**Breakout Game**

Game Design Document

John McDaniel

Overview

The Breakout Game will be a simple 2D game that introduces programmers new to Python using pygame and simpleGE.

The overview is simple: the user controls a paddle, trying to hit a bouncing ball, and colliding with bricks at the top. The paddle appears at the bottom in front of a black background. The user can move the paddle left and right using the arrows on the keyboard. The bouncing ball will move left and right at 2 to 5 pixels and fall 2 to 5 pixels starting at a random place up at the top, making it challenging but not impossible. The bricks will be different rows starting at red at the top and working down he rainbow to blue by the fifth row. When the ball hits the brick, it will break and disappear and bounce in the opposite direction. The goal is to hit all the bricks before the ball hits the bottom of the screen.

When the game starts, it will show a screen with instructions in the middle and two buttons underneath, allowing the player to start or quit.

When the ball leaves the screen and not all the bricks are hit, the player will lose, and it will show them how long they lasted. When all the bricks have been hit, the player wins, showing them the time it took. Both screens will allow the player to restart or quit.

Game Design



The Breakout Game is a simple system that gives users two options and instructions when they start the game. Each option is represented as a subclass in the simpleGE scene class. The buttons close out the current screen but bring up two different screens depending on which button is selected. If the user hits play, then the game will start. If quit is selected, then the tab will be closed. The game over screen will end when time runs out, and a screen allowing the user to play again or quit will appear. It will also have the user's previous time on the screen, and whether they won or lost the game.

The Menu Scene

The Menu Scene is easy to understand, and the game directions, play button, quit button.



The intro has four elements:

1. Instructions
2. btnPlay
3. btnQuit
4. title

Pseudocode

Class MenuScene(simpleGE.Scene):

Create Title

Create directions label

Create Play and Quit buttons

Add directions, btnplay, btnquit, lblScore to sprites

If Play button clicked:

Set response to "Play”

If Quit button clicked:

Set response to "Quit"

The Breakout Game Class

Main gameplay scene with timer, ball, bricks, and paddle.



The game will have many visuals

1. Paddle
2. Blocks
3. Ball
4. Time

Many non-sprite assets

1. Timer start
2. powerUp
3. paddleSound

Pseudocode

Class Game (simpleGE.Scene):

Create a paddle

Create a ball

Create a timer and a label to show the time

Set label position, size, and color

Load hit sound

Load paddle hit sound

Create an empty list for blocks

Set starting x and y positions for blocks

Set gap between blocks and block width

Define a list of colors for each row

For 5 rows:

For 10 columns:

Calculate x and y position for each block

Pick color for the row

Create a block with the position and color

Add the block to the block list

Add all sprites paddle, ball, label, blocks to the game

Start the timer

The Game Over Class

Game over class has the results, time it took, play, and quit options.



Pseudocode

Class GameOver(simpleGE.Scene):

Create Results

Create Time

Create Play and Quit buttons

Add results, btnplay, btnquit, time to sprites

If Play button clicked:

Set response to "Play”

If Quit button clicked:

Set response to "Quit"

Ball Class

Represents a basketball bouncing around the screen.

Pseudocode

Set size to 20 by 20

Set color to red

Set random starting x position

Set y position to 300

Set random horizontal speed (dx)

Set random upward speed (dy)

Make the ball bounce off screen edges

Move the ball by its speed (dx, dy)

If ball hits left or right wall:

Reverse x direction

If ball hits top wall:

Reverse y direction

If ball goes below screen:

Print "Game Over"

Get elapsed time

Start game over scene with time

Stop the game

If ball hits the paddle:

Reverse y direction

Move ball above the paddle

Play paddle sound

For each block in the block list:

If ball hits the block:

Reverse the y direction

Hide the block

Remove the block

Play hit sound

Paddle Class

Player-controlled paddle at the bottom of the screen.

Pseudocode

When paddle is created:

Set size to 100 by 20

Set color to gray

Set vertical position to 450

Each frame (process):

If left arrow is pressed:

Move left

If right arrow is pressed:

Move right

Keep paddle inside left screen edge

Keep paddle inside right screen edge

Block Class

Make each block

Pseudocode

When block is created:

Set size to 50 by 20

Set block color

Set block position (x, y)

Main function

The main function will start the code

Psuedocode

Function main()

Create the menu screen

Start the menu

Milestone Plan

1. Set the scene, by making rows of blocks

2. Add Ball bouncing

3. Add keyboard motion with the paddle at the bottom of the screen

4. Make sound effect when ball hits paddle

5. Make sound effect with ball hits blocks

6. Remove blocks when hit

7. Make the player lose when ball hits ground

8. Make player win when no blocks left

9. Make a start screen

10. Make a game over

Assets Plan

powerUp.wav

Custom audio by John McDaniel with jsfxr:

https://sfxr.me/#11111F3e6QCeRPdThqJGS1cS6Mk1QEoaZrTcemntnpZzaJLDQbgWKk89gQYQCtDwBdrFbXaN5M7iwUtJCyfjRWMCiqpN3jp6qyATEveQqLpqUUZiDTpWzmn7

blipSelect.wav

Custom audio by John McDaniel with jsfxr:

https://sfxr.me/#34T6PkrhthoR8FnXpZXV3nNvyR9RVXhXwBZkmh2torsbyHgvBqvSwdoe6KSTPphSytGDJzUezKDSYvq76ZzkyFM9j7GCizoR7Xkway4R6gfZxwABsWFZWEaj1