs.unity3d.com/Manual/nav-BuildingNavMesh.html
- Water
  - https://www.youtube.com/watch?v=FoZwgRE5LYI

Submission: Submission will be by uploading a file through ELMS. We will be giving you more

Group project: To make this feasible as it overlaps work on your final project, this will be a group project which we recommend you do in final project groups.

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. Submission instructions will be given later.

Don't Panic: Yes, this description is long. But the code is not as long as you might think. (There are just lots of little details.) Since there are many parts, you can get partial credit even if all the features are not implemented.