BRICK BREAKER GAME

-Nikunj Gupta (imt2014037)

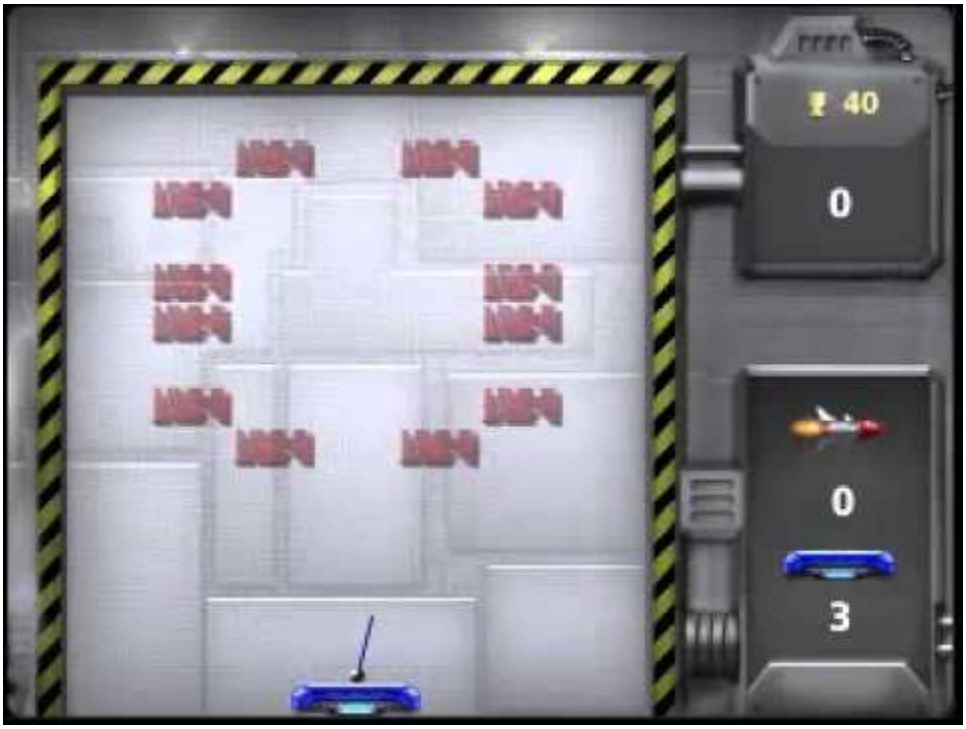
-A.Ravi Teja (imt2014008)

-D.Nithish (imt2014016)

-G.Sai Sarat (imt2014021)

DESCRIPTION:

When the program is initiated, a panel appears with a paddle in the lower part of the panel and a group of bricks present in the upper part of the panel. The user is required to deflect the ball using the paddle towards the bricks, with the left and right keys based on GUI-based application. The user is declared as winner if he manages to break all the bricks else the game is lost.



DESIGN:

For implementing this program, some of the built-in classes need to be imported such as Graphics, BorderLayout, colour, font present in java.awt package and others like ActionEvent, ActionListener, keyEvent, Keylistener in java.awt.event. JButton, JFrame, JPanel in javax.swing

The Graphics class is the abstract base class for all graphics contexts such as color, font, and other graphic applications.

A border layout class lays out a container, arranging and resizing its components to fit in five regions: north, south, east, west, and center. The ActionEvent class which indicates that a component-defined action occurred. This high-level event is generated by a component (such as a `button`) when the component-specific action occurs. The listener interface for receiving action events. KeyEvent is an event which indicates that a keystroke occurred in a component. This low-level event is generated by a component object (such as a text field) when a key is pressed, released, or typed.

In our program, we will write a class which extends and implements some of the above mentioned classes and interfaces.

1) We will write a class which contains the initial co-ordinates of the ball, paddle and the bricks.

2) The number of bricks will be given through a constructor. After knowing the number of bricks, we will write a function which manages the alignment of the bricks in the panel.

3) We will write a function which gives colour to the objects present in the panel like ball, paddle, bricks and background.

4) We will write a function which assigns the x and y coordinate velocities of the ball and will also reverse the y-coordinate velocity after colliding with the brick, we will keep track of the bricks which broke to display the score of the user at the end.

5) We will write a function which when the ball hits the boundary limits, it reflects. We will also write a condition that if the ball goes below the level of the paddle, then the game ends and the score is displayed.

6) We will override some of the functions from KeyEvent which will control the movement of the paddle by pressing the left and right keys.

7) We will also override a function from ActionEvent which will perform a task if a specific button is selected.

8) If the user failed to win, he has the option to restart the game by pressing on the restart button which will in turn call the restart function where the game will be started again.