# Space Invaders

## Documentation

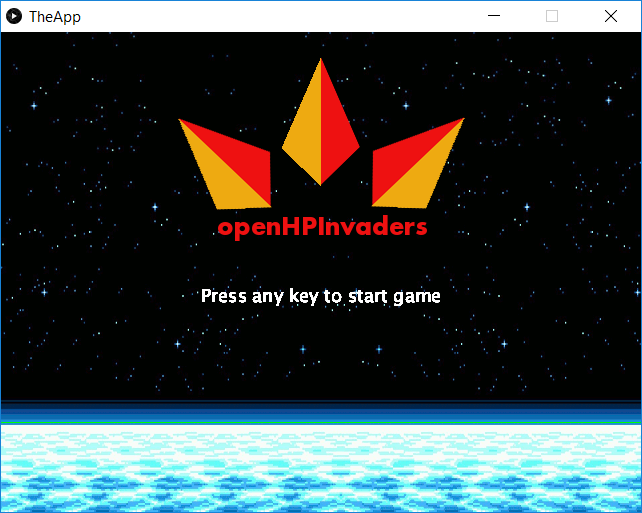
### Introduction

We created a game called “Space Invaders.” It is an arcade game created in Japan and first released in 1978. It is a two-dimensional shooter game in which the player controls laser cannon by moving it horizontally across the bottom of the screen and firing at descending aliens.

### UML Diagram

### Game start

The game is started by pressing any key, on the intro screen:



### Gameplay

The game controls are as follows:

**Key “a” move left**

**Key “d” move right**

**Key “s” fire**



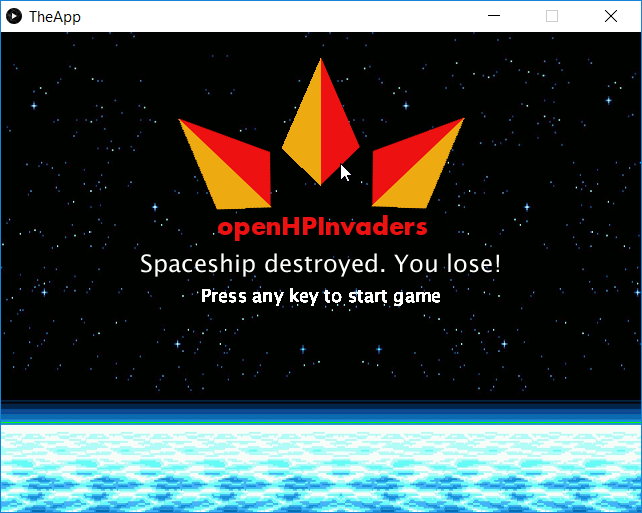
At the beginning, there are 6x4 = **24 aliens created**. The aliens are descending and firing the bombs randomly. An alien can’t fire a bomb, if there is another alien beneath.

The spaceship can fire laser bullets; there can be a **maximum of 3 active / visible laser bullet** on the screen at a time. An alien hit by a bullet is destroyed and removed from the screen.

The same applies to the spaceship, if hit by a bomb, it is destroyed.

The **spaceship / player has 3 lives**, displayed as spaceship picture in the right bottom of the screen.

End of the game:

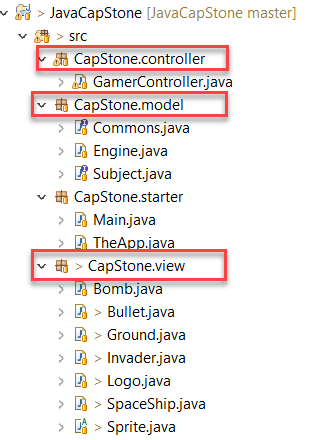


### Code and game logic

The application was created based on the **model-view controller principle**. Four Java packages were created, one for model, then for view and the controller, each containing several Java classes.

The fourth package was created for the Main and TheApp class.

As you can see in the UML diagram, all the visible game objects were created (inherited from) based on the Sprite class.



Classes **Engine and GamerController** represent the game logic and the game control mechanisms.

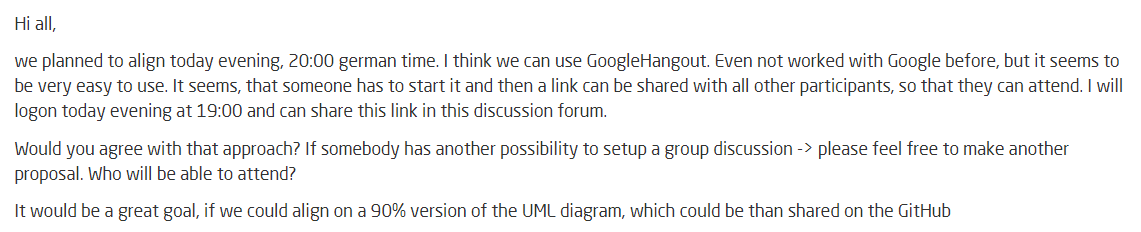
The class **Commons** contains all the important global variables for the application.

## Lab Report

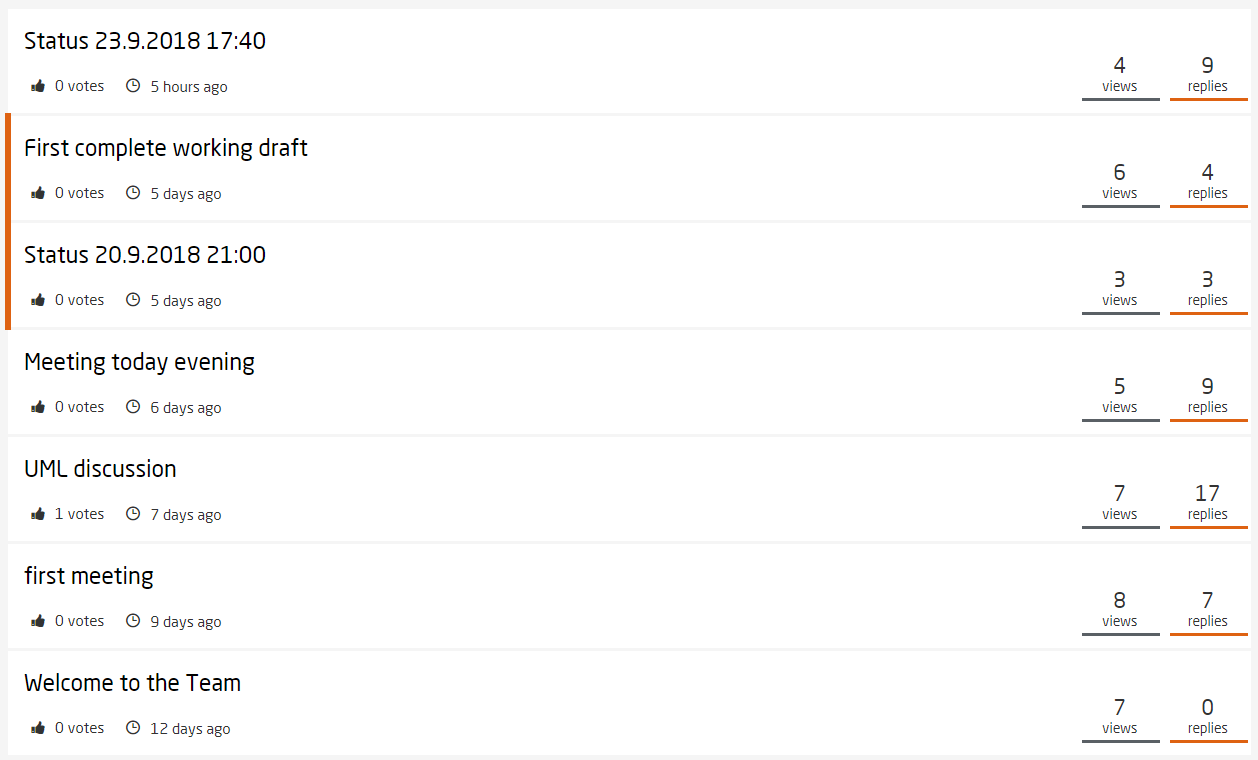
The project started by knowing each other in the team. The team consists of **8 people**, unfortunately **only the few were actively participating in the project**.

We managed to have one telco, in which 4 members of the team participated. Besides the telco, the communication was quite active.

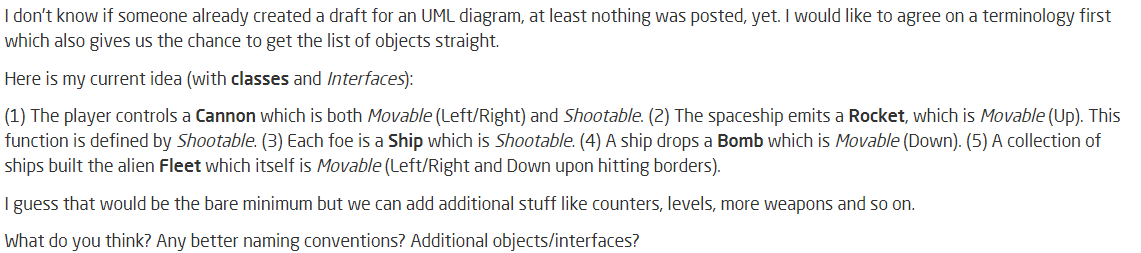
**Telco arrangement:**



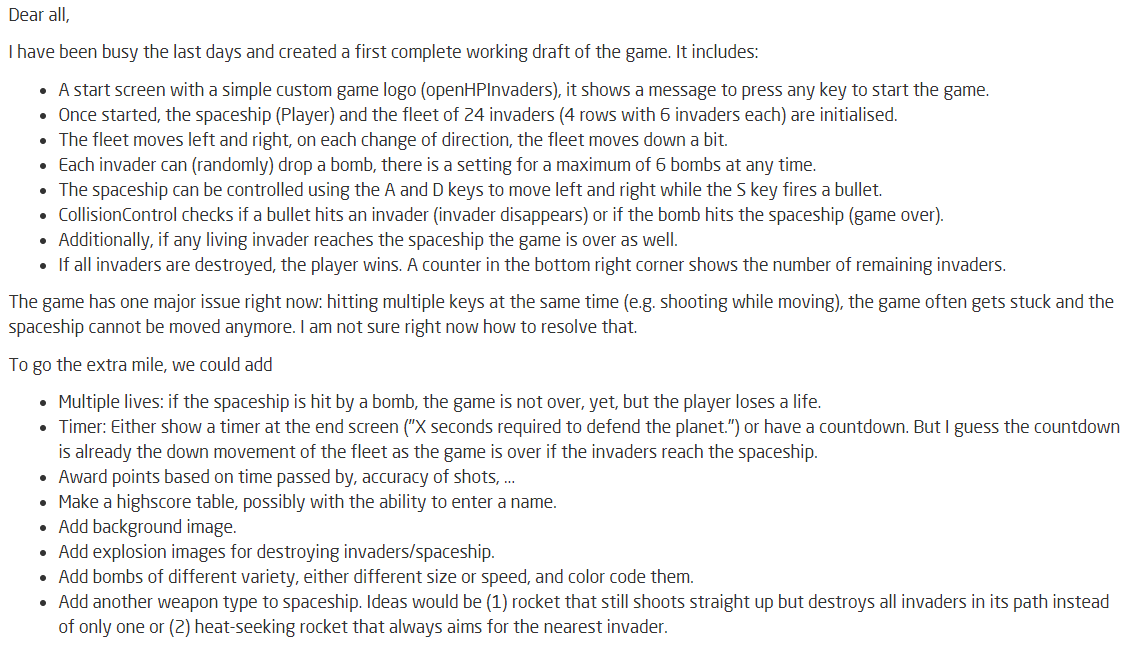
**Discussion topics:**



**UML discussion:**



**First working draft discussion:**



The working environment was set up first - the GitHub account. Then the JavaCapstone project from the course was downloaded as a basis for the game.

We started with the simple UML diagram, where the basic game Java classes were drafted.

Then the progress of the implementation itself was very smooth, due to one member of the team, who implemented most of the game logic and the code.

The whole project from the beginning to the end was finished pretty much in 12 days, but working **actively maybe 2-4 days**.