CS 175 | Scripting Languages

Programming Assignment 7 – Final Project

This assignment is about implementing a fully functional (but simple) game using all the game engine managers that were covered in previous lecture (Input Manager, Object Manager, Game State Manager etc…), in addition to the implementation of different game objects, adding Box2D to the engine, adding a camera system and adding sound.

Shooter Requirements:

The game will have minimum a main menu, help, one playable level (asteroid style shooter of course), a win screen and a lose screen.

The playable level should contain the following:

- Main character (Ship)
  - Physics based movement
    - Move forward in the direction you are facing
    - Rotation while moving
    - Caped acceleration
    - Drag that stops the ship if forward is not pressed
  - Collision with the sides of the map
  - Collision with enemies, health goes down
  - Shooting bullets in the direction the ship is facing
- Enemies
  - Collide with the ship and the map
  - Collide with bullets and get destroyed by playing a destroy animation
  - Collide with Main Character
- Bullets
  - Collide with map
  - Collide with Enemies
  - Get destroyed at any collision
- Sounds
  - Background sound
  - Minimum 2 sound effect (when you shoot, when enemy dies, etc...)
- Playable Level
  - Pressing "R" will cleanly restart the "Level"
  - Camera smoothly follows the main character
  - Win/Lose condition

NOTE: Most important thing!!!! I don't want to get any warnings or runtime errors at any point in the game

Code Requirements:

- **Game State Manager (Provided)**

- **Object Manager (Provided)**

- **Input Manager (Provided)**

- **Box2D Integrated (35% of the project)**

    - The ability to add physics object
    - The ability to have a collision reaction when Box2D detects collision
    - The ability to apply forces on objects
    - Most importantly, Box2D objects should work properly with **ALL MANAGERS.** In other words, if we restart a level, physics objects should restart properly etc...

- **Camera that follows the main character (20% of the project)**
    - Only camera translation is required

- **Sound (20% of the project)**
    - Background music that keeps looping
    - Sound Effects (minimum 2)

- **Levels: A minimum of 5 levels/states are required: (25% of the project)**

    - **Main Menu Level**
      The game should start using this state. From here, we can access the gameplay or the help state

    - **GamePlay Level**
      This is the playable level. You don't have to implement a full game here, but it should be able to entertain and be playable for a minute (at least).
      The game should have win & lose conditions.

    - **Win/Lose Levels (2)**
      As mentioned before, the gameplay level should have win and lose conditions. The game will switch to either the win/lose states accordingly

    - **Help**
      This level should be accessible from the main menu. It should contain a description of how to play the game, along with the controls.

PS: Submit a clean version of the game. In other words, I don't want the levels that were done in previous assignments.

- **Bonus points will be awarded for implementing: Particle Systems (25% of the project)**
    - One particle effect should be a reaction to an action in the game (Bullet collided with enemy, smoke trail behind rocket etc…)
    - One particle system alive at all time during the level
    - Particle systems should reset / be destroyed cleanly. In other words, they work as part of our engine.

Comments

No commenting!

What to submit

You must submit the (**Assignment 7**) folder in a single .zip file named correctly (go to the class page on moodle and you will find the assignment submit link). **Do not submit any other files than the ones listed.**

If you've forgotten how to submit files, the details about how to submit are posted in the syllabus. Failure to follow the instructions will result in a poor score on the assignment (and possibly a zero).

Special note:

The due date/time posted is the positively latest you are allowed to submit your code. Since the assignments can easily be completed well before the deadline, you should strive to turn it in as early as possible. If you wait until the deadline, and you encounter unforeseen circumstances (like being sick, or your car breaking down, or something else), you may not have any way to submit the assignment on time. Moral: **Don't wait until the last day to do your homework.**