**CSc 102 Project Proposal: Brick Breaker Matrix v1.10**

**1. Title**

**Brick Breaker Matrix v1.10**

**2. Background and Motivation**

***Brick Breaker*** is a ***Breakout*** clone in which the player must smash a wall of bricks by deflecting a bouncing ball with a paddle (Wikipedia, 2024). The paddle may move horizontally and is controlled with the computer’s mouse or the touch of a finger (in the case of touchscreen). The player gets three lives to start with; a life is lost if the ball hits the bottom of the screen. When all the bricks have been destroyed, the player advances to a new, harder level. There are 34 levels. Many levels have unbreakable silver bricks. If all lives are lost, the game is over (Wikipedia, 2024).

There are many versions of ***Brick Breaker***, some in which players can shoot flaming fireballs or play with more than one ball if the player gets a power-up (Wikipedia, 2024). Our version of the game will have a split-screen multiplayer mode, where the player with the highest score when all the lives of both players deplete wins. It will also include timed power-ups which the player will get when they destroy certain bricks. These power-ups include more balls, a longer paddle, and a stronger ball that can destroy a brick with one hit (the normal ball will take 2 hits to destroy a brick). We will implement a scoreboard and a life counter to allow the player to keep track of their progress and make the game more fun. The aim of this version of *Brick Breaker* is to destroy as many bricks as the player can before the player’s lives deplete. We will also implement sound effects and background music.

**3. Problem Statement**

The aim of this version of *Brick Breaker* is to destroy as many bricks as the player can before the player’s lives deplete. This game has a moderate complexity, but features will be added in an attempt to increase its complexity, attractiveness, and depth for casual and competitive play.

**4. Approach**

This project aims to implement a Brick Breaker game in Java with the following milestones:

* **Milestone 1**: Code a data structure (perhaps an array) that can contain the layout of bricks and the paddle.
* **Milestone 2**: Implement the ball’s movement and collision detection with bricks, paddle, and screen boundaries.
* **Milestone 3**: Add power-ups and their effects on gameplay.
* **Milestone 4**: Implement a split-screen multiplayer mode.
* **Milestone 5**: Create a scoreboard and timer to track player progress.
* **Milestone 6**: Add sound effects and background music to enhance the gaming experience.
* **Milestone 7**: Develop an easy-to-use GUI to improve the attractiveness of the game.
* **Milestone 8**: Present the work for assessment in the form of a GitHub code repository and Project Document, which will include the revision and extension of this proposal. The extensions will include a walkthrough of the steps through the SDLC that our group followed, and a Visual Guide that shows the game in action. We will do this through ‘storyboards’ with an explanation of each frame, that will show the start of the game, playing example of the game, and the end of the game.

**5. Technologies and Concepts**

* **Object-Oriented Programming (OOP):** We will use OOP principles such as abstraction, inheritance, encapsulation, and polymorphism to design and implement the game. Classes and interfaces will be used to represent game entities and their interactions.
* **Collection Framework**: We will use Java’s Collection Framework to represent groups of objects as single units, such as lists of bricks and power-ups.
* **Java Exceptions**: Exception handling will be implemented to manage errors and ensure the game runs smoothly.
* **Input/Output Streams**: We will use I/O streams to handle file operations, such as saving and loading game states.
* **Multi-threading**: Parts of the game will run concurrently to improve performance and responsiveness.
* **LibGDX Framework**: We will use LibGDX, a cross-platform game development framework based on OpenGL (ES), to develop the game for both Windows and Android platforms. LibGDX allows us to write the game code once and deploy it on multiple platforms.
* **GitHub**: We will use GitHub to track changes, coordinate work, and manage development.
* **Java Swing**: We will use Java Swing, a GUI toolkit, to create the game’s user interface, including JFrame, JPanel, and JButton.

**6. Timeline**

| **Milestone** | **Date** |
| --- | --- |
| Proposal | August 14 |
| Design | August 31 |
| Milestone 1: Code a data structure for bricks and paddle | Week 1 |
| Milestone 2: Implement ball’s movement and collision detection | Week 2 |
| Milestone 3: Add power-ups and their effects | Week 3 |
| Milestone 4: Implement split-screen multiplayer mode | Week 4 |
| Milestone 5: Create scoreboard and timer | Week 5 |
| Milestone 6: Add sound effects and background music | Week 6 |
| Milestone 7: Develop an easy-to-use GUI | Week 7 |
| Milestone 8: Present the work for assessment | Week 8 |
| Implementation | October 6 |
| Presentation | During the week that starts on October 7 |

**6. References**

* Wikipedia contributors. (2024, May 29). Brick Breaker. In Wikipedia, The Free Encyclopedia. Retrieved 20:08, August 8, 2024, from https://en.wikipedia.org/w/index.php?title=Brick\_Breaker&oldid=1226287109