Java Game

David Nyberg, Ryan Davis, Ryan Cooper, Charlie Davies

Vision

We started our project with the vision of an Android Studio mobile game.

After getting started and hitting many road blocks trying to learn Android Studio we switched plans to implement our own MVC to develop our Java game.

We wanted a simple ‘shooter’ type game where you control your player and can shoot at enemies while moving around the screen.

Design Patterns

MVC: The Model View Controller we implemented involves all of our game objects and our database connection.

- The Model package contains the structure and data (including database access)
- The View package has all of our GUI components (menus and canvas)
- The Controller package contains the brains of our code (opening the game, displaying objects, keyboard input)

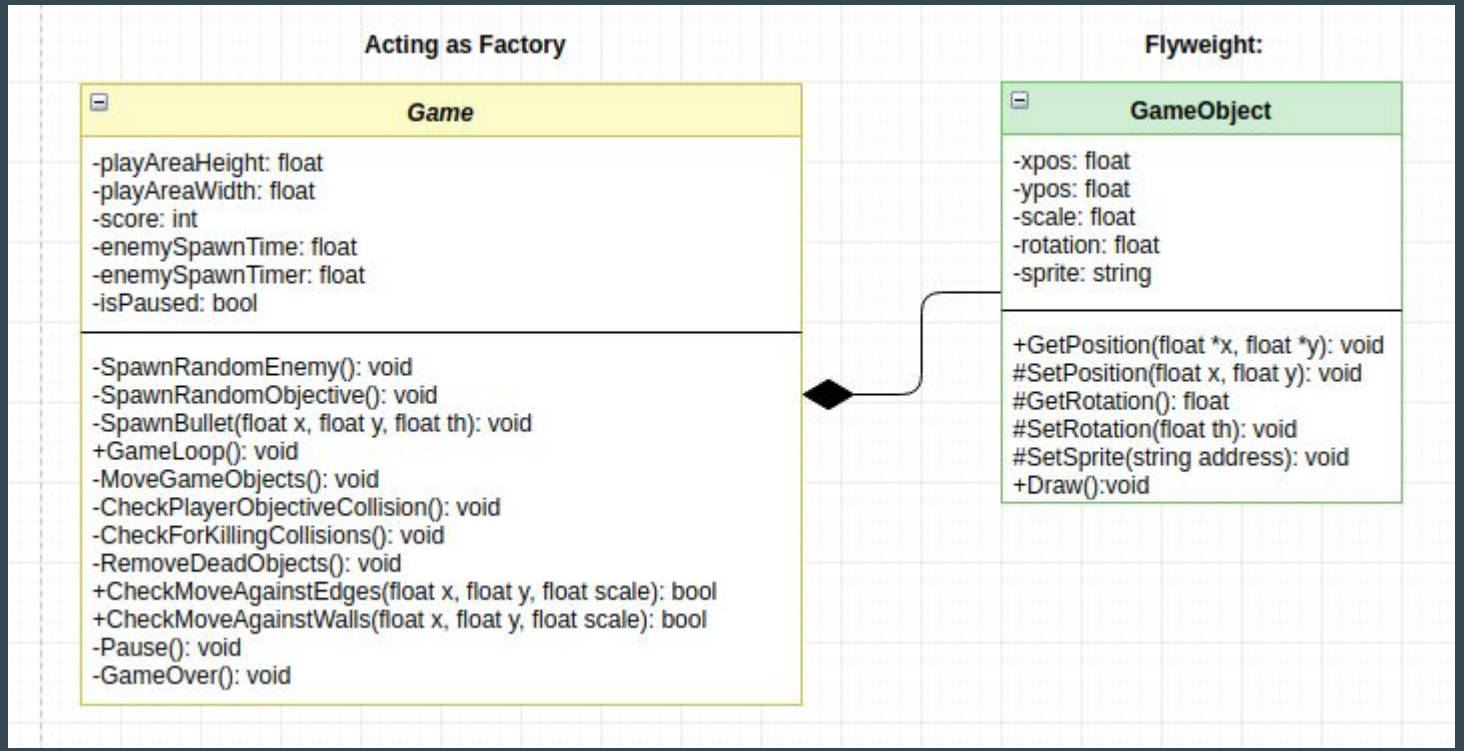
- All classes are involved, not going to show entire class diagram

Design Patterns

Flyweight:

Our main game controller controls the flyweight pattern that will spawn everything that will go into the game view. The player, enemies, bullets, and walls are all subclasses of a gameobject that all contain similar methods and attributes making efficient use of the flyweight design pattern. Additionally, they all reference the same image file.

Design Pattern: Flyweight



Demo

GitHub:
https://github.com/davidnyberg/OO-Project/blob/master/OOPGameDemo.mov

Google Docs:
https://drive.google.com/file/d/1NdIE-5kpN7AHXkGUN05xq3uGYP8amyEC/view?usp=sharing

Use Cases

Use Case ID:	#01	
Use Case Name:	Move Players Character	
Description:	The player can move the character in order to move towards treasure,	
	move away from enemies, or whatever direction the player chooses.	
Actors:	Player	
Pre-Condition:	Player has started game. Player has not died. Player is not trying to move in a	
	direction obstructed by a wall or off the edge of the map	
Post-Condition:	System moved the player to its new location and displays it	
Frequency of use:	Almost constantly by player while in-game	
Flow of Events:	Actor:	System:
	1. View UI to decide where to move	Move the player in the correct
	2. Press button to move in desired direction	direction and update the display
Variations:		
Exceptions:	If moving will hit a wall or edge of the map, do not move	
Notes:		

Use Cases

Use Case ID:	#02	
Use Case Name:	Player kills enemy to gain points	
Description:	The player shoots and the projectile hits an enemy	
Actors:	Player	
Pre-Condition:	Player has started game. Player has not died. Player is able to shoot	
	and the projectile can hit an enemy	
Post-Condition:	System updates score and the enemy disapears	
Frequency of use:	Very often by player while in-game	
Flow of Events:	Actor:	System:
	1. View UI to decide where to shoot	Display projectile in right direction
	2. Press button to shoot in desired direction	enemy gets hit and is removed
		update the score on display
Variations:	Hit a wall, projectile dissapears	
Exceptions:	If projectile misses enemy then do nothing	
Notes:	The projectile can be fired in any direction and can hit anything	