CS102	**Spring 2021/22**	Project Group	3F
Instructor:	**Uğur Güdükbay**		
Assistant:	**Hakan Türkmenoğlu**		

Criteria	TA/Grader	Instructor
Presentation		
Overall		

Till The Last Drop

Cracked Office

Ahmet Reşat Demir

Cahit Ediz Civan

Efe Kaan Fidancı

Ghulam Ahmed

Mustafa Kağan Özsoy

Detailed Design Stage

(First Draft)

24 April 2022

1. Introduction

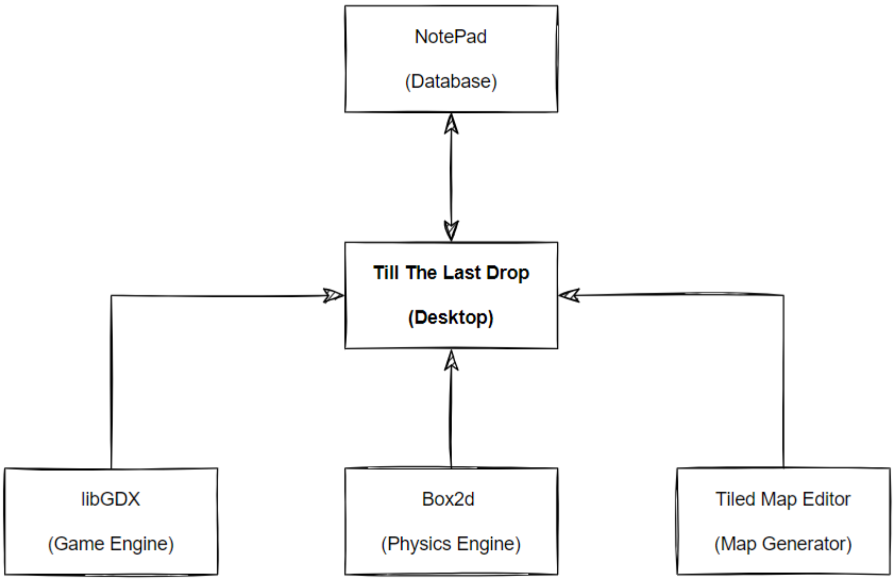
Our game will be a tower defense strategy game with the aim to prevent enemies from damaging our central core. We will have multiple towers with different attacking features allowing users to protect the center from the enemies. Players will try to keep enemy characters out of their territory as much as possible.

2. Details

2.1 System Overview

Our game is planned to be a desktop game. The technologies we use are a game engine, physics engine, map editor and a simple database. The game engine we use is libGDX which is available for Eclipse. For the physics mechanisms of the game, we use box2d. We use a tiled map editor for designing the map of levels. Since the progress will just be a game level progress, for example level 2 will not be played before level 1 is accomplished, a notepad for keeping this information is enough.

Our game is planned to be a desktop game. The technologies we are planning to use are the physics engine found within LibGDX, tilemap editor called Tiled and a simple database. For the physics mechanisms of the game, we use the box2d library of LibGDX. We created the map using the tilemap editor. Only the progress of the player will be saved in our game. Therefore,(for example level 2 will not be played before level 1 is completed) a text file for keeping this information is enough.



2.2 Core Design Details

1. GameObjects Package
    a. Base : The class of the base we are responsible to defend
    b. Bullet : The class of bullet that deals damage
    c. Creator : The class responsible for creating objects
    d. DoomTower: A basic sample subclass of tower class
    e. Entity: An overarching class for npcs and towers.
    f. Melee: A basic sample subclass of npc class for melee npcs

g. Npc: The class of npcs that is responsible to attack to base

h. Player: A class responsible for interactions of player

i. Powerup: The class responsible for powerups

j. Projectile: The projectile class that covers the types of objects that are being shot from towers to npcs or ranged npcs to towers and dealing damage.

k. Ranged: A basic sample subclass of npc class for ranged npcs

l. Shooter: Interface for objects that can shoot

m. Soldier: A basic sample subclass of ranged class

n. Tower: The overarching class for defending towers.

2. Constant: A class for keeping constants for scaling.

3. Scene Package

a. End menu: The class for displaying the end game menu

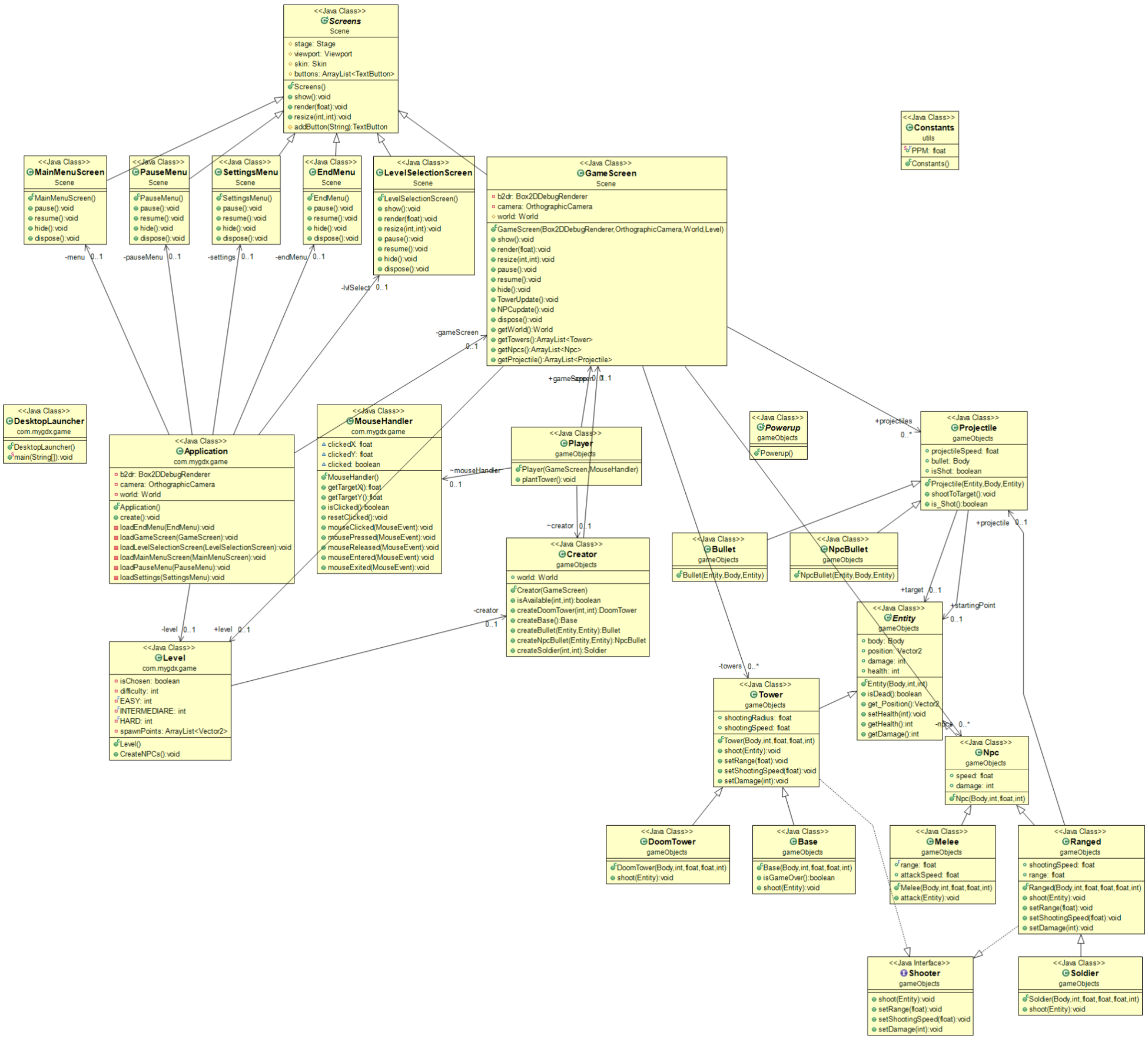
b. GameScreen: The class for displaying the gameplay which has the physical world and arraylists of physical bodies.

c. LevelSelectionScreen: The class that allows user to select levels from available ones

d. MainMenuScreen: The class and screen that greets the user when first entered to application

e. PauseMenu: The class that is responsible for pause screen during gameplay

f. SettingsMenu: The class that allows user to change the settings



2.3 Task Assignment

- Ahmet Reşat Demir: Responsible for user interface, organization of the classes, particularly scene package, and mechanics of the tower and some other classes gameplay mechanics in the GameObjects package.

- Cahit Ediz Civan: Responsible for the physical reactions of game objects, player interactions with the game screen and object design of NPCs.

- Efe Kaan Fidancı: Responsible for the animations, art design, in-game sounds, screens, github repository management, interactions between the screens, action event listeners, power ups and the game loop. Will mostly work with the classes in the gameObjects and the scene package.

- Ghulam Ahmed: Responsible for database and object oriented design.

- Mustafa Kağan Özsoy: Responsible for level design and user interface, mechanics of the projectiles.

All of us will work on the following parts of the project:

- Gameplay
- Application

3. Summary & Conclusions

Our software (as per our current schedule) will have four packages : gameObjects package, scene package, utils package and com package. Game objects package will contain the objects that will be implemented in the game, scene package will be where the core events in the game loop will take place (such as menu, game screen and level completion menu), utils package will contain the helper methods and instance variables to scale the screen, animations, etc., and finally the com package will contain the main method (the game). (Here is a link to our UML diagram since the resolution here come out a little distorted 🗎 ttld_-_UML_Diagram.png) The task division will be done according to the information given above.