# Assignment 1.1: The Core

## 1.1.1

Keywords

**Game** – game
**Player** – playable character
**Ball** – ball
**Projectile** – projectile
**Level** – level
**Wall** – wall, ceiling, moving ceiling
**Timer** – timer
**Pickup** – currency
**PowerUp** – power-ups
**MainMenu** – main menu
**Button** – option to start the game, button
**Door** – doors
**Sound** – music, sound effects
**Logger** – logger

Other classes not derived from keywords

**GameObject** – Superclass for all objects in the game, such as **Player** and **Ball**.
**Keyboard**

| Class Name: | Game |
|---|---|
| Superclasses: | |
| Subclasses: | |

| Responsibilities | Collaborators |
|---|---|
| Should call updates on all objects in the game | GameObject |
| Provide information about objects in the game | Player, Ball, Pickup |

| Class Name: | GameObject <Abstract> |
|---|---|
| Superclasses: | |
| Subclasses: | Ball, Door, Pickup, Player, Projectile, Wall |

| Responsibilities | Collaborators |
|---|---|
| | Ball, Door, Pickup, Player, Projectile, Wall |

| Class Name: | Player |
|---|---|
| Superclasses: | GameObject |
| Subclasses: | |

| Responsibilities | Collaborators |
|---|---|
| Modify and store information about the player character | |
| Handle physics and collision about itself | Ball, Wall |
| Act upon keyboard inputs | Keyboard |
| Summon projectiles | Projectile |

| Class Name: | Ball |
|---|---|
| Superclasses: | GameObject |
| Subclasses: | |

| Responsibilities | Collaborators |
|---|---|
| Modify and store information about a ball | |
| Handle physics and collision about itself | Player, Wall |

| Class Name: | Projectile |
|---|---|
| Superclasses: | GameObject |
| Subclasses: | |

| Responsibilities | Collaborators |
|---|---|
| Modify and store information about a projectile | |
| Handle collision with balls | Ball |
| Be summoned by the player | Player |

| Class Name: | Level |
|---|---|
| Superclasses: | |
| Subclasses: | |

| Responsibilities | Collaborators |
|---|---|
| Store information about a level | Ball, Door, Player, Wall |
| Provide information to classes requesting it | |

| Class Name: | Wall |
|---|---|
| Superclasses: | GameObject |
| Subclasses: | |

| Responsibilities | Collaborators |
|---|---|
| Store information about a wall | |

| Class Name: | Timer |
|---|---|
| Superclasses: | |
| Subclasses: | |

| Responsibilities | Collaborators |
|---|---|
| Keep track of game time | Game |
| End Game when out of time | |

| Class Name: | Pickup |
|---|---|
| Superclasses: | GameObject |
| Subclasses: | PowerUp |

| Responsibilities | Collaborators |
|---|---|
| Store information about a pickup | |
| Remove itself from the game if picked up | Game, Player |

| Class Name: | PowerUp |
|---|---|
| Superclasses: | Pickup |
| Subclasses: | |

| Responsibilities | Collaborators |
|---|---|
| Grant special powers to the Player or the Projectile | Player, Projectile |

| Class Name: | Door |
|---|---|
| Superclasses: | |
| Subclasses: | GameObject |

| Responsibilities | Collaborators |
|---|---|
| Open when certain Balls are destroyed | Game, Ball |

| Class Name: | Keyboard |
|---|---|
| Superclasses: | |
| Subclasses: | |

| Responsibilities | Collaborators |
|---|---|
| Provide Player with keyboard inputs | Player |

There are certainly some differences between the CRC cards shown above and our actual implementation, but overall, it is pretty close.
We have, for example, a GameLoop class instead of a Game class storing information about current objects in the game and calling the update methods on all the objects. Our Ball is called Bubble, and pickups and power-ups have not been implemented yet.

## 1.1.2

### GameLoop

| **Responsibilities** | **Collaborators** |
|---|---|
| Calls for updates in all objects in the game | Ball, Player, Pickup, Projectile, Wall |

### Player

| **Responsibilities** | **Collaborators** |
|---|---|
| Store information about the Player | |
| Move the Player | Keyboard |
| Check collisions with the Bubbles | Bubble |
| Check for pickups within range | Pickup |

### Bubble

| **Responsibilities** | **Collaborators** |
|---|---|
| Store information about the Bubble | |
| Move the Bubble | |
| Check collisions with the Walls | Wall |
| Do gravity physics | |

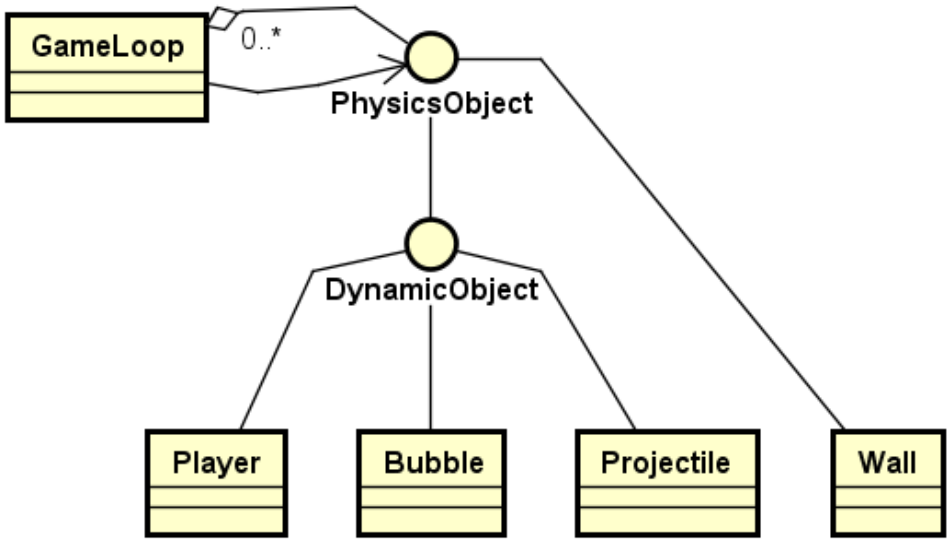
### Projectile

| **Responsibilities** | **Collaborators** |
|---|---|
| Store information about the Projectile | |
| Move the Projectile upwards | |
| Check for collisions with Bubbles | Bubble |

## 1.1.3

We have picked these classes as the *main* classes, because these three game objects are at the core of the game, together with the GameLoop, which is responsible for keeping the time and update all objects once every $1/60^{th}$ of a second.
Classes with less responsibilities are the JavaFX page controllers, for every state of the game, like Main Menu, Options and Game screens. I view these as less important classes because they only exist to be able to navigate the different screens of the game, not playing the game itself.
We cannot merge these classes into other classes, because they are very distinct classes to make the UI function and to put the methods contained in these classes in other classes would be misplacing them.

### 1.1.4


GameLoop
0..*
PhysicsObject
DynamicObject
Player
Bubble
Projectile
Wall

## 1.1.5

A diagram showing an update call on the Player. I chose to make this one into a sequence diagram because it shows the most interaction between main classes as possible in our project.