

Radial Breakout Functional Specification

Dmitriy A. Dunin

CS U540

Overview

The *Radial Breakout* program is a 2D video game application, which is a variant of the classic *Breakout*. The basic goal of the game is to remove all bricks from the playing field with a ball while not allowing the ball to go out of bounds using a “paddle” to bounce the ball back into the playing field.

Unlike vanilla *Breakout*, *Radial Breakout* has no default boundaries. The player must keep the ball in bounds by moving either of the two paddles on their respective radial tracks to bounce the ball back into the playing field. Another important variation is the way the bricks are removed from the playing field. Once the ball hits a colored brick the bricks will move towards their respective gravity source until they exit the playing field. The flying bricks will act as a hazard to the player.

Due to the nature of the project and the given time constraints this iteration of *Radial Breakout* will consist of only 1-2 levels and will not feature a scoring system.

Basic Gameplay

The game is setup with two paddles on two concentric circular tracks. The paddles can move either independently or dependently of each other, but will stay on the track. These paddles are used to creating surfaces for the ball to bounce off of to return back to the playing field when it is going out of bounds. Both paddles will start at the bottom of the screen and the ball will start slightly above the paddles and move straight down. The ball’s speed and direction will change depending on paddle speed and the ball’s impact angle.

Once the ball is bounced back into the playing field it will continue traveling along its path until it either encounters a block or a paddle. If neither is encountered the ball will go out of bounds and the game will reset. When the ball hits a block the block will become dislodged and move towards its respective gravity source. Reference the following table for block color and movement.

 - North  - East  - South  - West

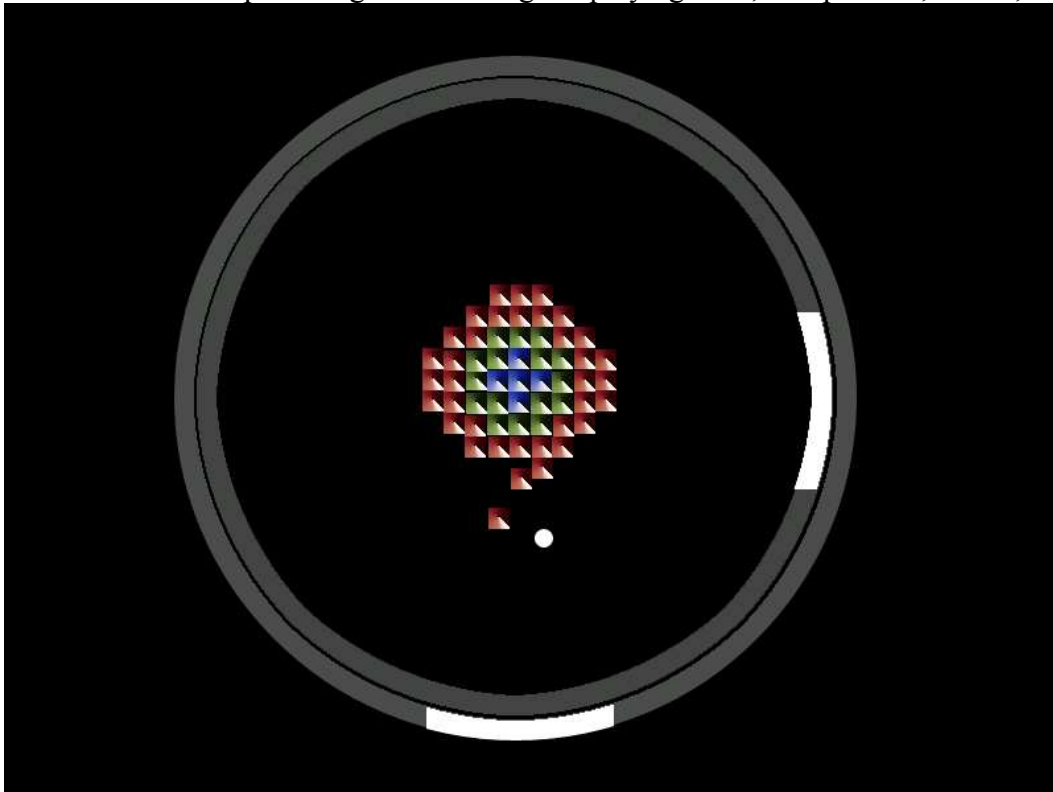
There can also be composite colored blocks consisting of 2-3 colors. These blocks will move in their respective color’s direction. For example:

 - Northeast

The composite blocks can move in any combination of the 2-3 colors. So a block colored blue and green will usually move southeast, while a block colored yellow and blue can move west but might drift east as well. This adds another degree of difficulty to the game. Composite blocks will most likely not be seen until the more advanced stages of the game.

If a moving block collides with a paddle the paddle will no longer be movable by the player. If the same paddle gets hit again the paddle will disappear. The game ends when all the blocks have been cleared or if the both paddles are gone.

Below is a mock-up of the game showing the playing field, two paddles, bricks, and ball.



Process Flow

The primary interface for *Radial Breakout* is the mouse with the keyboard providing an interface for modifiers. The mouse will control the movement of the paddles, both direction and speed. By default, only the inside paddle is moved with the mouse. Moving the mouse to the right moves the paddle clockwise along its path. Conversely, moving the mouse to the left moves the paddle anti-clockwise.

The table below displays keyboard presses and corresponding control modifications:

- F1 – Display Help (controls)
- F2 – New Game/Restart
- ALT – Move outside paddle.
- CTRL – Link paddle movement

Technical Implementation

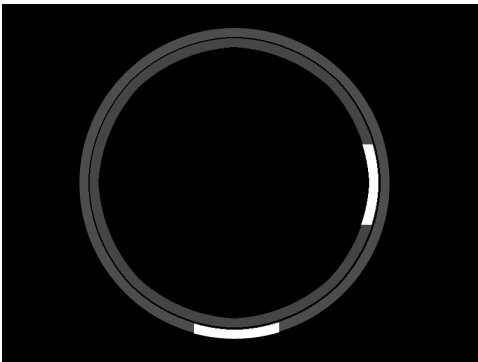
The program will be written in Java utilizing the Java 2D API. The following subsections cover 2D transformations of objects as required by the project.

Movement

The ball and blocks will move linearly along variable or pre-defined vectors, respectively. The blocks will have internal, pre-defined vectors for movement sometimes changing vectors quasi-randomly if they are composite blocks. The ball's initial movement vector is pre-defined, however, this vector will be altered upon collision with a block or paddle.

Rotation

The paddles will be made of two thick circular ellipses and the paddles will be arcs on the ellipse. The movement of the mouse will rotate the ellipse(s) and appropriate paddle arc(s).



Selection

Selection is covered under collision detection between the ball and paddle and blocks. In all cases tight hit boxes will be utilized. The paddles will have 2-3 hit boxes in order to have more accurate hit detection without sacrificing performance.