****

**CS 319 - Object-Oriented Software Engineering**

**Final Report**

***Hurdle Race***

Group 4

Ecem Afacan 2

Haluk İncidelen

Selim Mıdıkoğlu

*Course Instructor: Uğur Doğrusöz*

**1. The Implementation Process**

As a first step we implemented our frame display components, like

runners, run ways, hurdles, bacground pictures. As a second step we

implemented our classes for game frame whichdisplays game bacground, runner,

hurdles and all components in action. So for this display we implemented our

hurdle class, runner class, GameScreenPanel, GameManager class and more. Than

we put Key Event for player to control runner’s jumps. Later on, we implemented

our MainMenuPanel class. We declared our buttons(Start Game, Options, Exit).

Than we made some changes inside of the classes. For example we changed the

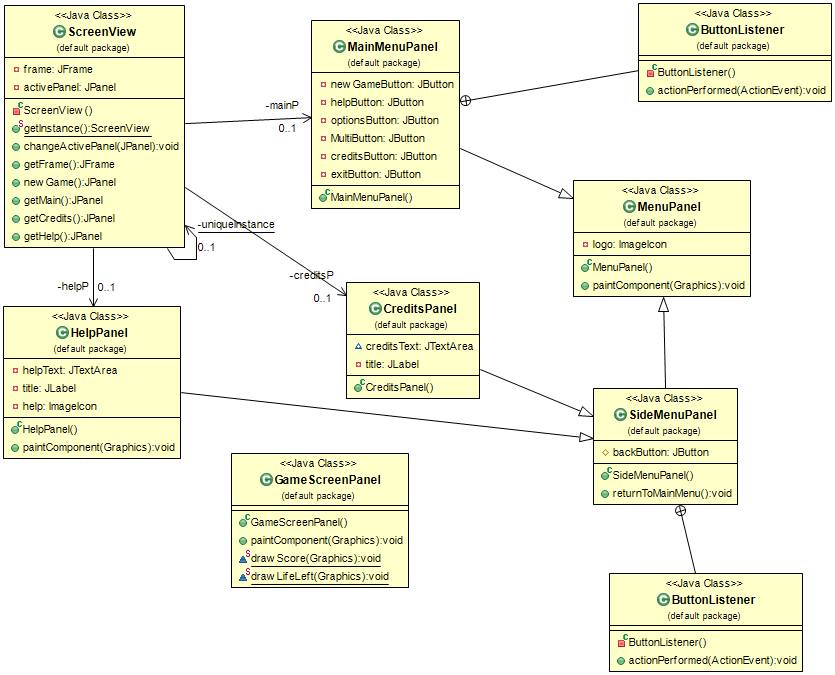
jump method. Since we increase y axis of our buffer image with specific number

like 1. We forget to add jump limit on y axis. We resolved this problem with

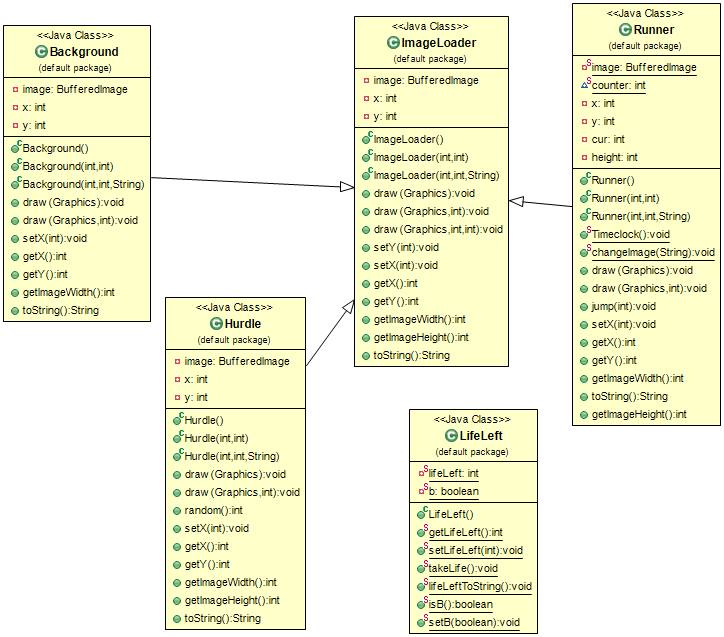
putting a maximum limit of height which runner can be able to pass onto hurdle.

As last part we runned our game too many times to see if there is any problem or

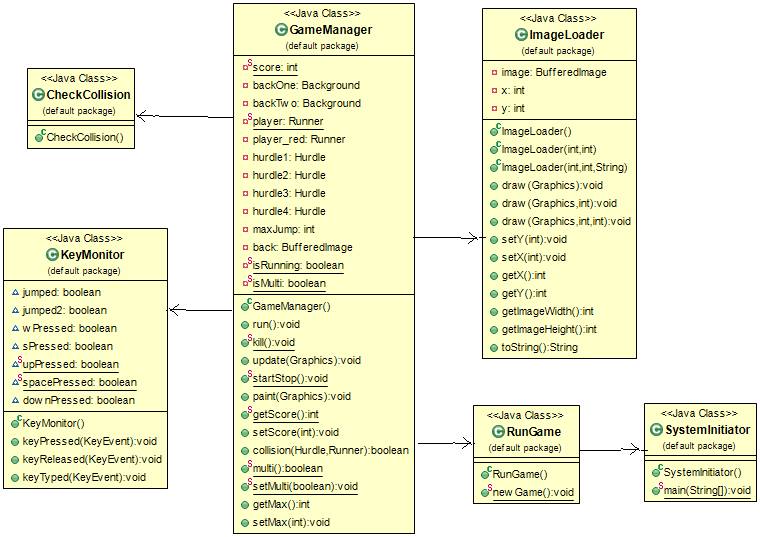
not.



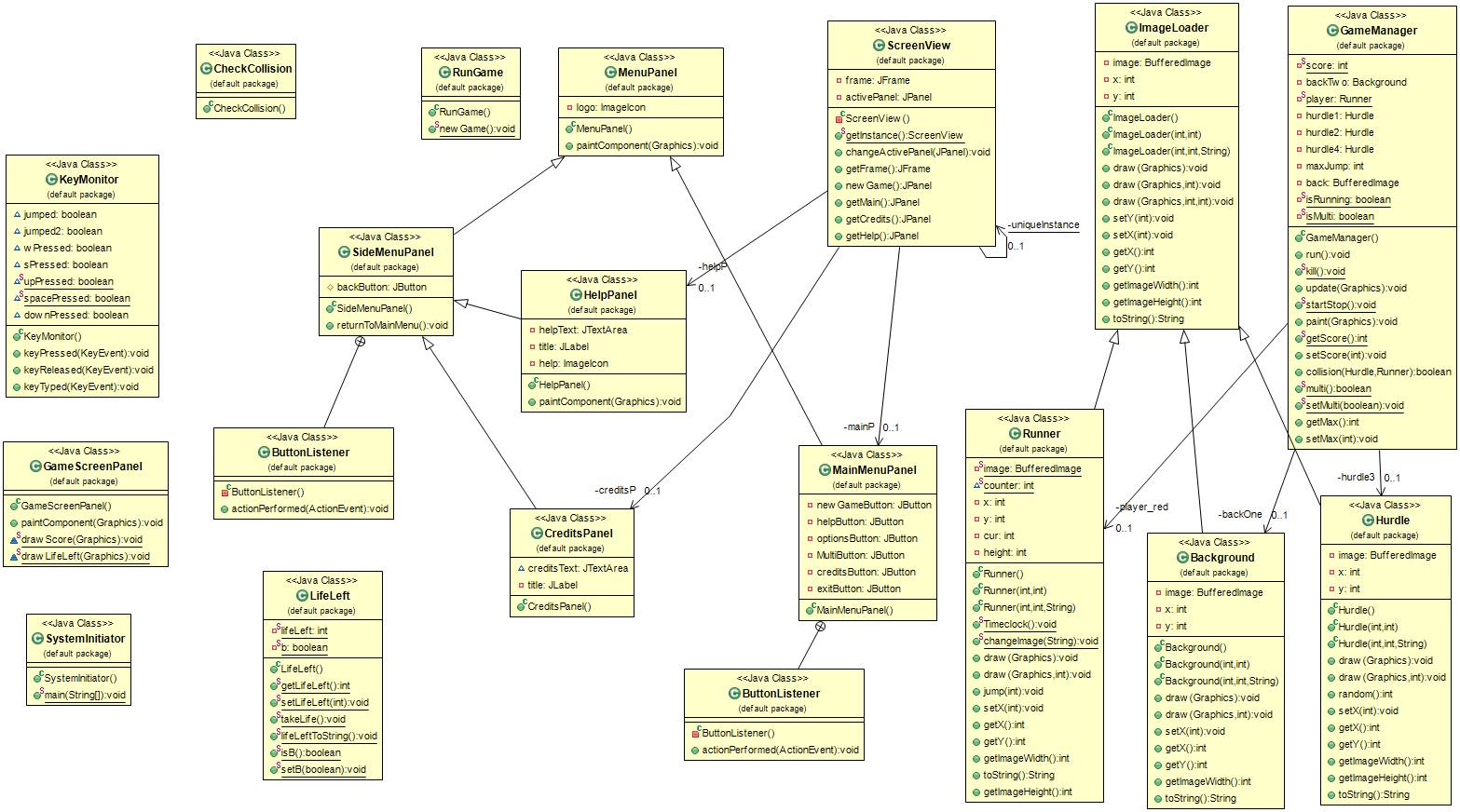
**Figure 1:** User Interface Subsystem UML Diagram



**Figure 2:** Model SubSytem UML diagram

****

**Figure 3:** Controller Subsystem UML diagram



**Figure 4:** Overall System UML Diagram

**2. The Things Are Not Implemented**

In our game there are two unimplemented part which is the game music and bonus boost. We did not add any music. That is not because we could not implemented it, it is because we were kind of missing the time that we should add.

The bonus booster that we want to implement is a boost for runner to run faster. When runner does not hit specific amount of hurdles than he gets this running fast boost. So we could not implement it because it was a little bit hard. Because at the same time we are counting hurdles, and deciding when to give the booster and moving runner buffer image coordinates more faster. So we did not implement this future we want to add in previous steps of the process.

**3. Users’s Guide**

Hurdle Race is ordinary type of run race games. It is 2D Graphic designed game which runner tries to not hit any of normal or water hurdles and tries to finish the game. The game can be played as single player or multiplayer mode. So there is optional game playing which makes it interactive game as well since there is multiplayer option.

The thing which makes the Hurdle Race different from other run racing games is that, it has more powerful color displays. Additionaly, the game is easy to learn and does not require powerful systems to work in an efficient way. Game provides easy user interface with couple of button and game control buttons. Simplicity and nicely implemented graphical user interface of the game provides joy.

Menu: For starting the game player can click on “Start Game button” with mouse. In Menu bar buttons there is additional “Help” button which gives clues about game and how to play futures. And there is “Exit ” button which stops execution of the game and exit the game.

Motion: To control over runner, player do not have to push some buttons for running. Only thing that player should do is pushing a button for jumping over the hurdles which is “Up ” direction button on keyboard . That is all about to control actions of runner. Additionally for emergency, game provides pause button which is “P” on the keyboard.

Aim of the Game: Aim of the game is to finish the race by the minimum number of hurdle crashes or 0 crash. In multiplayer mode the aim is same as well ,and additionally finishing the race with smaller amount of time.

Setup and Initiation: Every platform that has JRE(Java RunTime Environment) can execute this game. Because game is jar file and there is no need to installation.