Programming Project 3

Due: Tue, Oct 14. (Design document due Oct 7)

Objectives:

- Learn how to implement a simple real-time simulation.

You have two options:

1. implement physics for the Jetpac game
2. design your own physics based game

General requirements:

Your game must involve

- some continuously working force (like gravity)
- some way for the user to directly or indirectly manipulate the forces working on objects in the game (E.g., in Jetpac the user can control the jetpack and thereby directly influence the forces working on the player character. In Spin the Black Circle the player can change the rotation of the wheel and thereby manipulate how the force of gravity is working on the ball.)
- collisions between a moving objects and a stationary object
- collisions between two moving objects (You only need to model the linear effects. You do not need to account for rotation.)

General strategy:

Implement a class called `PhysicalBody` to represent physical properties of objects in your game. It will have fields for position, size, velocity, acceleration, mass and forces acting on the body and it will have all the methods needed to update these fields appropriately. That is, this `PhysicalBody` class will take over some of the functionality that is in your Sprite class right now. The `Sprite` class is where you keep track of the texture and what part of the texture to draw and will be responsible for drawing only. Each physical body in your game that needs to get drawn to the screen will thus be represented by an object of type `PhysicalBody` and will also be associated with an object of type `Sprite`.

About option 2:

Option 2 (designing your own game) is obviously more work than option 1 (reimplementing an already existing game with some modifications). I am going to recognize the extra creativity and effort that people put in when choosing option 2 by giving out bonus points.

Requirements for the Design Document:

Your design document should have three parts:

Your design document should contain a description of the game you going to implement. If you choose option 1, describe what exactly your world is going to look like, what objects will be in it, and how the player/player character interacts with these objects. If you choose option 2, give a more general description of your game (what is the setting, the goal, how does the player interact with the game), then also describe what your world is going to look like, what objects will be in it, and how the player/player character interacts with these objects.

The second part of your design document should be a specification of the `PhysicalBody` class. Give member variables and methods and their signatures, like you did for the Sprite class design document.

The third part: in high level pseudo-code, write down how user input is handled and the state of your game gets updated.

Grading:

The grades will be based on the following criteria:

- whether the basic requirement of implementing a game about physics is met
- whether all requirements given above are met
- the correctness of the physics implementation
- the overall correctness of the code
- the overall organization of the code
- whether the code is appropriately documented (through comments)
- the quality of the design document

As I said above, I will give bonus points for effort that I think goes beyond the expected, such as: your own game designs, particularly polished games or nice graphics, particularly innovative ideas, …. (This is an incentive for you to take what you have learned in class and play around with it. I believe that only if you do that, you will really learn how to use it effectively.)