CS102	**Spring 2017/18**	Project Group	5F
Instructor:	**Aynur Dayanık**		
Assistant:	**Mustafa Can Çavdar**		

~ Kepler's Journey ~

Gravity Guys

Eray Ünsal Atay & Berk Bozkurt & Hakan Sivük & Yusuf Nevzat Şengün & Mehmet Tolga Tomris & Muhammed Yusuf Toy

Criteria	TA/Grader	Instructor
Presentation		
Overall		

Detailed Design Report

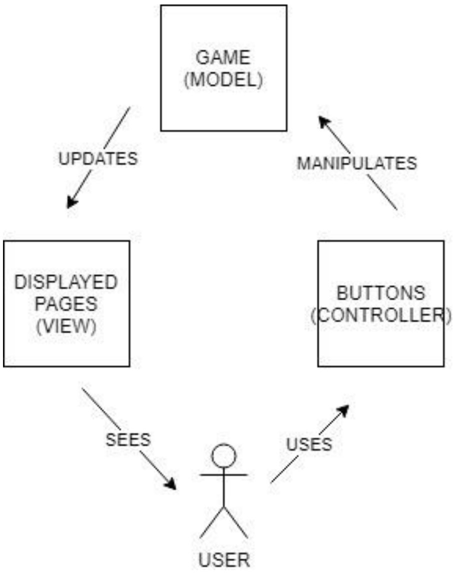
(version 1.0)

20 April 2018

1. Introduction

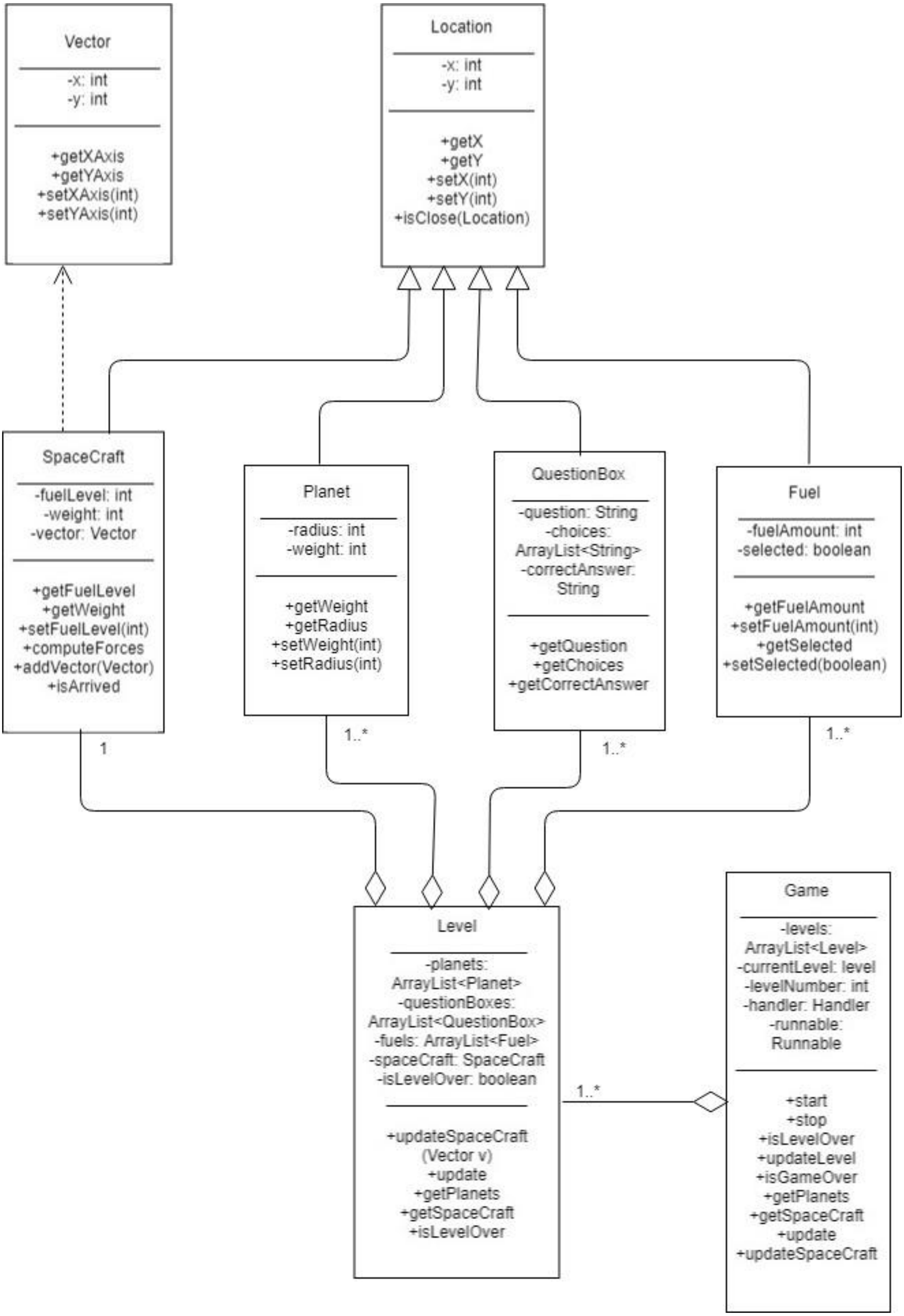
Our application is about an astronaut leaving Earth in search of a planet that is suitable for life. It aims to simplify practical learning process of Kepler's Laws and entertain the learners at the same time. We are targeting the students who are interested in physics and willing to learn gravitational laws.

2. System Overview

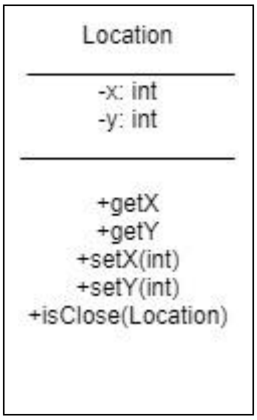


The application is simply based on MVC pattern that is shown on the left. The model is the logical structure of the game. It doesn't contain any information about the user interface. The view represents the user interface of the application such as game play panel. The controller is received the user interactions and according to them manipulates the game. Then, the game updates the view and as a result of the interaction, view is updated. The application will be used on mobile devices and firstly we will develop the application in Android. In order to keep data such as high scores and loaded games, we will use Firebase system.

3. Core Design Details

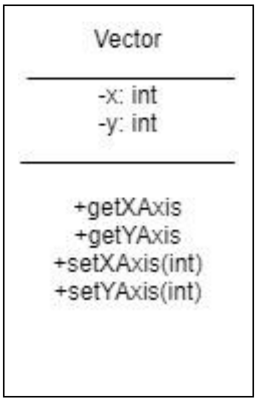


3.1. Location:



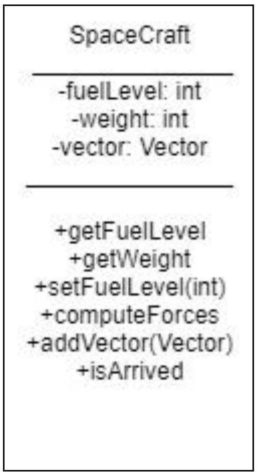
It is a class to set the x-y locations of the planets, fuel boxes, question boxes and spacecraft in our application. It takes two parameters in its constructor in order to set the default values of the x-y coordinates and these can be changed through mutator methods. It also contains accessor methods to get location information. This class is extended by Planet, SpaceCraft, QuestionBox and Fuel classes.

3.2. Vector:



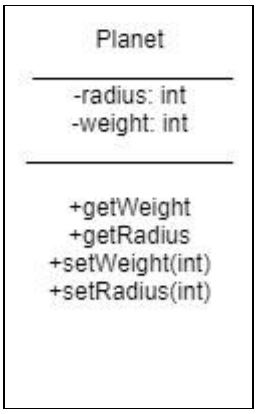
It is a class to represent the vectors. It has two properties which are x and y. x represents vector magnitude on x-axis and y represents vector magnitude on y-axis. getXAxis and getYAxis are accessor methods, setXAxis and setYAxis are mutator methods of x and y.

3.3. SpaceCraft:



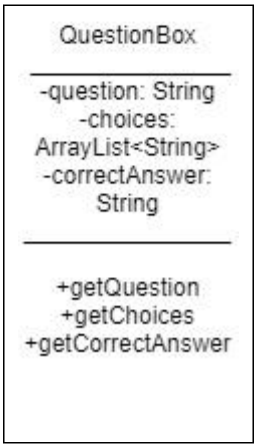
This class represents a space-craft. fuelLevel keeps its current fuel level. weigh keeps its weigh. vector keeps its current direction. There are accessor and mutator mehods for FuelLevel and there is accessor method for weight. computeForces calculates the resultant vector that effects spacecraft curently. addVector updates vector by taking a new vector and adding it on vector. isArrived returns if we are arrived.

3.4. Planet:



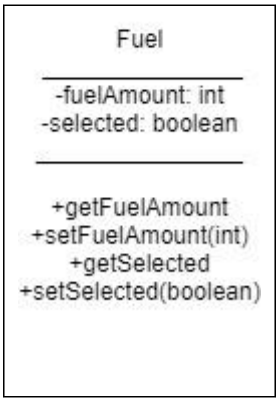
It is a class to create the planets, extending Location. It has two properties which are radius and weight. It takes four parameters in its constructor to set the default values of the radius and weight of the planet. It calls the constructor of the Location class to set the x-y coordinates of the planet. Mutator and accessor methods are also exist.

3.5. QuestionBox:



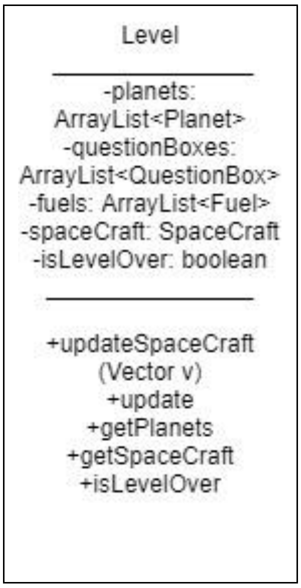
It is a class to create the question boxes, extending Location. It has properties which are: question, choices, correctAnswer. It sets all of them to its default values in its constructor and calls the constructor of the Location class to determine the x-y coordinates of the question box. It also contains accessor and mutator methods.

3.6. Fuel:



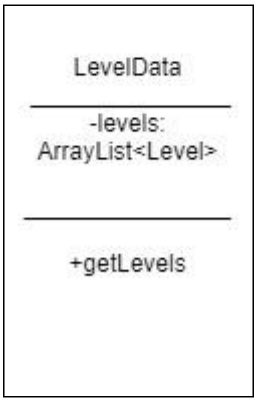
This class represents fuel that are located on space. fuelAmount tells the amount of fuel in it. selected denotes whether this fuel is taken. There are accessor and mutator methods for them.

3.7. Level:



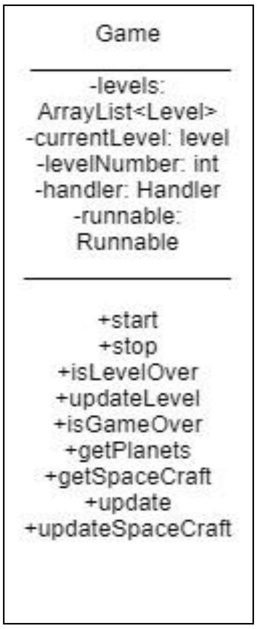
This class creates a level by taking Planet, QuestionBox, Fuel, SpaceCraft objects as its instances. Additionally, isLevelOver denotes whether level is completed. updateSpaceCraft method update the spacecraft’s fuel level and movement vector with a Vector parameter. update method change locations of the planets, question boxes, and fuels. getPlanets, getSpaceCraft, and isLevelOver are accessor methods of the class.

3.8. LevelData:



This class is simply keeps the level plans of the game. In the constructor levels ArrayList<Level> is created by the developers according to the lay-out (location of planets, question boxes, fuels) we determine. getLevels is the accessor method of the class.

3.9. Game:



This class can be seen as a collection of levels. currentLevel is the level that is played currently. levelNumber is the number of the level that is played currently. handler and runnable are for updating the currentLevel every 10 seconds. start method starts the timer and stop method stops it. update calls update method of the currentLevel and updates the current-Level. updateLevel updates the levelNumber and currentLevel if current level is completed. updateSpaceCraft for updating the space craft’s vector and fuel amount.

4. Task Assignment

1. Eray Ünsal Atay: QuestionBox class and some parts of the Game class
2. Berk Bozkurt: Vector class and some parts of the Game class
3. Hakan Sivük: Planet class and some parts of the Game class
4. Yusuf Nevzat Şengün: SpaceCraft class and some parts of the Level class
5. Mehmet Tolga Tomris: Location class and some parts of the Level class
6. Muhammed Yusuf Toy: Fuel and LevelData classes