Dungeon Diver

https://github.com/comp195/spring-2021-final-project-dungeon-diver

Ethan Lo: e_lo2@u.pacific.edu

Last Revision: 2/7/2021

System Architecture

Software Modules

This system should not need any other software to run.

Hardware Components

This system will only need to run on one computer.

User Interface

The system will have a simple user interface.

Interfaces to External Systems

This system will not need to have connections to external systems.

Hardware, Software, and System Requirements

Minimum:

OS:

-Windows 7

-Mac OS X 10.6+

Processor: Intel Core 2 Duo

Memory: 2GB RAM

Graphics: GeForce 7600 GS

Storage: 2 GB available space

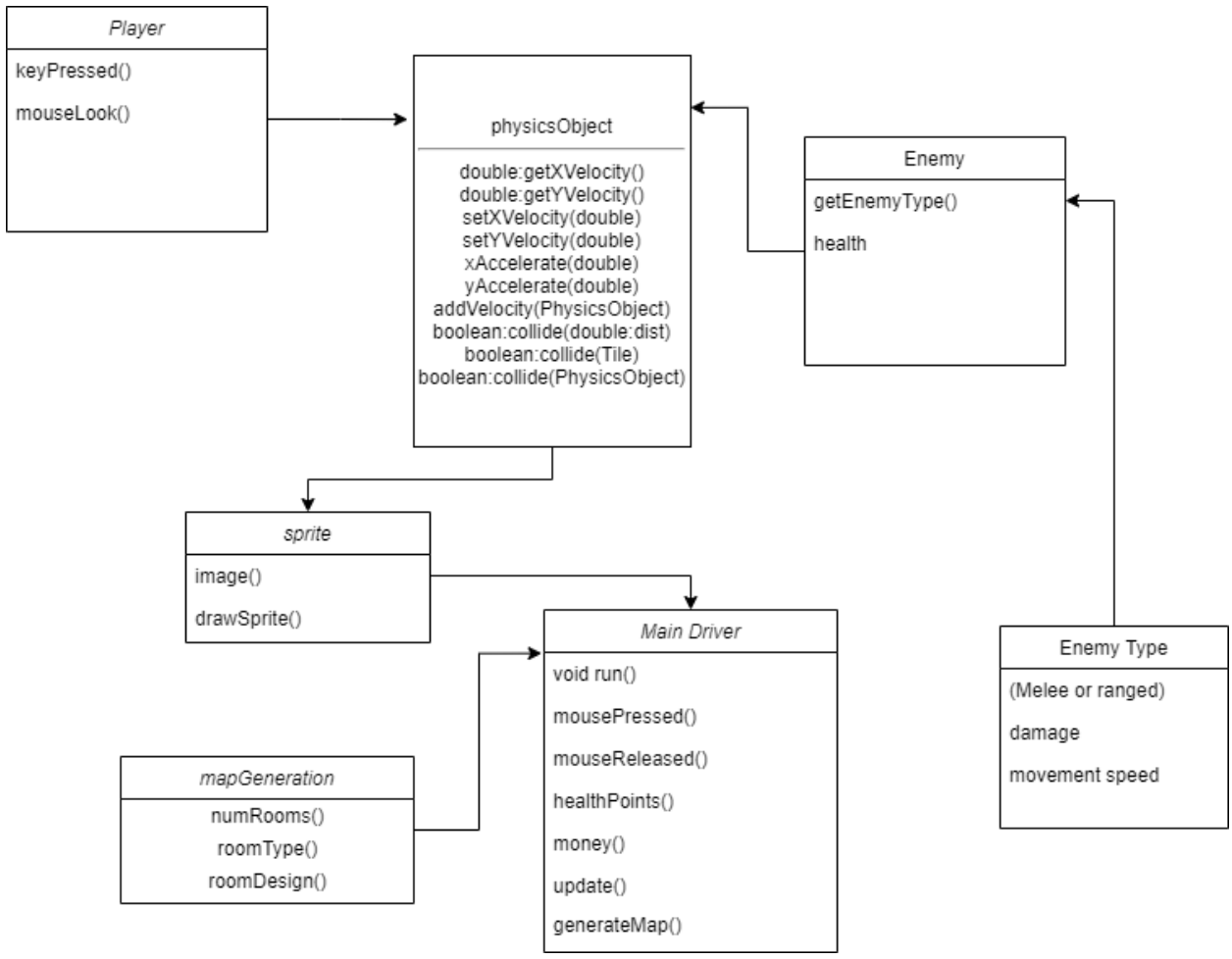
External Interfaces

None

Software Design

For now, this is a barebones diagram. It will become updated and more fleshed out as I code.

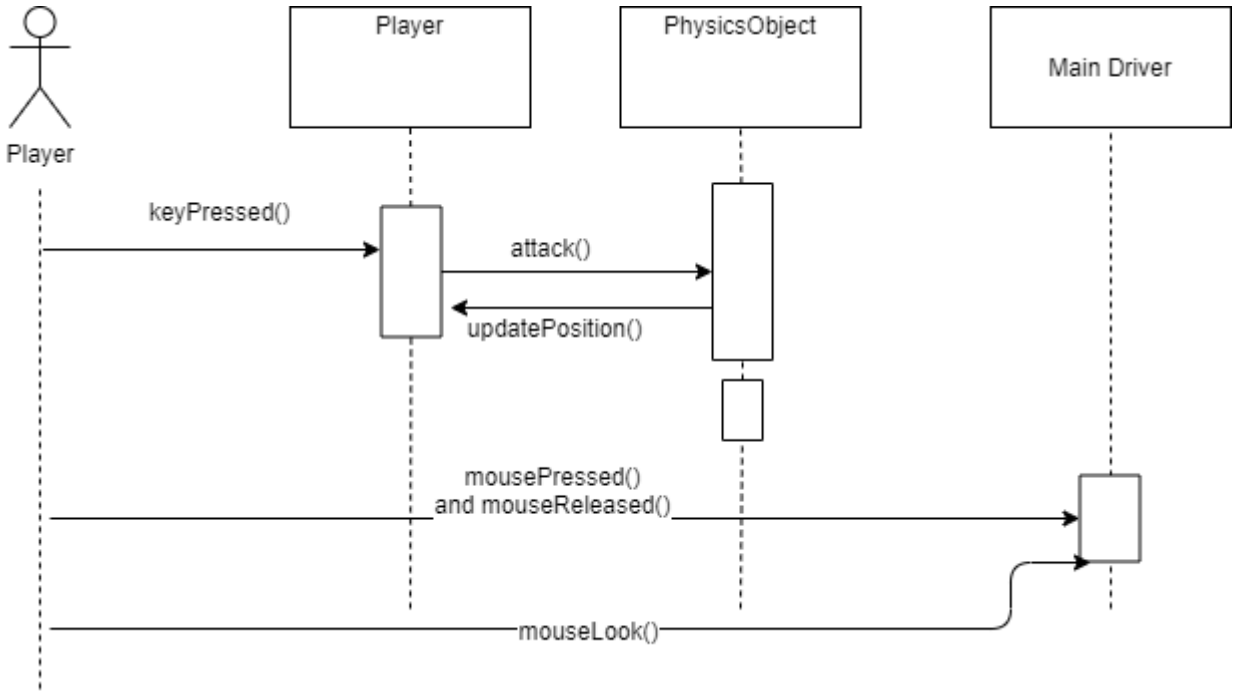
Class Diagram & Class Specifications



The player and enemy are both physics objects that are updated each frame in the main driver. There are different enemy types, and thus each enemy or enemy type has its own parameters of damage and speed. The player will move with keys, and look around with the mouse. Since this is a top down game, the player sprite will change when the mouse moves directions, making the player sprite look up, down, left, or right.

Interaction Diagrams

This is a very simple and barebones diagram. I will try to update it as I go. For now, it just shows a simple player interaction. When the player attacks an enemy, which is a physicsObject, it will cause the enemy to recoil back (forgot to include in diagram). The same happens if an enemy hits the player. The player will also be able to dodge and dash. The player uses the mouse to look in different directions.



Design Considerations

Since this game is randomly generated, that means that the design of the maps and levels are not the main focus. I plan to have simple maps with simple designs generated, thus allowing me to focus more on the actual gameplay. However, with that being said, it is essential that the random generation works in the first place.

User Interface Design

The user interface will be simple and easy to understand. Because this game is randomly generated, there will be no level selecting. The options in the menu would be something like: Start, Option, and Exit. I am debating on whether or not to allow users to save their progress.

A similar game I would liken this to is called *Risk of Rain*. In this game, you only have one life. If you die, you start the game over from scratch, but all the levels are in different order and in slightly different design. However, I am expecting my game to have users take longer to clear levels. Thus, I might enable the user to save their progress.

Once you start playing, I am planning to have a simple UI. The picture below is a great

example as to what my game should be like. You can clearly see what you need to know, such as life and money. However, the user interface itself does not clutter the view, allowing the user to clearly see the game. What is different however is that the player will be able to look in all directions, as opposed to only looking up, down, left, or right. You will still be able to use WASD to move around however. ESC to pause.

Glossary of Terms

Top Down

– Game Genre in which the camera's point of view is looking "down" from the sky

References