Mini Project Documentation

Brick Breaker Game

University of Hyderabad

This is a report on Brick Breaker Game the mini project submitted by team named KV

Karan Soren(17MCMC12)

Balakisti Vijay(17MCMC27)
under the Guidance of
Dr P S V S Sai Prasad
Assistant Professor for the
fourth(IV)semester of

Master Of Computer
Applications
(MCA)

Index

Introduction:

Our Sincere Intention is to make an interactive Game based upon the classic game brick breaker.The object of brick breaker game is also to break the bricks which are arround the top of the game screen, when the ball hits the bricks then the bricks will be breaken, we have paddle at botton that moves based on user input, the user make sure that ball bounces off the paddle without getting off the bottom of the screen.In our implementation a camera will track the position of the paddle as the user moves the paddle.

Game Overview:

The Game consists a set of Bricks that are needed to be breaked, the purpose of game is completing the all bricks breaking without losing the ball to the bottom of the screen,the game cannot be saved and loaded when tried to continue from where it is saved after even closed, here when user breaks bricks then user will get 5 points for each brick and if the user continuously breaks all the bricks then it will be the highest score and then user will be the winner., if the user miss the ball to bounce on paddle the ball fall to the bottom of the screen then user will loose his score and also he loses the game and then restart will be required.

Background Screen: The Background of the game screening will be in Light Gray.

Defining Walls: We have filled the walls with black color-border on left, right, top sides of the screen.

Defining Ball: We have a ball which strikes the bricks in black color oval shape.

Movement of ball: We defined the movement of the ball in X-axis and Y-axis and after bouncing on walls or paddle or bricks, it will change its direction of movement in different angles on X-axis and Y-axis.

Defining Paddle: We have a colored paddle which is the base of the ball at the bottom of the screen,its movement will depend on the values of X-coordinate that will get according to the user input that is using left and rignt arrow keys.

Intersection of Paddle and Ball: We have made the ball in such a way that it will bounce on the paddle and it will change its movement according to X and Y-axis direction, without intersecting on the paddle the ball will not go to top in game screen.

Bouncing of Ball on Wall: We also made the ball in such a way that it also bounces on the top sides of the wall and then the ball direction will be changed according to X and Y-axis.

Arrangement of Bricks: We have arrange the Bricks in 3 rows and 7 columns i,e. Total there are 21 bricks are arranged in the game and we also defined the color of bricks as Magenta.

Collision of Ball with Bricks: Here the collision of ball with the bricks is done when the ball is pushed up with a certain angle, the ball touches the walls or bricks based on the angle.

Winning Condition: Here when the ball hits the bricks then, the user gets 5 points based upon that when user breaks all the 21 bricks without letting the ball to the bottom of the paddle then, user will be the winner.

Game Score: When the ball breaks one brick then user will get 5 points simultaneously when the ball falls down the user will lose the game and user needs to restart the game, when all bricks will be broken then user will get 5 points for each brick and the user will be the winner.

Rules of Game:

1. This is an interactive game based upon the classic game brick breaker.
2. The purpose of this game is to break the bricks which are on the top of the screen which resides at various angles from the ball.
3. The ball from the paddle moves only in 5 directions :
    - Blue left paddle -> 150 degree angle
    - Yellow left paddle ->120 degree angle
    - Green middle paddle ->90 degree angle

- Yellow right paddle ->60 degree angle
- Blue left paddle -> 30 degree angle

4. When the ball hits the brick then user gets 5 points for each brick.
5. When the ball miss and falls down then user loses his/her game and the game ends up there.
6. When User completes the breaking of all bricks then user will be the winner.
7. We also have paddle which is the base of the ball for the pushing the ball to the top in various angles, this paddle is flexible to move in horizontal direction.

Environment Set Up:

In order to build game application, we use

- Software->Eclipse IDE
- Language->Java

We use windows version of Eclipse (version: -2020-03-R).So, download this version of Eclipse.
https://www.eclipse.org/downloads/packages/

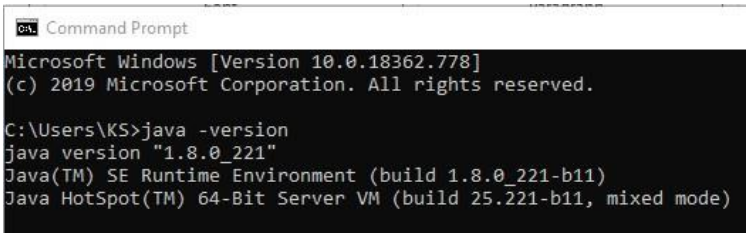
In order to install Eclipse, we need to install Java SE Development Kit.

1. Download Latest version of Java SE Development Kit 11 from this link:- https://www.oracle.com/java/technologies/javase-jdk11-downloads.html
2. Java SE Development kit setup is appeared on the screen. Click on Next
3. Then, Java SE Development Kit installation path shows on the screen, Change the path and Click on next or path will remain by default then Click on next. Then, installation is started.
4. Finally, we will see Java SE Development Kit is successfully installed. Then, Click on close.

And we have to install Java JDK in order to install Eclipse

1. Download Latest version of Java JDK from this link:- https://www.oracle.com/java/technologies/javase-jdk14-downloads.html
2. Just downloading and installing Java JDK will not run Java. For that go to C:Drive -> Program Files -> Java -> jdk -> jdk-12.0.2 Then, copy the path -> C:\Program Files\Java\jdk-12.0.2
3. Then ,go to Control Panel
    a. Click on System and Security ->System
    b. On left side, we get Advanced System Settings, Click on it.
    c. Click on Environment Variables Option
    d. On System Variables, search for path and click on Edit.
    e. On Edit Environment Variables, add this path:- C:\Program Files\Java\jdk-12.0.2

f. Then, on Edit Environment Variables ,click on OK
g. Then, Click Ok on Environment Variable
h. Whether Java JDK is successfully installed or not .Then Type this command **java –version** on cmd:



And finally to install and run Eclipse IDE for java developer, open the installer package:-

1. Click on install
2. Click on Accept Now in the Eclipse Foundation User Agreement. Then, our installation begin.
3. Then, Click on launch
4. Eclipse IDE for java developer is opened in the screen
5. Select any directory as workspace and Click on launch
6. Then, Eclipse Welcome Screen is appeared on the screen.
7. Then, Click on File -> New -> Java Project
8. On Java Project, give any project name for example – MiniProject.Then,Click on next.
9. Then, Click on Finish.
10. On the left side, there is Project Explorer. Click on Project file that we have created as MiniProject
11. Under MiniProject,we have source file -src. Right-Click on that src and click on New -> Class.
12. On Java Class, give any package name for example brickBreaker, and give any class name for example MainFile.
13. Then,we type program in this MainFile .To Compile ,Click on Run on the Menu Bar.

We attached four files:-

- BrickDestroyer.java
- BricksMap.java
- MainFile.java
- Brickbreaker.jar->For direct run

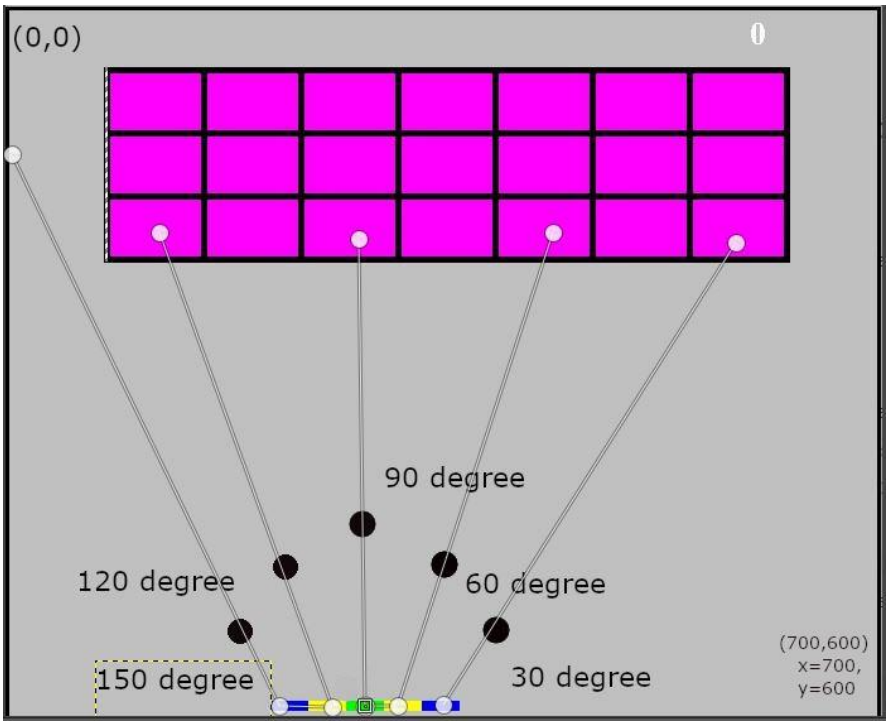
To run those file, open Eclipse IDE:-

1. Click on File ->New ->Java Project
2. On Java Project, give project name- MiniProject
3. Under MiniProject,we have source file -src. Right-Click on that src and click on New ->File->Advanced->Click on Link to file

Then,import one by one:-

- BrickDestroyer.java
- BricksMap.java
- MainFile.java

Explanation of algorithm:



In the above figure, in order to hit a ball in different angle

We have sub-divided one paddle into 5 sub-paddle.
Each sub-paddle has different angle when a ball get hit from them.

- Blue left paddle -> 150 degree
- Yellow left paddle ->120 degree
- Green middle paddle ->90 degree
- Yellow right paddle ->60 degree
- Blue left paddle -> 30 degree

In the source code,

- Blue left paddle -> paddleX1
- Yellow left paddle ->paddleX2
- Green middle paddle ->paddleX3
- Yellow right paddle ->paddleX4
- Blue left paddle -> paddleX5

```
if(play) {
    if(new Rectangle(ballposX,ballposY,20,20).intersects(new Rectangle(paddleX1,550,30,8))) //intersecting of ball with blue left paddle
    {
                ballXdir=3;
                ballYdir=-ballYdir;
                ballXdir=-ballXdir;// with intense speed in X-coordinate
                //the ball will move 150 degree from blue left paddle
    }
    else if(new Rectangle(ballposX,ballposY,20,20).intersects(new Rectangle(paddleX2,550,30,8)))    //intersecting of ball with yellow left paddle
    {
        ballXdir=1;
        ballYdir=-ballYdir;
        ballXdir=-ballXdir;
        //the ball move 120 degree from the yellow left paddle
    }

    else if(new Rectangle(ballposX,ballposY,20,20).intersects(new Rectangle(paddleX3,550,30,8)))    //intersecting of ball with green middle paddle
    {
                ballXdir=0;/*when the ball intersect green middle paddle then, ball should move
                opposite direction without reflecting in X-axis*/
                ballYdir=-ballYdir;/*when the ball intersect green middle paddle then, ball should move
                opposite direction with reflecting in Y-axis*/

                //the ball will move 90 degree angle from green middle paddle.
    }

    else if(new Rectangle(ballposX,ballposY,20,20).intersects(new Rectangle(paddleX4,550,30,8)))    //intersecting of ball with yellow right paddle
    {
        ballXdir=-1;
        ballYdir=-ballYdir;//by this,ball will move towards positive direction()
        ballXdir=-ballXdir;
        //the ball will move 60 degree from yellow right paddle
    }

    else if(new Rectangle(ballposX,ballposY,20,20).intersects(new Rectangle(paddleX5,550,30,8)))    //intersecting of ball with blue right paddle
    {
                ballXdir=-3;    // change the value
                ballYdir=-ballYdir;
                ballXdir=-ballXdir;//so that it change direction 30 degree angle with intense speed in X-coordinate
```

Source Code:If Ball get hit or intersects paddle

Begin

If play (true)

If ball intersects paddleX1

Then, state(ball) change to opposite direction (Since, intersection with blue left paddle, the ball will move at 150 degree angle from it)

Else_If ball intersects paddleX2

Then, state(ball) change to opposite direction (Since, intersection with blue left paddle, the ball will move at 150 degree angle from it)

Else_If ball intersects paddleX2

Then, state(ball) change to opposite direction (Since, intersection with blue left paddle, the ball will move at 150 degree angle from it)

Else_If ball intersects paddleX2

Then, state(ball) change to opposite direction (Since, intersection with blue left paddle, the ball will move at 150 degree angle from it)

Else_If ball intersects paddleX2

Then, state(ball) change to opposite direction (Since, intersection with blue left paddle, the ball will move at 150 degree angle from it)

```
if(ballposX <0) { //ball while intersecting the left black wall with x-coordinate

    ballXdir=-ballXdir; //ball will change direction which move to other direction after getting hit by the wall
}
if(ballposY <0) { //ball while intersecting at top corner of the left black wall

    ballYdir=-ballYdir; //ball will change direction which move to other direction
}
if(ballposX >670) {//ball while intersecting at top corner of the right black wall

    ballXdir=-ballXdir; //ball will change direction which move to other direction
}
```

Source Code:If ball get hit or intersects the walls

If Ballposition in X-coordinate< 0(While intersection the left wall with corresponding to x-coordinate)

Then, ball will change direction which move to other direction after getting hit by the wall.

If Ballposition in Y-coordinate< 0(While intersection the top wall with coordinate to y-coordinate)

Then, ball will change direction which move to other direction after getting hit by the wall.

If Ballposition in X-coordinate > 670(While intersection the right wall with corresponding to x-coordinate)

Then, ball will change direction which move to other direction after getting hit by the wall.

```
if(ballRect.intersects(brickRect))
{
    map.setBrickValue(0, i, j);
    score+=5;
    totalBricks--;

    // when ball hit right or left of brick
    if(ballposX + 19 <= brickRect.x || ballposX + 1 >= brickRect.x + brickRect.width)
    {
        ballXdir = -ballXdir;
    }
    // when ball hits top or bottom of brick
    else
    {
        ballYdir = -ballYdir;
    }
```

Source Code:If ball get hit or intersects bricks

If state(ball) intersects Brick

Then,score will be increased by 5

totalBricks will decreased by 1

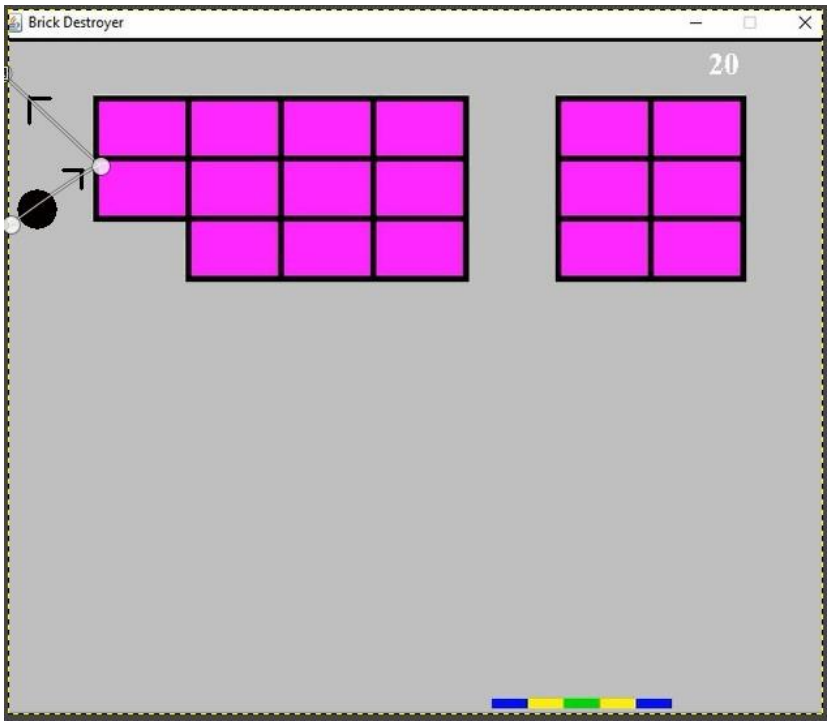
if ball hit from left or right of brick

Then,it will reflect ball in X-coordinate.
i.e.ball direction in X-axis = -(ball direction in X-axis)

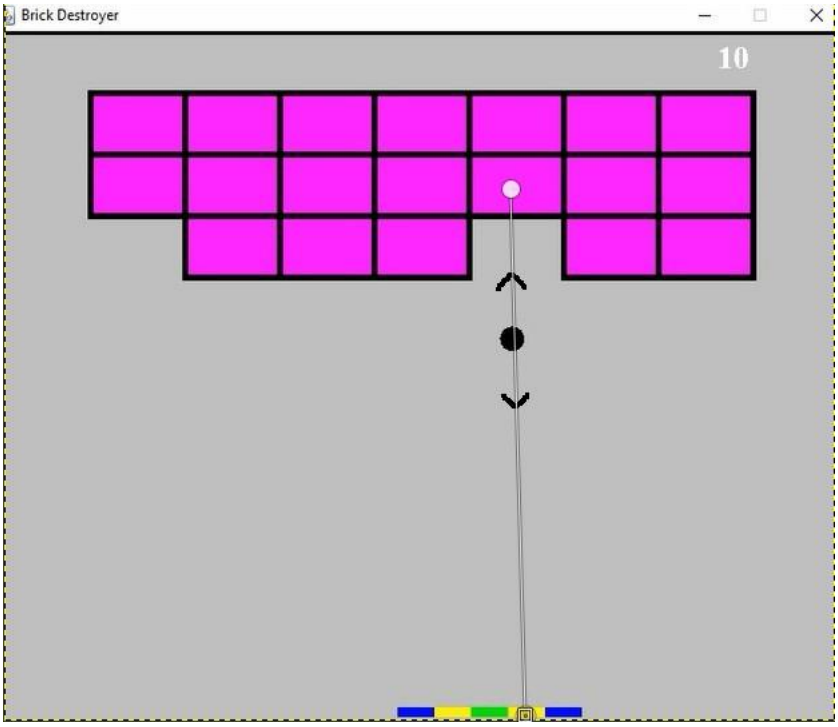
If ball hit from top or bottom of brick

Then,it will reflect the axis of the ball in Y-coordinate.
i.e.change to opposite direction
ball direction in Y-axis = -(ball direction in Y-axis)

End



In the above figure,the ball navigate to upward direction of the screen after striking the left side of a brick .This is due to ballXdir =-(ballXdir) which leads to change the opposite direction when the ball bounce the left part of a brick.



In the above figure,after striking brick straight upward,the the ball navigate back to downward of the screen this is due to ballYdir=-(ballYdir) which leads to change the opposite direction in Y-axis when the ball bounce the bottom part of a brick.

Game Credits:

KV Team members:-

Game designed by Karan Soren(17MCMC12)

Game debugged and documented by Balakisti Vijay(17MCMC27)