Test Plan: Ricochet Rage

CPSC 427 – Video Game Programming

Term: 2024W1

Instructor: Chris Kerslake

Team: Sigma Studios

Milestone: 3

Abhigyan Dabla (28266047)

Maysam Farahani (95167003)

Jason Liu (16585622)

William Cheng (86583648)

Sean Ford (42305409)

# Table of Contents

[**Table of Contents 2**](#_hhz74rcrd7i8)

[**Setup 3**](#_jb22w985ub07)

[**Controls 4**](#_e1a90kt0jg0d)

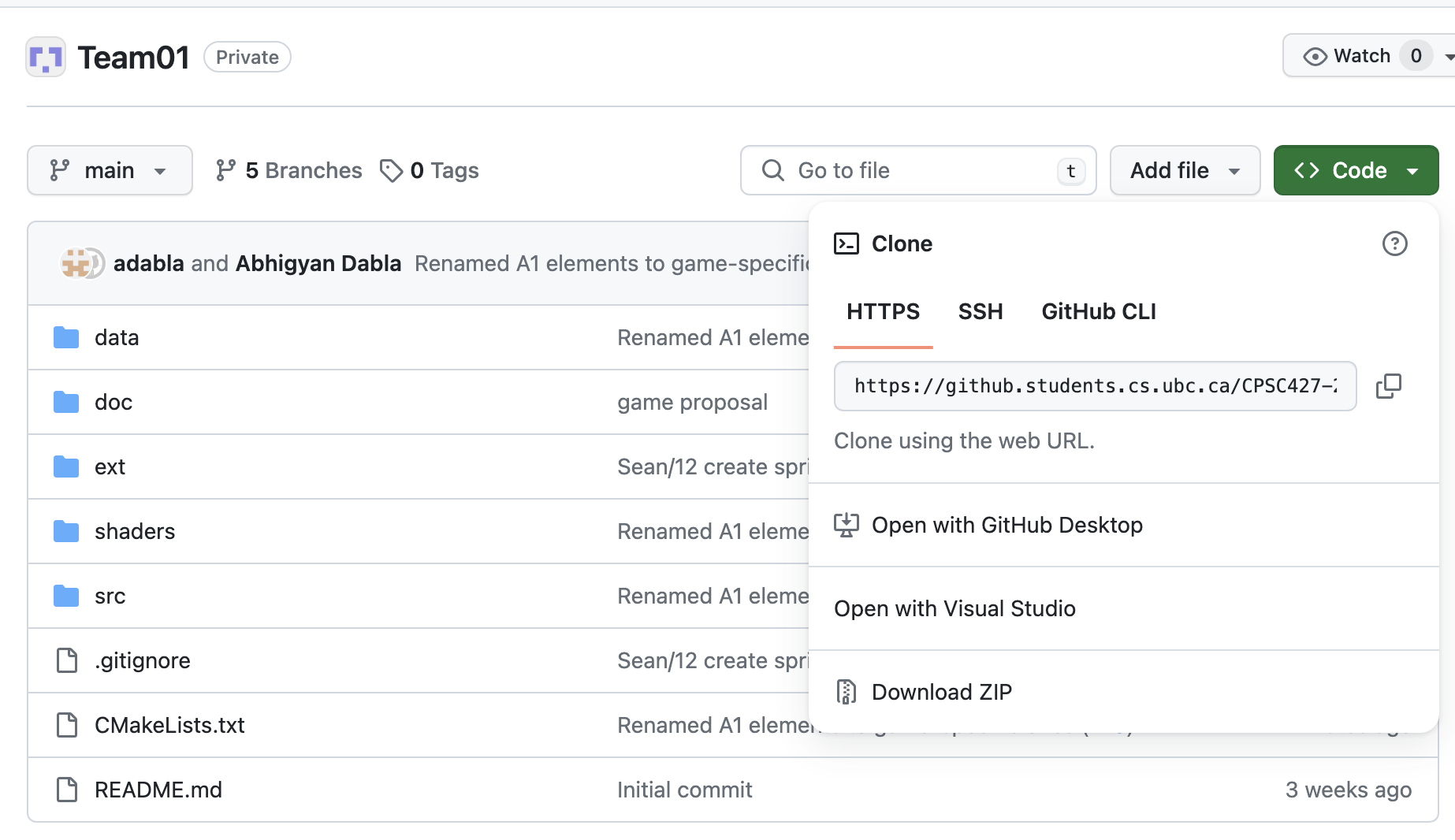
[**Features 5**](#_t40s4brq4uy)

# 

# Setup

In order to test *Ricochet Rage*, please follow the steps below to set up your development environment and run the game:

1. Set up your development environment to be compatible with C++ 14, CMake, and OpenGL if you have not already. Instructions for environment setup can be found on Canvas [here](https://canvas.ubc.ca/courses/147789/pages/tutorial-02-dev-env-setup?module_item_id=7236387).
2. Next, clone the GitHub repository to your local machine or download a copy as a ZIP file, then uncompress on your machine.



1. After successfully downloading the codebase to your local machine, open a terminal and create a new directory within the project called build/ by running mkdir build
2. Change into the directory by running the command cd build
3. Generate build files by running the command cmake ..
4. Compile and build the project by running the command cmake --build .
5. Finally, run the game with the executable by running ./ricochet-rage

# 

# Controls

The following key bindings are currently implemented for *Ricochet Rage*:

* **W**: Move up
* **A**: Move left
* **S**: Move down
* **D**: Move right
* **Space**: Dash
* **Left mouse click**: Shoot
* **Escape**: Pause Game
* **Right Click or Left Click + Left Control (for Mac users):** Mouse Gestures for healing

Additionally, players can aim using mouse controls for more precise targeting.

The player can use mouse gestures to heal the player. By holding the right-click, or left-click + left control, the player can draw a heart to heal the player by 50 HP.

# Features

The following sections outline the required and creative features implemented for Milestone 3. Use the descriptions below to test and verify each feature.

**Playability**

We incorporated multiple game levels to ensure that players can enjoy 5 minutes of non-repetitive gameplay.

*To test these features, play the game and progress through the multiple game levels.*

**Robustness**

The game is free of memory leaks or components that hoard memory throughout the game lifecycle. The game window can also robustly handle window resizing and random user input. Further, the game is free of any input lag or stuttering.

*To test this feature, 1) use tools or applications like Valgrind or Task Manager to ensure that memory is not being leaked or hoarded throughout our game, 2) try spamming random inputs to the game and see no unexpected behavior or crashes take place, and 3) observe the FPS counter in the game window while testing to see the consistency in game performance.*

**Stability and Reporting**

All missed features in prior milestones (i.e. key-frame/state interpolation) have been implemented and any bugs identified during prior milestones have been resolved and updated, as seen in the bug report.

*To test the key-frame interpolation, see the color distortion of the background while in the game screen. To see how bugs have been resolved, read the “Latest Update” column in the bug report, and test out the bug fixes yourself.*

**Complex Prescribed Motion (Basic Creative Feature - 10%)**

The enemy boss teleports from time to time. The animation for the teleportation applies a Quadratic Bezier Curve multiplier to the scale of the boss. This is non-linear and requires 3 set points. As a result, the boss appears to be warping into nothing when it is teleporting.

*To test this feature, allow the enemy to teleport to another area. Determine if the enemy animation to teleport follows a non-linear Bezier curve. Once the enemy appears in the new location, ensure the enemy scale is the same as before.*

**Mouse Gestures (Basic Creative Feature - 10%)**

For the player to heal, the player can now draw on the screen with the right click or left click + left control. The player can draw a heart and heal 50 HP. Show the HP healed if successful.

*To test these features, attempt to draw an object with right-click or ctrl + left-click, ensuring that there is a red outline of what is drawn. Once the right-click or ctrl + left-click is released, the drawing should disappear. To test the recognition of a heart, draw a line, circle and ensure that the healing is not triggered. Draw a heart and test if it is registered and the player is healed.*

**Dynamic Shadows (Advanced Creative Feature - 20%)**

All moving character entities now cast dynamic shadows relative to a light source positioned in the center of the room. Using a ray casting method inspired by https://ncase.me/sight-and-light/, these shadows will update every time an entity moves, and stretch as far as the room walls.

*To test, simply move around the room with WASD and note the shadows dynamically projecting off of both the player character and the enemies, always relative to the light source regardless of position.*