Game: Asteroids

Game story:
The player is a trainee pilot who is piloting an aircraft. The trainee pilot is being trained to destroy asteroids.
To pass the training and become qualified as a pilot, the trainee must destroy 110 asteroids in 3 minutes, whilst not letting more than 50 asteroids fall on the surface of the earth.

Implementation:
it starts in the main method. In the main method an instance of the game class is created and its show method is called.
The game class is the starting point of the application.
All the GUI elements are initialised here and added to the scene.

Typical implementation of the player
The player has a player class which stores the coordinate and path of the player, and provide an interface to this data.
There is then the PlayerRenderer which renders and controls the player, using the data defined in the player class. The playerrenderer renders the player inside the constructor of the game class.

Typical implementation of the asteroid
The asteroid has an asteroid which stores the basic data of an asteroid. In addition it provides an interface to this data. Also, the asteroid class implements a periodically executed function which ensures that asteroid coordinate are modified periodically, create thus an asteroid that moves on the screen.
Then there is the asteroid renderer which creates the image version of the asteroid using the data provided by the asteroid class. This renderer also implements a periodic function which ensures that the latest coordinate of the asteroid are used to render it.
Then there is the asteroid factory which uses the asteroid class and the asteroid renderer to create and render asteroid on the screen.
The factory also stores the created asteroids in data structure and then makes then available for collision detection.

That’s the concept of the game in a nutshell. The rest can be understood by looking at the code.

Future of the game:
- implementation of agents to separate logic and UI layers
- lives for the player and a restart button