

Assignment 2

Asteroids

*Created by group OP27-G59 for the course
CSE2115 Software Engineering Methods
of the Computer Science curriculum
at the Delft University of Technology*

Group members:

Onur Gökmen
Joseph Catlett
Irtaza Hashmi
Helena Westermann
Ceren Uğurlu

CONTENT

Exercise 1: UML Class Diagrams

1.1 Class Diagram Description	3
1.1 Class Diagram	4

Exercise 2: UML Sequence Diagrams

2.1 End of Game.....	5
2.2 Asteroid - Missile Collision.....	6
2.3 Register New User.....	7

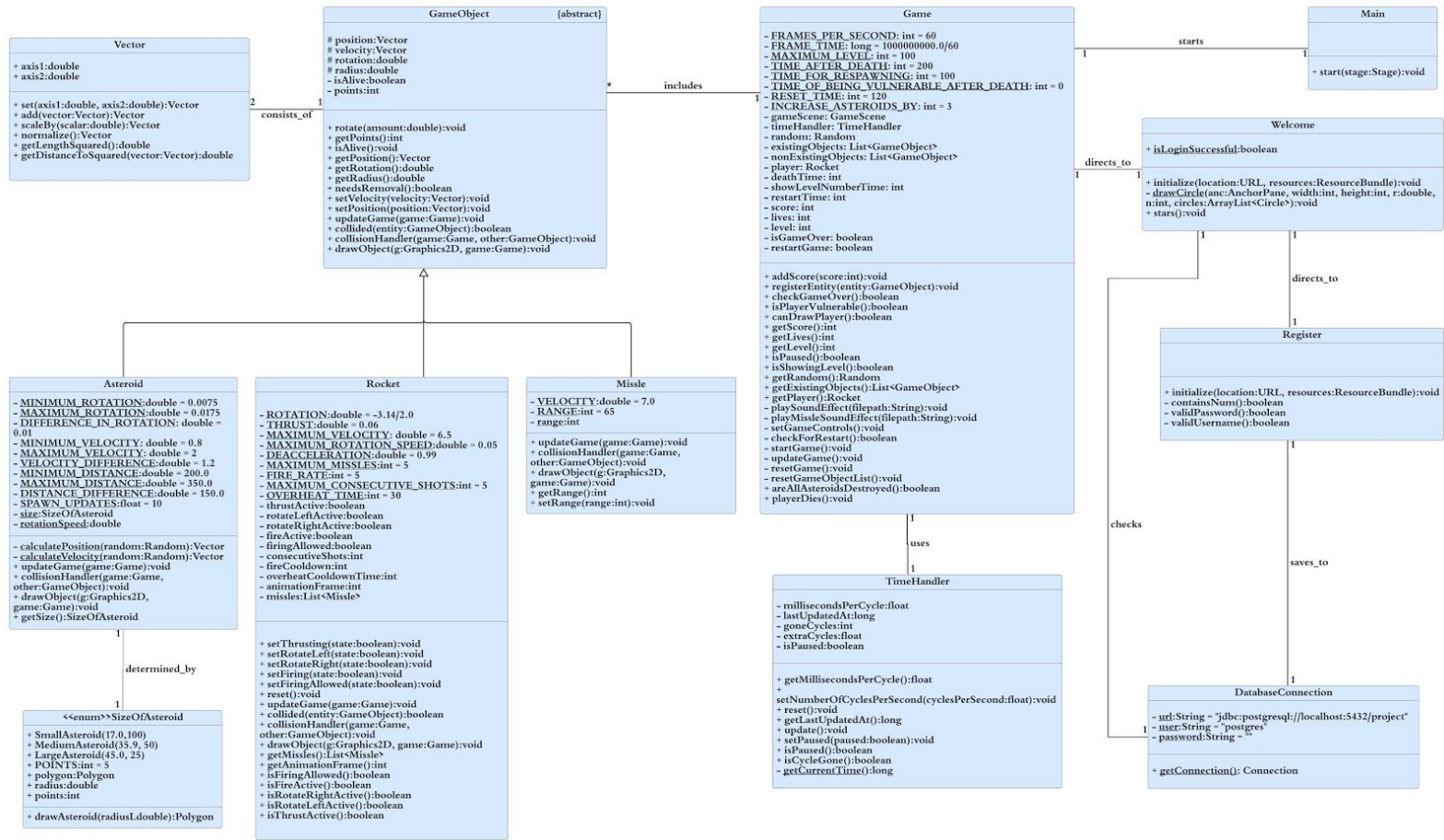
Modelling Class Diagrams

Description:

As one can depict, we have our class diagram on the next page. For the game logic, we have an abstract `GameObject` class that defines the game objects in the game. This includes asteroids, the rocketship and the missiles. The `Asteroid`, `Missile` and `Rocket` class inherit from the `GameObject` class. There is also an enum called `SizeOfAsteroid` that is related to the `Asteroid` class. This enum keeps of the three different types of asteroids possible in the game (small, medium and large). In order to keep track of the game objects in the game, we use a `Vector` class. The `Vector` class makes our job so much easier to locate different objects in 2D space as well as change their positions as desired. Finally, we have our `Game` class that is extended by a `JFrame` to launch the game. We create game objects in the `Game` class and add or remove them according to the player's actions. The `TimeHandler` class helps us keep track of time and the number of cycles passed that helps us update the game and repaint it. It also allows us to pause the game.

On the other side, we have the user interface (UI) that is launched before the game. This UI handles logging in and authentication as well as registering for the new users. We have a `DatabaseConnection` class that connects with the database and gets all the necessary data about the user (e.g. checks whether the user already exists or getting the user's high scores). We have a `Register` class, that helps us to register a new user as well as a `Welcome` class which is a `JavaFX` class in order to launch the welcome page. Finally, we have the `Main` class that launches the whole application (Welcome page as well as the game itself).

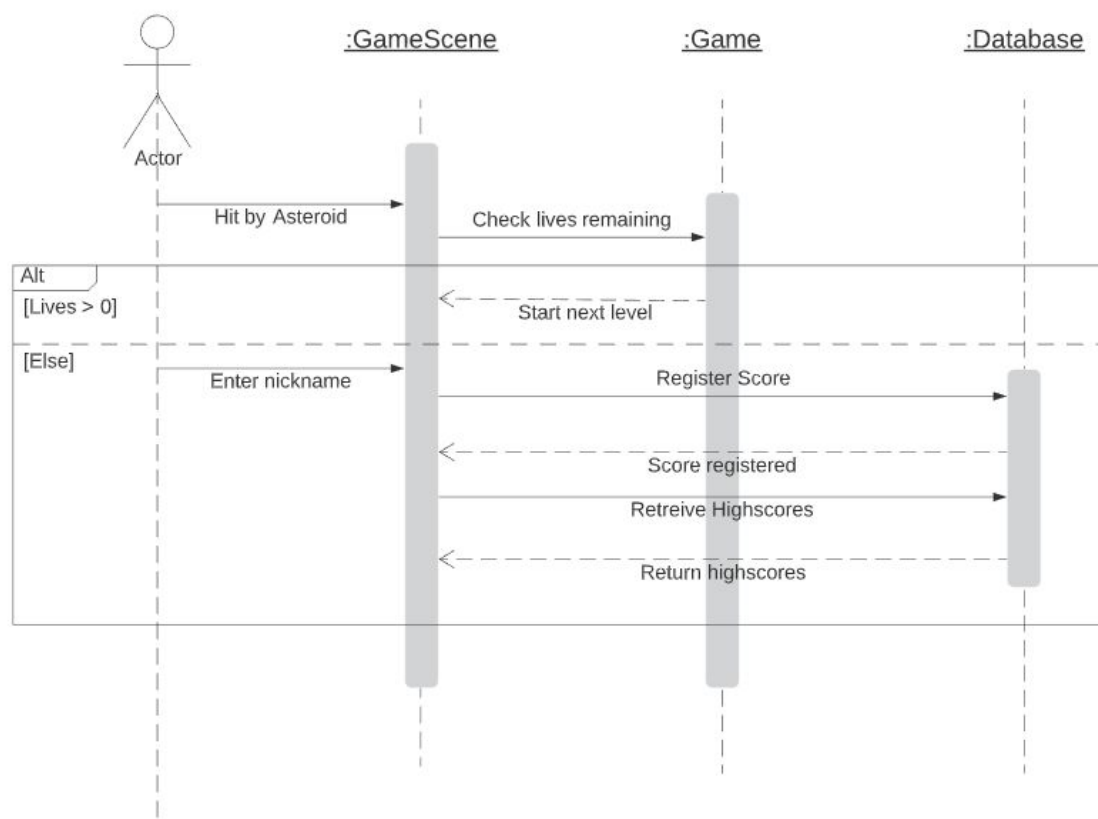
CLASS DIAGRAM



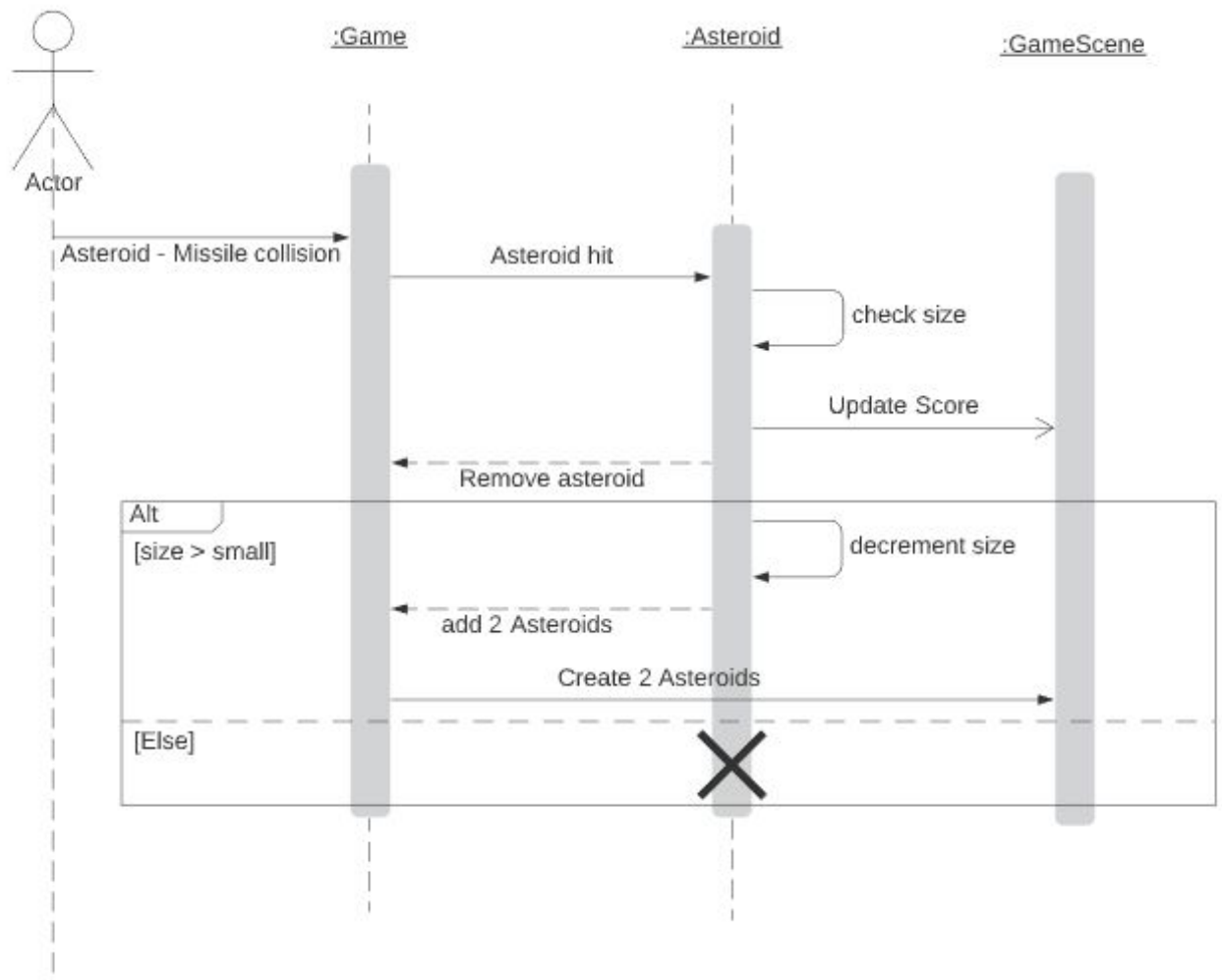
2

Modelling Sequence Diagrams

UML Diagram 1: End of Game



UML Diagram 2: Asteroid - Missile collision



UML Diagram 3: Register New User

