

BRICK BALL BREAKOUT GAME

¹Shekhar Sonkar, ²Omkareshwar, ³Sheikhadnan, ⁴Harsh Dwivedi

¹Shekharsonkar34@gmail.com

²Omkareshwar879@gmail.com

³Sheikhadnan01sa786@gmail.com

⁴Harshdwivedi1097@gmail.com

^{1,2,3,4}BCA 6th Semester, Kalinga University, Raipur

Abstract

Brick Ball Breakout is an arcade Java game that seeks to facilitate players with a satisfying gaming experience. It has a bat, a small metal ball, and rows of rectangles or triangles stacked in different patterns. The action is to control the paddle so that the ball hits it and destroys the bricks and completes the level. The player has to go through levels with gradual complexity where they are confronted with quicker balls, different brick patterns, and specific power-ups. Game development with Java is feasible due to the object-oriented paradigm that allows for abstractions required for integrating game mechanics, collision detection, and input handling. Furthermore, graphically, Swing, which is part of Java, is used to draw the graphics for the game to make the game more attractive. With the help of this project, one can experience Java programming in the context of a classic gaming environment and its potential for further development.

Introduction

Java is preferred by game development companies as well as for the creation of modern, interactive applications. Creating mobile games. So it is a matter of no wonder why the long list of available modules keeps on broadening, since the language is very versatile and it has a really large collection of open-source content. Countless successful mobile games worldwide have Java as their platform for development. Among typical names which may familiar to you are Minecraft, Mission Impossible II, and Asphalt 6 a few. We are planning to make a game incorporating the latest java concepts OOPS, AWT, SWING etc.

Project Context

Embedded in the activities is the Breakout Ball game, a game which was invented in the early



1970s by the American computer pioneers. The breaking bricks layer of bricks occupies the top third of the screen, and the objective is to settle all of them completely. The ball starting from the left-hand side of the screen moves in a straight line, striking the screen's top and sides downwards. When you miss the brick it rolls back to you and that brick breaks apart. For each time that the ball contacts the bottom of the screen, the player loses his turn. The player takes the role of a paddle which is horizontally movable and is used for bouncing the ball upward, maintaining it on the screen. The player that has the ball uses their platform to keep the ball and at the same time that brings the scoring together. The aim is to demolish the bricks without losing the ball overnight by hitting with your platform. The process involved Java swing, OOPS concepts and many other programming languages.

Literature Review

Computer games have become one of the vital components that children largely use into their spare time. Moreover, these are a significant feature of our culture as a whole meanwhile. It is not rarely as adults to see kids spend many hours emulating what they are watching. Few children take the role of football coaches while other become designers of spaces and places yet others are into creating empires while some are interested in brands, offerings and services interaction with machines. Some children chose to become wizards and emperors. In the past, video games have been regarded as a damnable activity to engage in in opposition to more important tasks like home-works or playing outside. Nowadays, what now is initiated in schools: the usage of this powerful medium by researchers, teachers and the designers of learning resources is being asked to help children to learn instead of hindering it. Instead of using a metaphor of door closing the free movement of goods and services are allowed. whereas did a traditionalist school against the game of computer, there is now much preference to find out that whether game might be a new tool for teaching in the information era. The essay is intended as a review that gives a helicopter view of the current trends in the use of computer games as a driving force for educational learning both from inside and outside the school. It concentrates on the major areas of the respective research, for example, the games' potential to promote both teaching through enjoyment, learning by doing and learning through uniting, among others. However, the speakers are careful and balanced enough to point out the intricacies and the possible thorns to using games today in our educational systems and models of learning. Next, the essay reveals the ways in which the study suggestions researchers, designers, and educational policy makers can use this huge fact base to develop educational resources and learning environments that do more than just covering the need of a 'fun' atmosphere but actually create kid engagement that computer games seem to have all kinds of such children.

Problem Identification

In the game we have a ball, a paddle and 30 bricks, but we also have tornado, thunder and rain. I have established a vector-image of ball, paddle and some brick in Inkscape. Through creating a game process with a timer we keep the user awake. Resultant begets resultant while we are moving from one direction to another; not giving the least attention to angles. The shirt is marked with top, bottom, left and right tags. I was impressed by pybreakout game and what followed much later is another game. It was done as an open source project using available PyGame library and created by Nathan Dawson. The player moves the paddle around, trying to hit the ball into the four corners of the court. Aim to clear a the top part of a window glass.

Methodology

Brick Breaker is basically the Breakout game with the player having to smash through the bricks by deflecting a moving ball using a paddle. The paddle may be adapting to horizontal movements and is also capable of moving diagonally. functionality such as zooming in and out by moving the BlackBerry's trackwheel, the computer's mouse or the finger's touch was showed (for example when using the touchscreen). At the beginning, player has 3 lives; one life is subtracted from the score when the ball hits the bottom of the screen. It is out only when everything else been destroyed that the player is awarded a new, difficult level. There are 34 levels. I play in many stages, including the endless silver bricks in my highest. If all units are killed, the display reverts to a board with two moons and two banners. And there are many brick breakers games, some which let you to burn flaming fireballs in addition and let you to play with more than one ball if the player upgrades to this benefits. Ultimately, Brick Breaker proves to be a true crowd favorite with many a professional gamers seeking to reach high scores. The game's addictiveness was highlighted by The Vancouver Sun: It seems that "dozens of mettings in close forums, support groups and an "Addiction to Brick Breaker" Facebook page, where partners complain of spouses being addicted to the game.

Setting up the development environment

First, the project's folder set up must be done. This needs Java 8 or higher to be installed on your machine. To complement this, we use either a code-editor or integrated development environment to assemble the code.

Requirements

1. Install Java (JDK 1.8)



2. Install any IDE- IntelliJ (Community edition) or Eclipse or VSCode
3. Setup environment for Java on your local machine
4. Create a Java project using Command Line App as the template
5. Give project name and the base package

Design the Game terminal

Game terminal is the window where the game is to be played. It consists of the following aspects:

- Window size (length, breadth and height)
- Window visibility
- Game title
- Default close operation

Requirements

1. Create a Main class inside the base package
2. Import Java Swing dependency
3. Create a JFrame object to implement the GUI functionality
4. Design the window terminal by writing your code

com/brickball/Commons.java

```
package  
com.brickball;  
public interface  
Commons {  
    int  
    WID  
    TH
```

```
=
300;
int
HEI
GH
T =
400;
int
BOTTOM_
EDGE =
390; int
N_OF_BRI
CKS = 30;
int
INIT_PAD
DLE_X =
200; int
INIT_PAD
DLE_Y =
360; int
INIT_BAL
L_X = 230;
int
INIT_BAL
L_Y = 355;
int PERIOD
= 10;
}
```

Design of the gameplay

Creativity and ideas constitute game design, which covers the broader field of game development and creates a game to entertain or to educate the gamer or audience. Games can be defined by "what is players' action" and "what is players' experience". This is usually any of the consistencies with the video game that get you to interact with it, such as jumping over a pit. The core of the main game feature – a gameplay layer – is designed to bring to life the game experience. It designs the game and shapes it, and it decides the game's rules and objects it would have, and it thinks of different ways players can play.

Requirements

1. Create a Gameplay class inside the base package
2. Import Java AWT and Swing dependencies
3. Extend the JPanel container class by implementing KeyListner and ActionListener interfaces. Check out more on these classes and interfaces:
 - i. JPanel
 - ii. ActionListner
 - iii. KeyListner
4. Design the background template
5. Implement the code for ball positioning
6. Implement the code for moving the paddle using KeyEvent andActionEvent

com/brickball/Sprite.java

```
package  
com.brickball;  
import  
java.awt.Image;  
import  
java.awt.Rectangle;
```

```
public class Sprite {
```

```
    i  
    n  
    t  
    x  
    ;  
    i  
    n  
    t  
    y  
    ;  
    int  
    imag  
    eWi
```

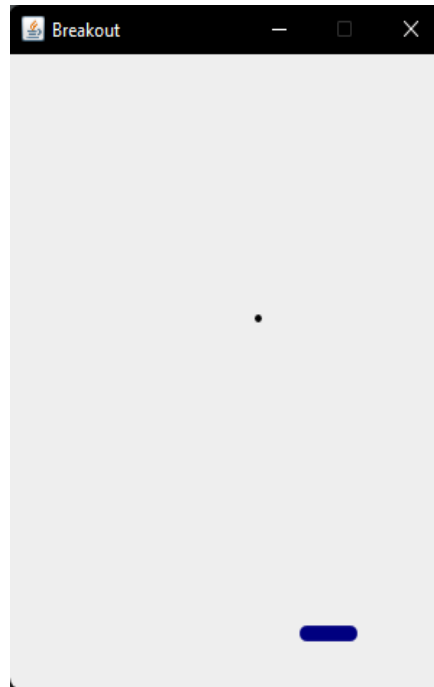
```
dth;  
int  
imag  
eHei  
ght;  
Ima  
ge  
imag  
e;  
protected void setX(int  
    x) {this.x = x;  
}  
int getX() {
```

```
        return x;
    }
    protected void setY(int y) {this.y = y;
    }
    int getY() {
        return y;
    }
    int getImageWidth() { return
        imageWidth;
    }
    int getImageHeight() { return
        imageHeight;
    }
    Image getImage() { return
        image;
    }
    Rectangle getRect() {
        return new Rectangle(x, y,
            image.getWidth(null), image.getHeight(null));
    }
    void getImageDimensions() { imageWidth =
        image.getWidth(null);
        imageHeight = image.getHeight(null);
    }
}
```

Congratulations!!! we have successfully designed the gameplay!

Expected Outcome

You should be able to view the gameplay in the terminal.



Design the props (bricks) using 2D Array

The breakout game starts with four to eight rows which have each two rows of bricks in different colors. The sequence of colours, viewed from below, is from the bottom up, yellow, green, orange and red. As the player uses a single ball, which strikes the bricks directly or, ricochets it off the walls or the paddles positioned below so as to eliminate the bricks. If the player's paddle does not intercept correctly the ball, he would lose a turn.

Requirements

1. Create a Map Generator class inside the base package
2. Import Java AWT dependency
3. Design the bricks
4. Implement the code for the bricks to disappear once the ball hits
5. Implement the code for calculating the score in the 'Gameplay' class

com/brickball/Brick.java

```
package com.brickball;  
import javax.swing.ImageIcon;
```

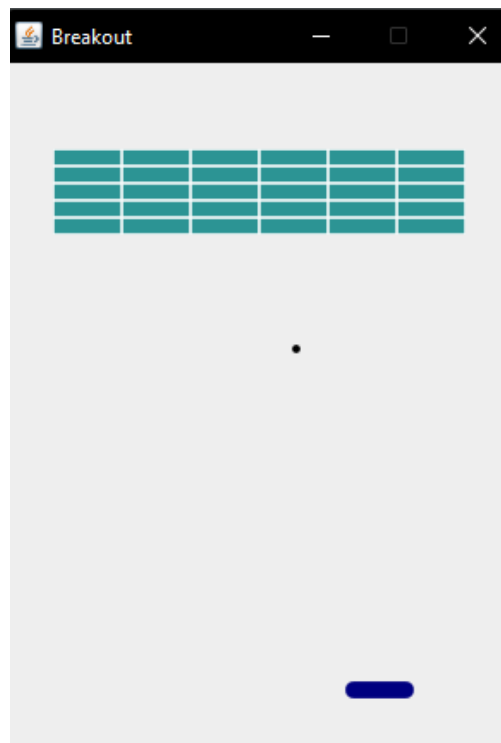
```
public class Brick extends Sprite {private boolean destroyed;
public Brick(int x, int y) {
    initBrick(x, y);
}
private void initBrick(int x, int y) {this.x = x;
    this.y = y;

    destroyed = false; loadImage();
    getImageDimensions();
}
private void loadImage() {
    var ii = new ImageIcon("src/resources/brick.png");image = ii.getImage();
}
boolean isDestroyed() {return destroyed;
}
void setDestroyed(boolean val) {destroyed = val;
}
}
```

This is the Brick class.

Congratulations!!! we have successfully designed the entire game!

Brick-Ball Gameplay



Run the game in the terminal

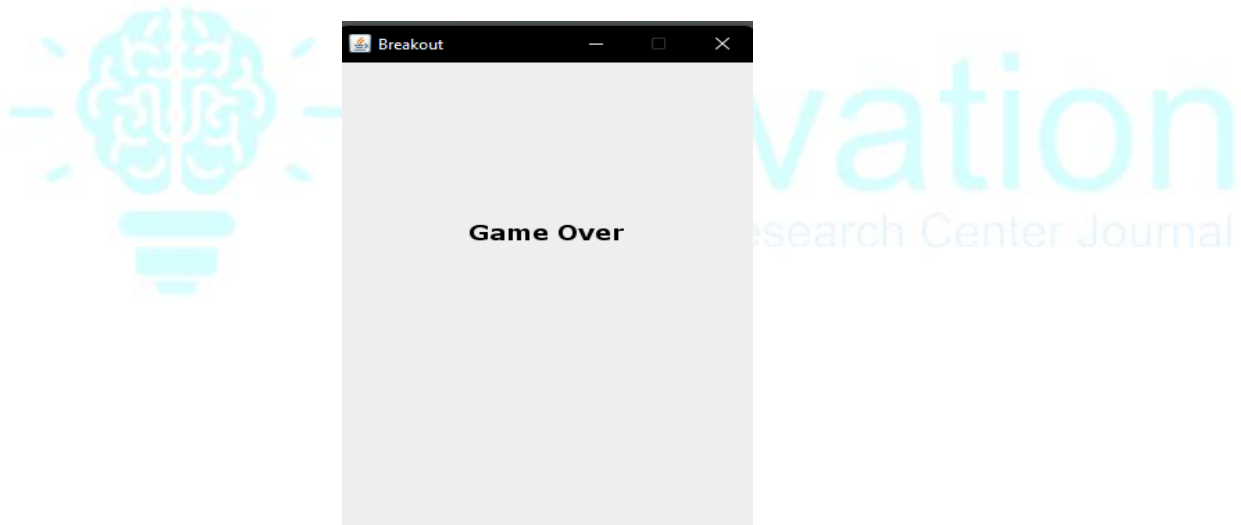
Wrap up the code of the game and start it off right away using the terminal directly.

Requirements

1. Build the project
2. Import any dependencies, if left
3. Run the Main() class
4. Play the game in the terminal

Expected Outcome

You should be able to play the game & When the ball drops it should show Game-Over Screen.



Technologies Used

This is a simple Java based mini-game inspired by the arcade game called Breakout. This game was originally developed by Atari Inc. The game was created in 1976. The aim of this game is to break bricks without dropping the ball from your platform. The project makes use of :

- Java JDK SE 19
- Eclipse IDE for Java 2022-12
- Java swing
- OOPS concepts
- AWT

- JFrame
- JPanel

System Requirements

The System Requirements for this project to run is just a Windows computer with the Latest version of Java JDK Platform installed, And the Latest Eclipse IDE for Java Development.

References

Here are some resources and websites that assisted in the development of this mini-project:

1. Java: A Beginner's Guide, Author: Herbert Schildt
2. Effective Java, Author: Joshua Bloch
3. <https://github.com/>
4. <https://zetcode.com/>
5. <https://www.crio.do/>

