Assignment 3

10 oktober 2016 - Group 666

Exercise 1 - Design patterns

1. Why and how

We implemented the builder (factory) pattern to make the creation of entities more flexible, since they are often pretty complex and variable, and we want to be able to easily change their behaviour in the future.
We implemented the Observer pattern because our code used to work with a list that would be checked every tick for entities that died or got created. Since this isn't very elegant, we decided to implement observers that would notify the game upon creation or death of an entity.

2. Class diagram
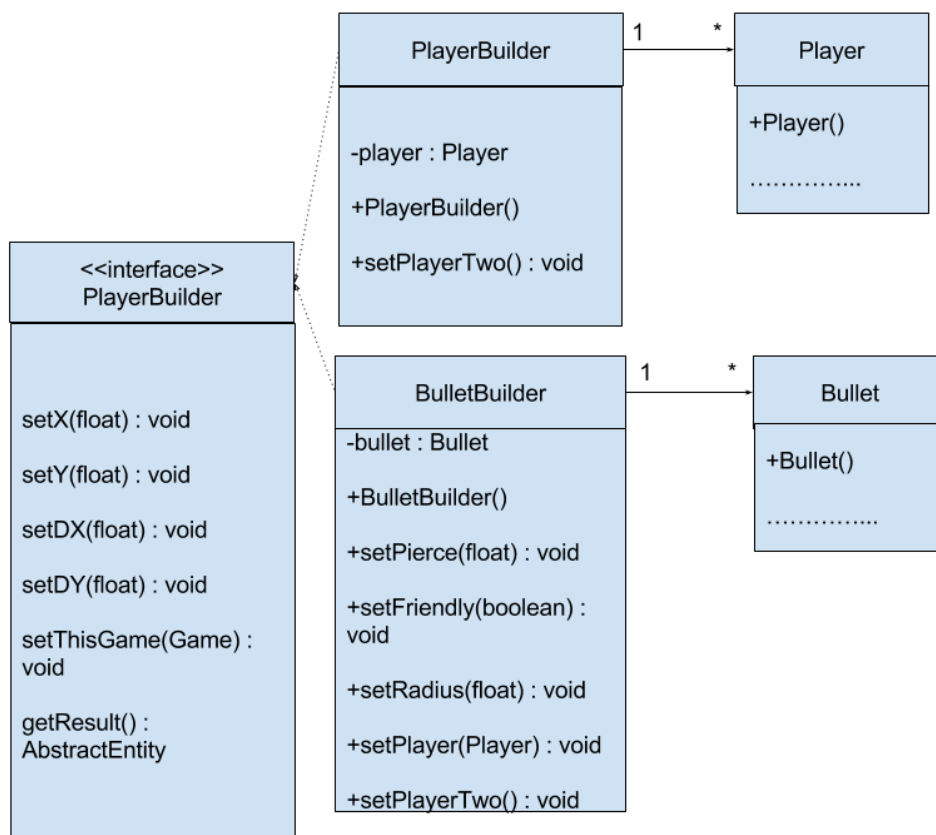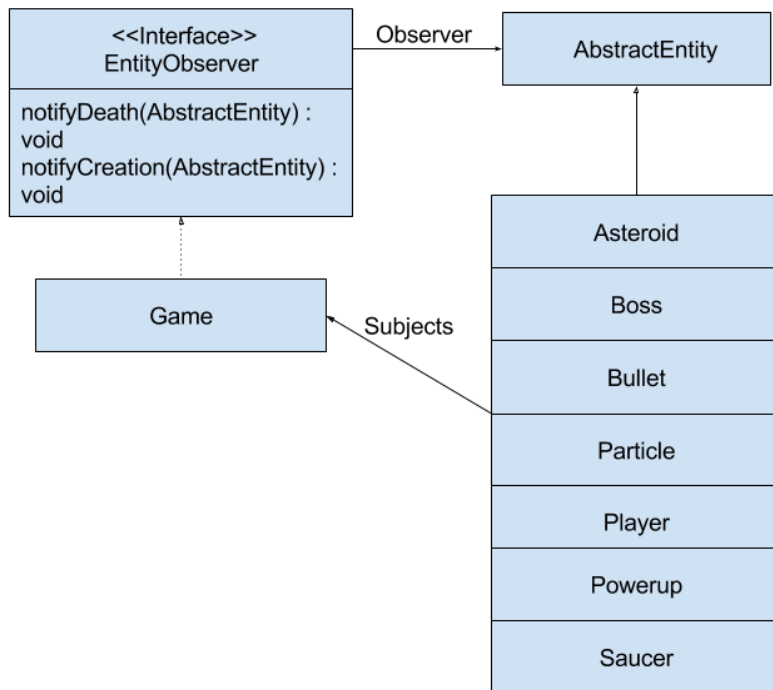
3. Sequence diagram

Exercise 2 - Your wish is my command

1. Requirements

- The responsibilities of the Game class shall be split up.

2. RDD & UML

Exercise 3 - 20-Time

1. Requirements

Functional requirements

Must haves:

- A special boss mode shall allow a player to fight a boss immediately.
- The boss shall shoot hostile bullets in the direction of the player.
- If the player touches the boss, the player shall die.
- The boss shall have multiple lives.
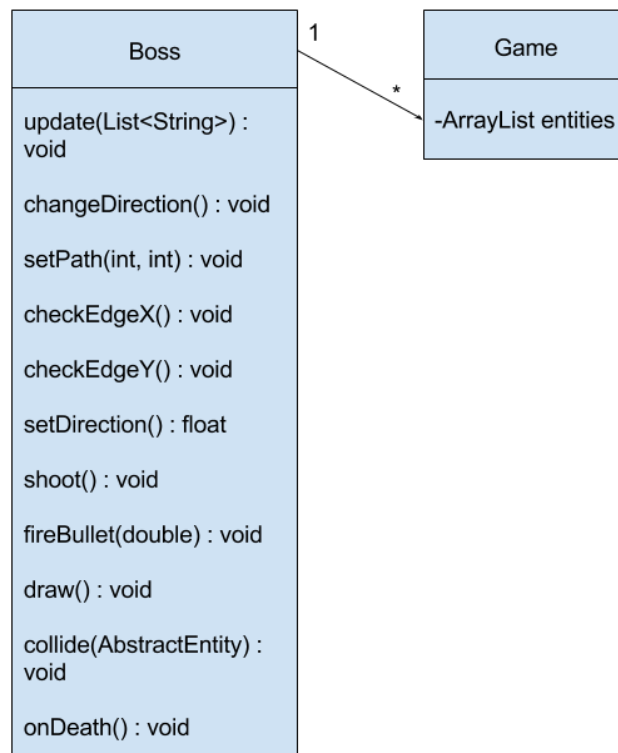- The boss shall stay on the screen.

Should haves:

- In certain modes, the player shall be confronted with a boss after a period of time (for example, a certain amount of waves).
- If the player kills the boss they shall be rewarded with 20000 points.

Could haves:

- The game shall have different types of bosses that behave and look differently.

2. RDD & UML

Boss	1	Game
update(List<String>) : void		-ArrayList entities
changeDirection() : void	*	
setPath(int, int) : void		
checkEdgeX() : void		
checkEdgeY() : void		
setDirection() : float		
shoot() : void		
fireBullet(double) : void		
draw() : void		
collide(AbstractEntity) : void		
onDeath() : void		

Boss is a part of entities, a private ArrayList belonging to the Game class. Boss is also a subclass of AbstractEntity.

Obvious classes	Uncertain candidates	Nonsense
Boss	Spawner	
	Game	
	GameState	
	DisplayText	
	DisplayEntity	

Comparison

Spawner had to be changed slightly to allow it to use the new Boss class. DisplayText and DisplayEntity had to be changed to allow them to display the text for the boss battle mode and the boss itself respectively. Game also required slight tweaking to account for the possibility of bosses. The addition of a boss battle mode also required the addition of a new mode in GameState.