

[11# Plans for GUI]

[5] – [SoftProLab Team]

Supervisor:

Dr. Katalin Balla

Members:

Hagverdiyev Subhan	NHL9KN	subhan.hakverdiyev@gmail.com
Madatov Ali	XVEARA	ali.madatov@hotmail.com
Shoaib Areeba Tabassum	EZXFWK	areebashoaib99@gmail.com
Salahov Kamal	IG5LSM	salahovkamal@hotmail.com
Singh Gurdeep	ERXIPV	sandhugoldy433@gmail.com

27.04.2021

11. User interface specification

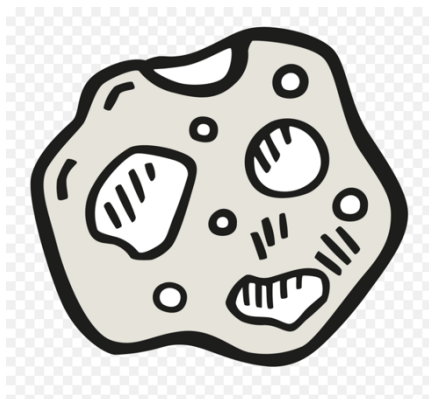
11.1 Graphical User Interface

Our GUI will be built using Swing framework of Java, so most gui components will be generic, but there will be a few elements to make game for appealing for eyes.

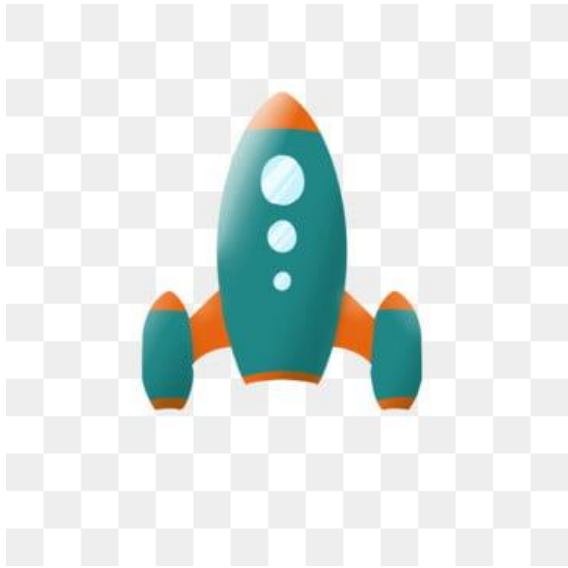
11.1.1 Main Menu



11.1.2 Asteroid



11.1.3 Settler



11.1.4 Robot



11.1.5 Teleportation Gates



11.1.6 Station



11.2 Architecture of the graphical system

11.2.1 Principles of the GUI

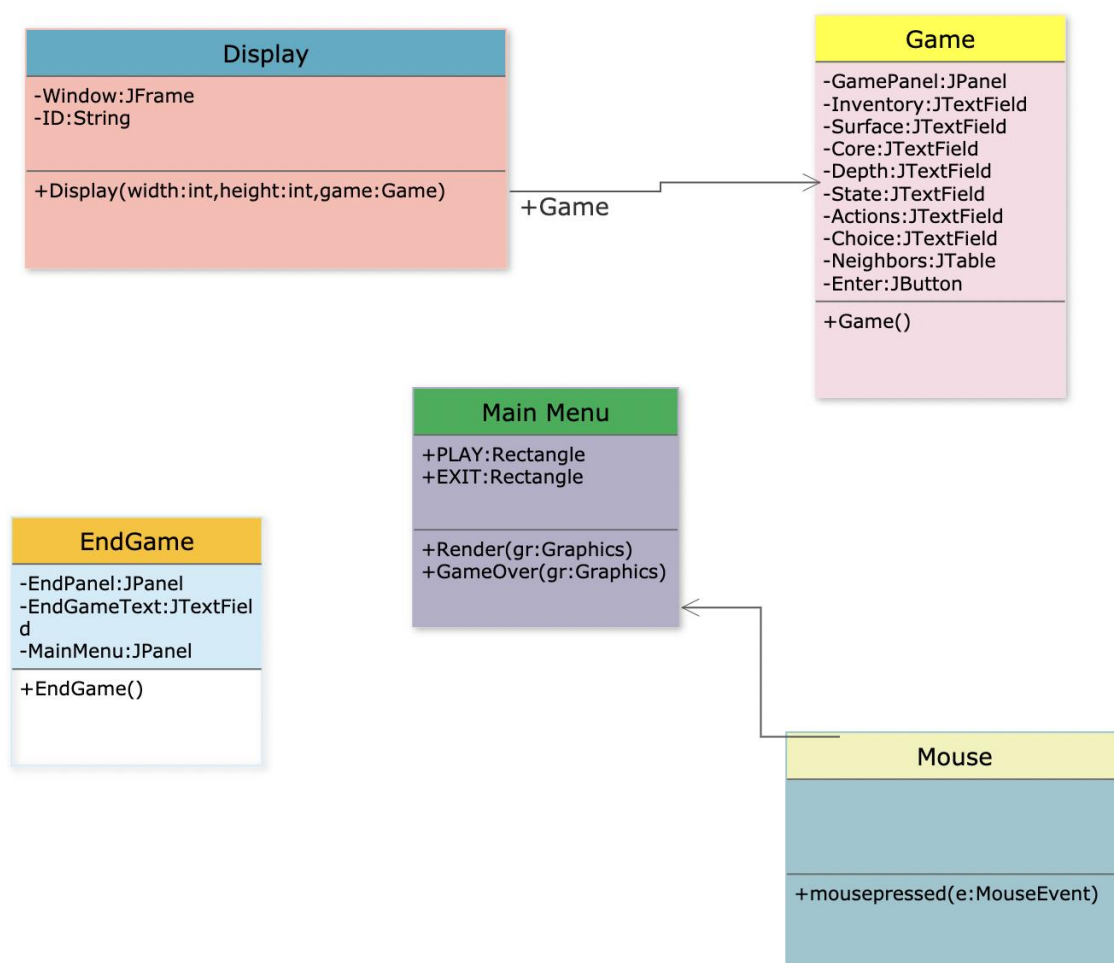
The principle of display is pull based. When the class performing the Controller function (Main) modified the model, calls the Game () method of the Game class.

As a result, the Game class calls the model, more specifically waiting for each element necessary to be drawn in order the main game window to be ready.

It then listens mousepressed() method of Mouse class to change game state.

Finally, he asks Main Menu if the game is over and modifies it accordingly even the drawn state.

11.2.2 GUI Structure diagram



11.3 GUI Classes

11.3.1 Display

- **Responsibility**

This class represent and act as container for our game. It sets the frame of the game using JFrame. It's the display where we draw graphical elements for our main menu and the game.

- **Superclasses**

None

- **Interfaces**

None

- **Attributes**

- Window: Private JFrame variable for setting up our window.
- **ID**: Private String which contains the name of our game

- **Methods**

- **Display (int Width, int Height, Game game)**: Setting up the JFrame in the constructor of our display container. Adding the game into the window and deciding the size of window using width and height variable. Setting up the location of window on screen and making it visible.

11.3.2 MainMenu

- **Responsibility**

Responsible for the main HUD of the game. This is the first screen user will see when they will run the application. It contains Play and Exit button.

- **Superclasses**

None

...

- **Interfaces**

None

- **Attributes**

- **PLAY**: It's our rectangle box with two shades inside which the letters (Play) has been written.
Type: Rectangle
Visibility: Public
- **EXIT**: It's our rectangle box with two shades inside which the letters (Exit) has been written.
Type: Rectangle
- Visibility: Public

- **Methods**

- **void Render (Graphics gr)**: Using the inbuilt Graphics2D and Graphics class. We fill rectangles of certain width and size at the certain location in our window. We used

drawString function of graphics class to write the Play and Exit inside these filled rectangle.

- **void GameOver(Graphics gr):** In this function we draw the string Game over using graphics class in the display as soon as our settler dies. This function will execute and draw the GameOver at certain place in screen.

11.3.3 Mouse

- **Responsibility**

This class is responsible for taking mouse input from the user. It's heavily connected with our main menu hud as certain function will execute depending upon which button is clicked.

- **Superclasses**

MouseListener

...

- **Interfaces**

None

- **Attributes**

None

- **Methods**

Void mousepressed (MouseEvent e): This method is responsible for the game getting executed and user exiting from game by pressing mouse button on some button. By pressing certain button it results in certain enumerator being assigned to our game and changing its state to said enumerator.

Pseudocode:

Int a equals to MouseEvent e's X position

Int b equals to MouseEvent e's Y position

IF a is Less Than Certain Game WIDTH AND a is Greater Than Certain GameWIDTH

**IF b is Less Than Certain Integer value AND b is greater Than Certain Integer Value
SET GAME's IDENTITY TO START**

ENDIF

ENDIF

IF a is Less Than Certain Game WIDTH AND a is Greater Than Certain GameWIDTH

**IF b is Less Than Certain Integer value AND b is greater Than Certain Integer Value
EXECUTE SYSTEM'S EXIT.**

ENDIF

ENDIF

11.3.4 Game

- **Responsibility**

This game class is responsible for containing main elements of game, such as buttons for available actions, inventory etc.

- **Superclasses**

None

- **Interfaces**

None

- **Attributes**

- **GamePanel**: Private JPanel that is responsible for holding other elements
- **Inventory** : JTextField object that shows the names of the items in the inventory
- **Surface**: JTextField object that shows all the minerals stored on the asteroid
- **Core** : JTextField object that shows the mineral (if there is any) core in the asteroid
- **Depth**: JTextField object that shows the depth of asteroid
- **State**: JTextField object that shows the state of asteroid
- **Neighbors**: Jtable which lists neighbors of asteroid and their info
- **Actions**: JTextField objects where available actions are listed
- **Choice**: editable JTextField object, where you can write the name of action you want to perform.
- **Enter**: JButton which will perform action written in Choice field upon pressing.

- **Methods**

- **Game ()**: constructor which initializes objects, sets their dimensions and their position of the window. Then adds them to frame and makes them visible.

11.3.5 EndGame

- **Responsibility**

This class is responsible for representing end game screen of game.

- **Superclasses**

None

- **Interfaces**

None

- **Attributes**

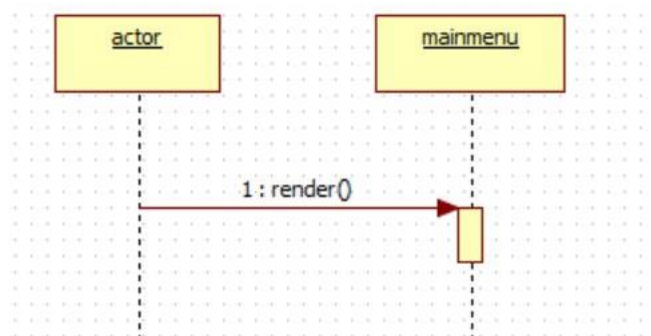
- **EndPanel**: Private JPanel that is responsible for holding other element.
- **EndGameText**: JTextField which contains endgame message for user.
- **MainMenu**: JButton which sends user to the main menu of the game upon being pressed. „Main Menu” is written on it.

- **Methods**

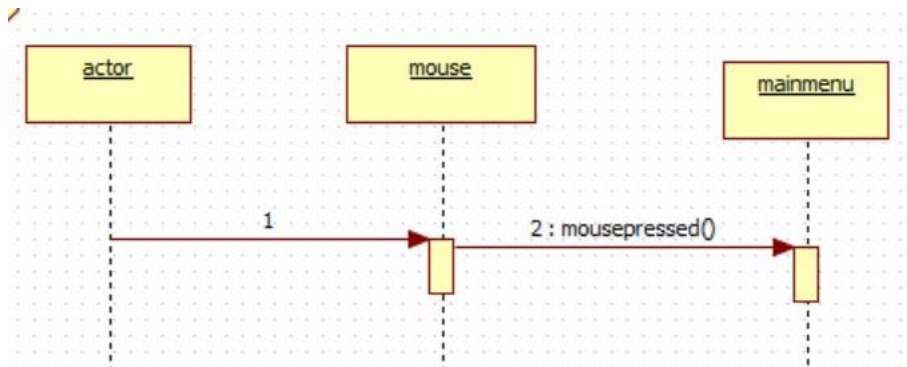
- **EndGame ()**: constructor which initializes objects, sets their dimensions, adds them to frame and makes them visible.

11.4 Dynamic connection between the model and the GUI

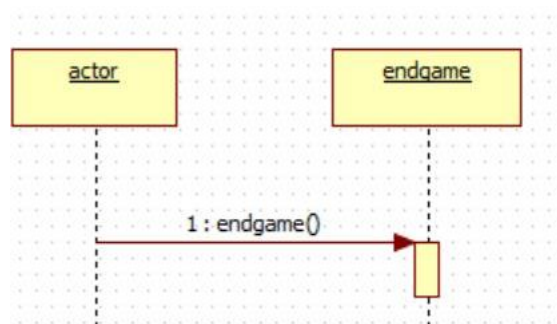
11.4.1



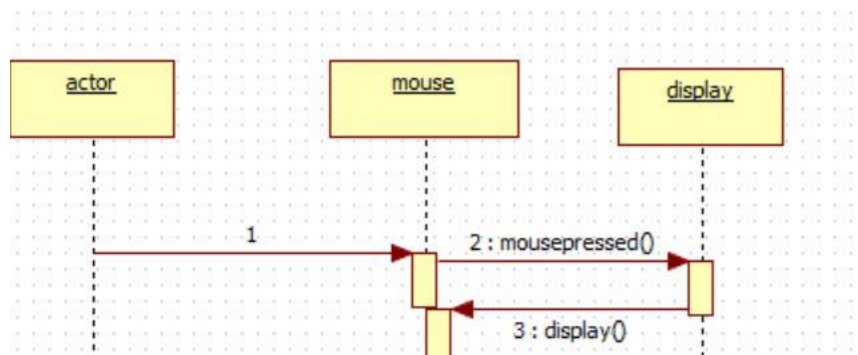
11.4.2



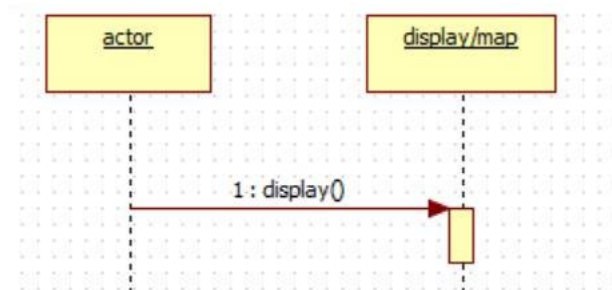
11.4.3



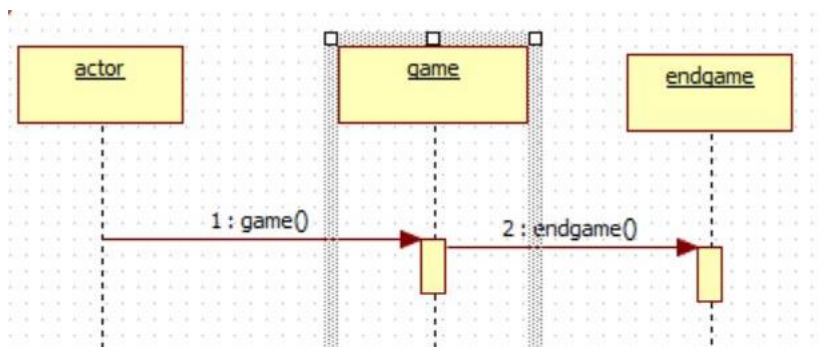
11.4.4



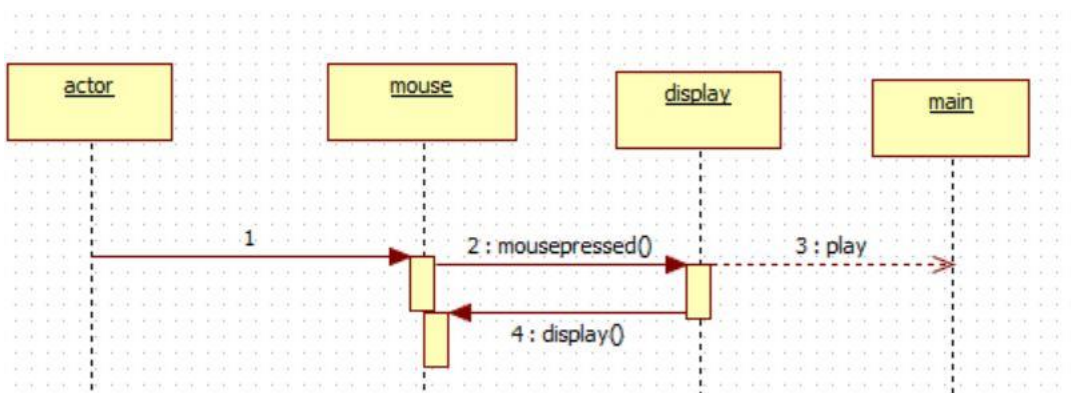
11.4.5



11.4.6



11.4.7



11.5 Protocol

Start (date & time)	Duration (hours)	Performer(s) name	Activity description
24.04.2021	30 minutes	Whole Team	Dividing tasks
25.04.2021	1.5 hours	Gurdeep Singh	11.3.1, 11.3.2, 11.3.3
26.04.2021	1.5 hours	Salahov Kamal	11.1, 11.3.4, 11.3.5
26.04.2021	1.5hours	Areeba T.Shoaib	11.4
26.04.2021	1.5 hours	Ali Madatov	11.2.1
26.04.2021	1.5 hours	Subhan	11.2.2