Final Project

Due: Dec 12, 2019 11:59 PM

The aim of this project is to add basic functionalities to your Game Engine.

Overview

This project will have you expand on functionalities implemented in your game engine, and have you upgrade existing, or add extra functionalities from variety of options suggested. If you wish to add any specific component or service to your engine which is not listed below, you may do so after consulting with the instructor. Finally, a minimal showcase of the functionalities added to your engine must be included in your final submission. The show cases can be separate or bundled in one game.

Requirements

1. Camera system Translation
2. Up to 3 of the following options (+1 for extra credit) Per-group member
    - Camera system scaling
    - Camera system rotation
    - Circle colliders
    - Box colliders
    - Circle colliders interaction with box colliders (needs the last two implemented)
    - Audio System
    - Lighting (needs one collider type implemented)
    - Physics (counts as 2, must implement at least one type of collider)
    - Game Object hierarchy system (what we talked about today)
    - Animation
    - Animation Graph (needs the last option implemented, but also automatically handles changing the animation, counts as 2)
    - Custom (Only one, consult the instructor)

Camera System Transition:

Final project will be using AffineTransform to keep track of position of objects. Create an affine transform instance as the location of your camera, using its inverse matrix to calculate the position of every other object (encapsulation is recommended.) Your camera needs to be able to move in 4 directions. To showcase, add controls to the camera and allow the user to move the camera. (Difficulty 1)

Camera System Rotation:

Expand the “Camera System Translate” functionalities so it can also represent rotations. To showcase, add controls to the camera and allow the user to rotate the camera. (Difficulty 2)

Camera System Scaling:

Expand the “Camera System Translate” functionalities so it can also represent camera scaling. To showcase, add controls to the camera and allow the user to scale up or down the camera. (Difficulty 1)

Circle Collider:

Make a collision system that can detect collision of two circles. Must extend component class. To show case, have two circles on the screen that collide, and out put the collision recorded. (Difficulty 1)

Box Collider:

Make a collision system that can detect collision of two rectangles. Must extend component class. To show case, have two rectangles on the screen that collide, and out put the collision recorded. (Difficulty 2)

Box Collider:

Make a collision system that can detect collision of two rectangles. Must extend component class. To show case, have two rectangles on the screen that collide, and out put the collision recorded. (Difficulty 2)

Circle Collider interaction with box collider:

Expand your collision system, so it can detect collision of a circles with a rectangle as well. Must extend component class. To show case, have a rectangle and a circle on the screen that collide, and out put the collision recorded. (Difficulty 3)

Audio system:

Adding the capability of loading and playing audio files. For showcase, create a sample program with background music, and have it play another audio file on command (by pressing a key for example.)

Lighting:

Use the ray casting to illuminate a scene. Said scene must include various objects with colliders which can block light (cast shadow.) Removing such object must remove the shadow.

Physics:

Create a physics component that can govern the movement of objects by keeping track of their velocity, friction/drag, mass, and applied forces. Must extend Component class.

To show case, have an object with Physics component and give the user control of its movements (which is now governed by physics class.) (Difficulty 4)

Game Object hierarchy system:

Upgrade the system through which you keep track of game objects' positions, so every object can be broken into parts (children,) where position of each part is expressed relative to the parent object. Example, a hand can be made from 5 fingers (children,) the position of which are stated relative to the wrist (parent.) To showcase, demonstrate creating any figure by instantiating its different parts, and setting only one object as the parent. In your main function, annotate the lines which are instantiating the parts of this collection using comments. Finally, in run time move and rotate the collection by only moving the parent object. (Difficulty 3)

Animation:

Create an animation system that as input, only receives an Array of BufferedImages (frames of the animation) and a double representing the number of seconds it should take to cycle trough that array. This system should then handle the timing for moving through the animation frames. To show case include an animated object in a game. (Difficulty 2)

Animation Graph:

Upgrade your animation system so the developer can create a graph representing different animations used for a specific game object (example, Walking in 4 different directions.) This Method then should allow the developer to set logic to traverse this graph and access the correct animation instance. To show case, include an animated object in a game which changes its animation based on player input. (Difficulty 4.5)

Documentation:

Along your submission include a text file listing the options you chose to implement, as well as the controls required for testing your submission.

Extra Credit:

For extra credit choose a total of 5 options, instead of 4.

Grading

Camera System Translate	20
Each Option	80
Extra Credit option	10
Total	**110 points**

Submission

Submit a zip file containing all your project files to ELMS. Follow this naming format:

<First Name><Last Name>P?.zip

Academic Integrity

Make sure you read the academic integrity section of the syllabus so you understand what you must not do. Note that we check your submission against other students' submissions, and that we are required to report academic dishonesty cases to the University's Office of Student Conduct.