

Intro

Overview

Rules

Reqs

Solutions

UML

Collab

Testing

BRAWL SNAKE

Binary Brawlers: Cooper Braun, Navin
Kunakornvanich, Bradley Russell, Francesca
Strickland-Anderson



Intro

Overview

Rules

Reqs

Solutions

UML

Collab

Testing

PROJECT OVERVIEW



Main features

- Movement & growth
- Food Effects
- Difficulties
- Collision Detection
- Game Board

Limitations

- No visual feedback for special effects
- No multiplayer
- Snake wrapping mechanic bug where it runs into itself

Assumptions

- The player knows the rules and controls of the game
- Food and obstacles can't spawn in the same location
- Snake's initial size and speed are fixed at the start and consistent with the difficulty levels

[Intro](#)[Overview](#)[Rules](#)[Reqs](#)[Solutions](#)[UML](#)[Collab](#)[Testing](#)

Description

- A single-player game where the snake eats food to grow while avoiding collisions with itself, obstacles, and the board's edges.

Rules

- Objective - Eat food to grow and score points without colliding with obstacles, edges, or yourself
- Controls - Use arrow keys or WASD to move the snake
- Food - The reds apple grows the snake by one segment, the golden apple grants five segments, and the star fruit grants invincibility for a short period



Intro

Overview

Rules

Reqs

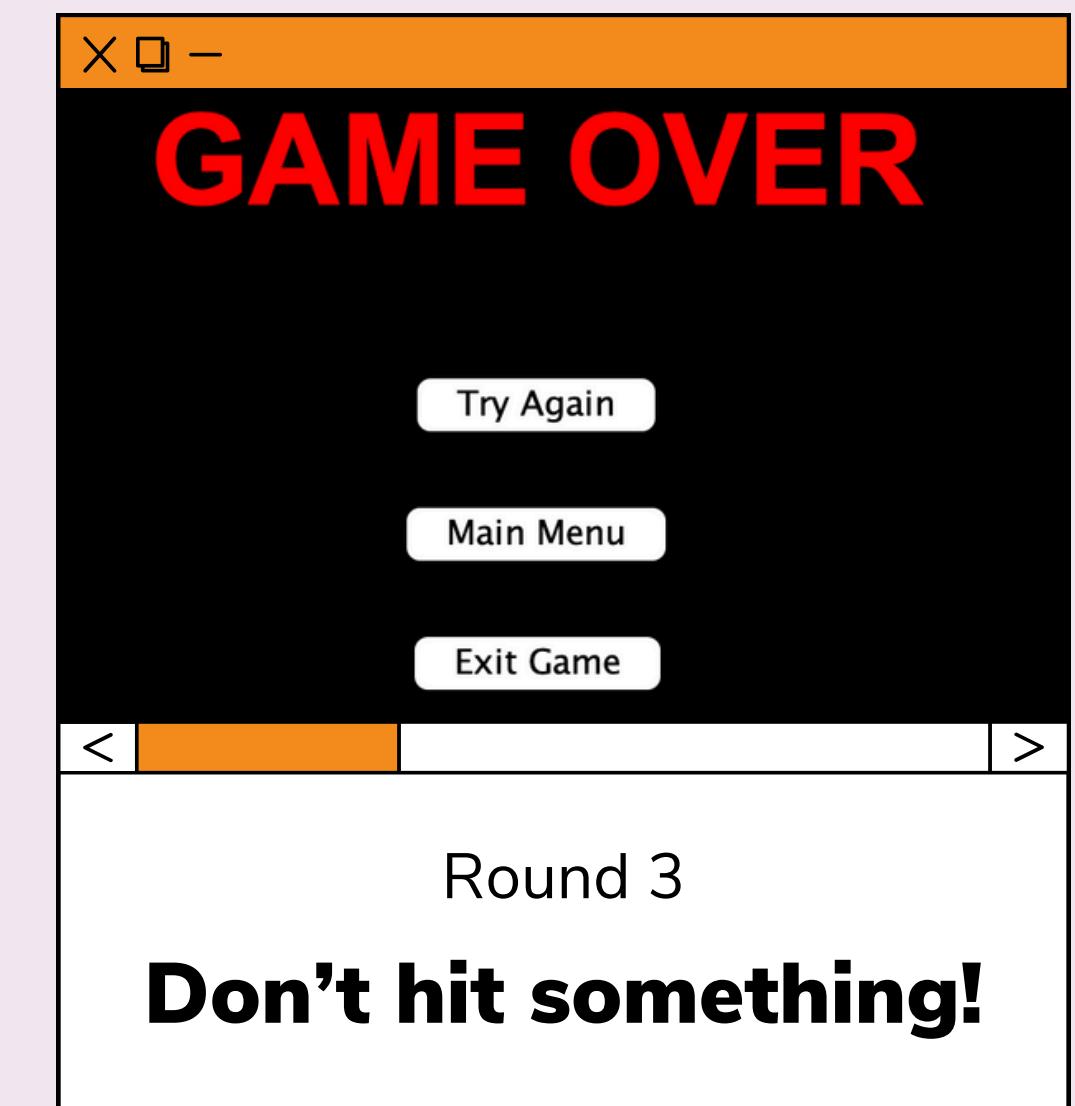
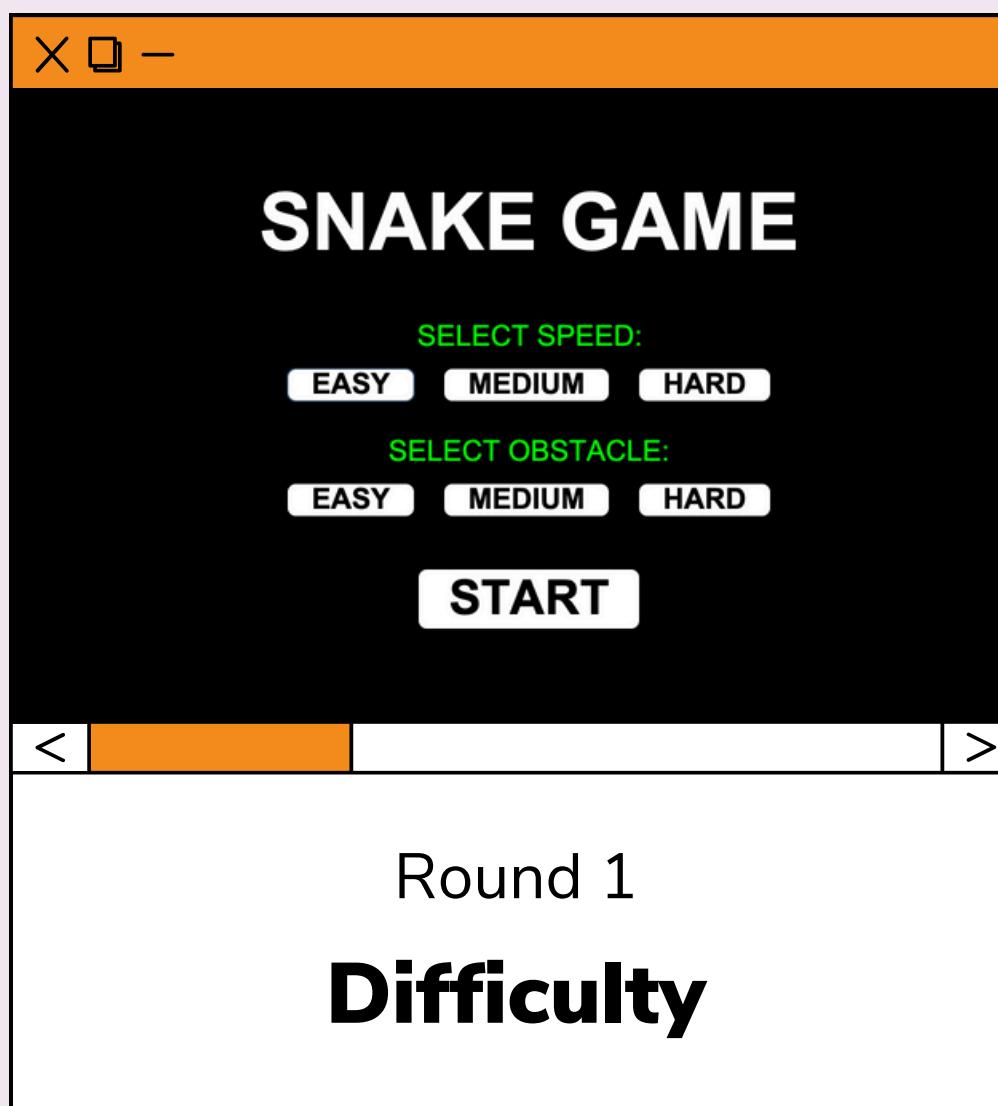
Solutions

UML

Collab

Testing

IMAGES



Intro

Overview

Rules

Reqs

Solutions

UML

Collab

Testing



FUNCTIONAL:

- Movement in four directions, collision detection, & growth

○ ○ ○ ○

NON-FUNCTIONAL

- The game should run smoothly with minimal lag
- The user interface must be intuitive (clear visuals)

Intro

Overview

Rules

Reqs

Solutions

UML

Collab

Testing

SOLUTION APPROACH

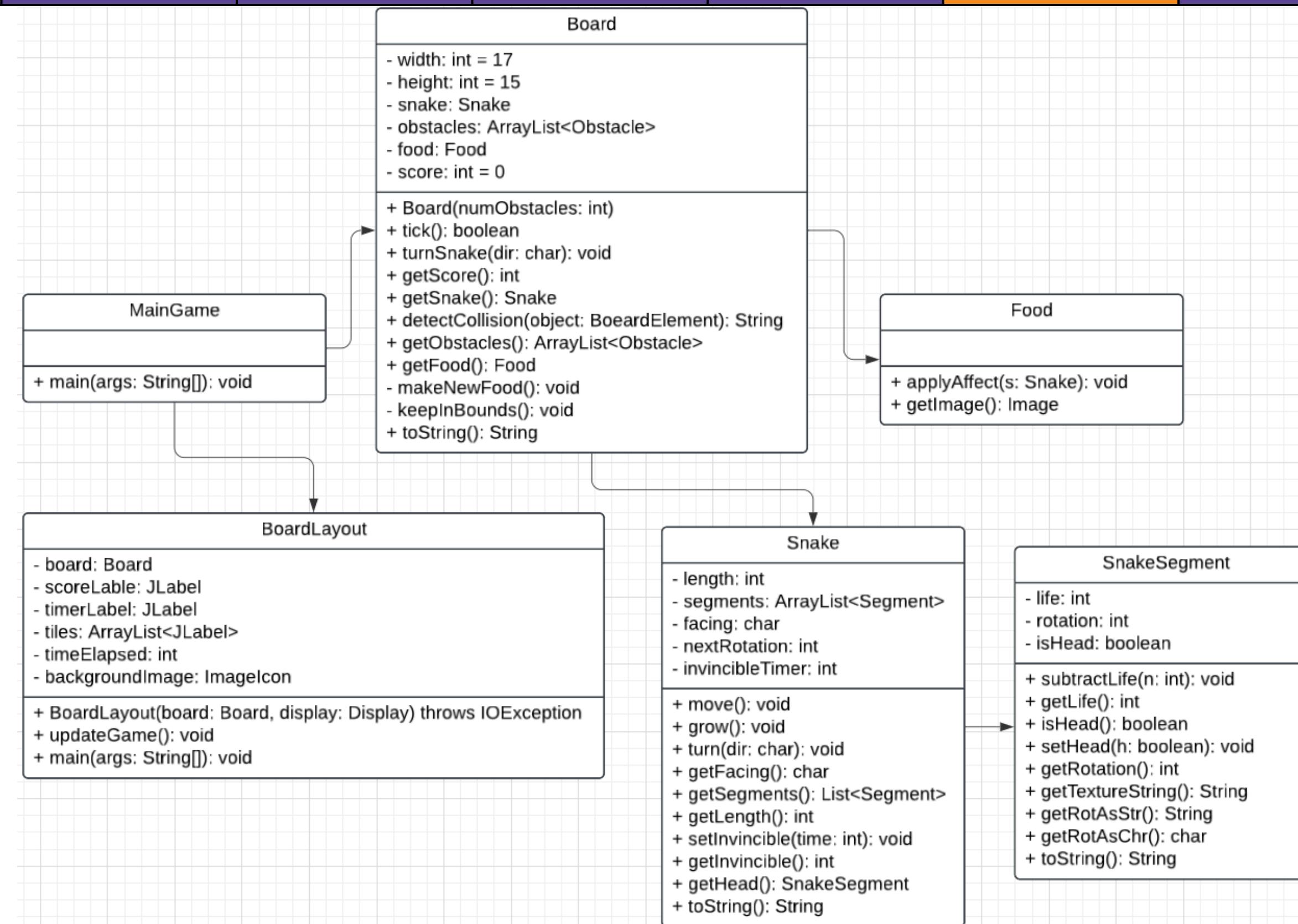


→ Major Components

- Game Logic - Handles snake movement, food consumption, collision detection, and scoring. Ensures game state updates occur in sync with player inputs
- Backend Classes - Snake and SnakeSegment define the snake's structure and movement. Food and its subclasses provide regular and special food behaviors. Board manages the game environment.
- Frontend Classes - Game Window and Panels display the game board, snake, and UI elements like the score and timer.

→ Important Features

- Difficulty levels that adjust the levels of speed and obstacles
- Special food items to add depth
- Smooth and immediate response to directional input from the user



Intro

Overview

Rules

Reqs

Solutions

UML

Collab

Testing

TEAM COLLABORATION

Communication

- ★ Group chat
- ★ Group meetings

Difficulties

- ★ Getting used to coding in a team/different styles

Lessons

- ★ Communication
- ★ Deadlines

Intro

Overview

Rules

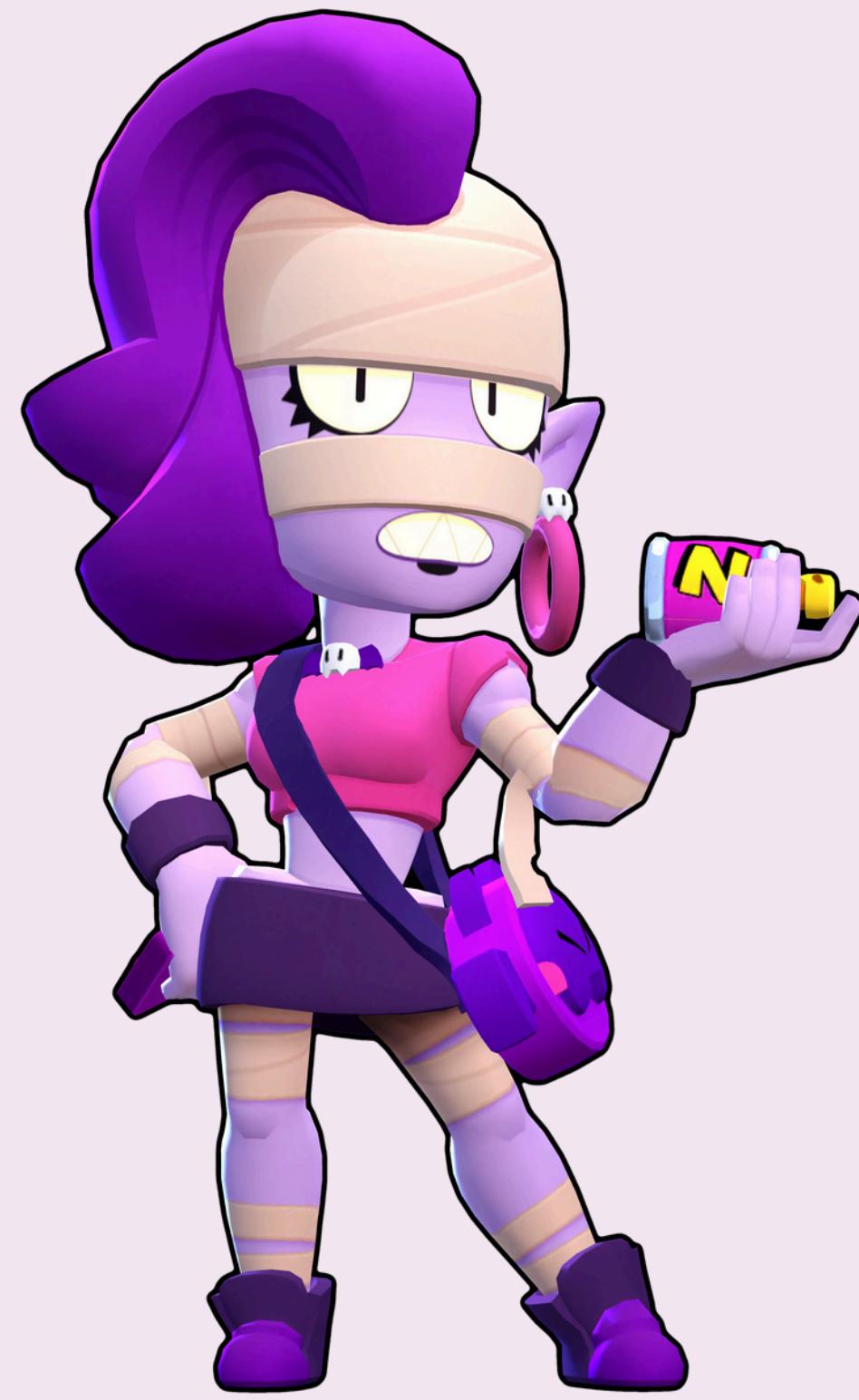
Reqs

Solutions

UML

Collab

Testing



User Testing

- We relied on friends and roommates to try the game and provide feedback on gameplay, control, and the look of the game

Deliverability

- Considered deliverable when users report smooth gameplay, no critical issues, and an engaging and fun experience across all difficulty levels

