General description of game:

The Legend of Zata is a legend of Zelda inspired RPG game with sword-swinging, squid-squishing action.

In this the game the player will spawn onto a small open field and will be immediately tasked with culling the local octorock population. By Using a wooden sword already with the player they will player slice through the octorock’s completing their heroic quest and earning themselves a quest Complete Screen; however should the players health reach zero the player character will disappear and they will be met with a game over screen.

To Help with the immersion ad to that old Zelda feel animations have been added reminiscent of link to past animations. This really adds to the Zelda feel. For the convenience of the player a health display user interface has been put on the top corner of the screen so that the player can monitor their health so as to avoid quest failure.

The Enemies the player has to fight are based on the octorock of legend of Zelda and are called land squid. Upon Game start these land squid will quickly close into the Player and deal damage Overtime whilst they’re in range of the Player. The player will need to move and avoid the land Squid whilst attacking in order to earn a quest completion; However, this is made difficult by the land squids ability to track and home in to the player.

This game also has a Tile Map system with Handcrafted Terrain sprites, this adds to the look and feel of this Zelda Like game.

Mock-up screen

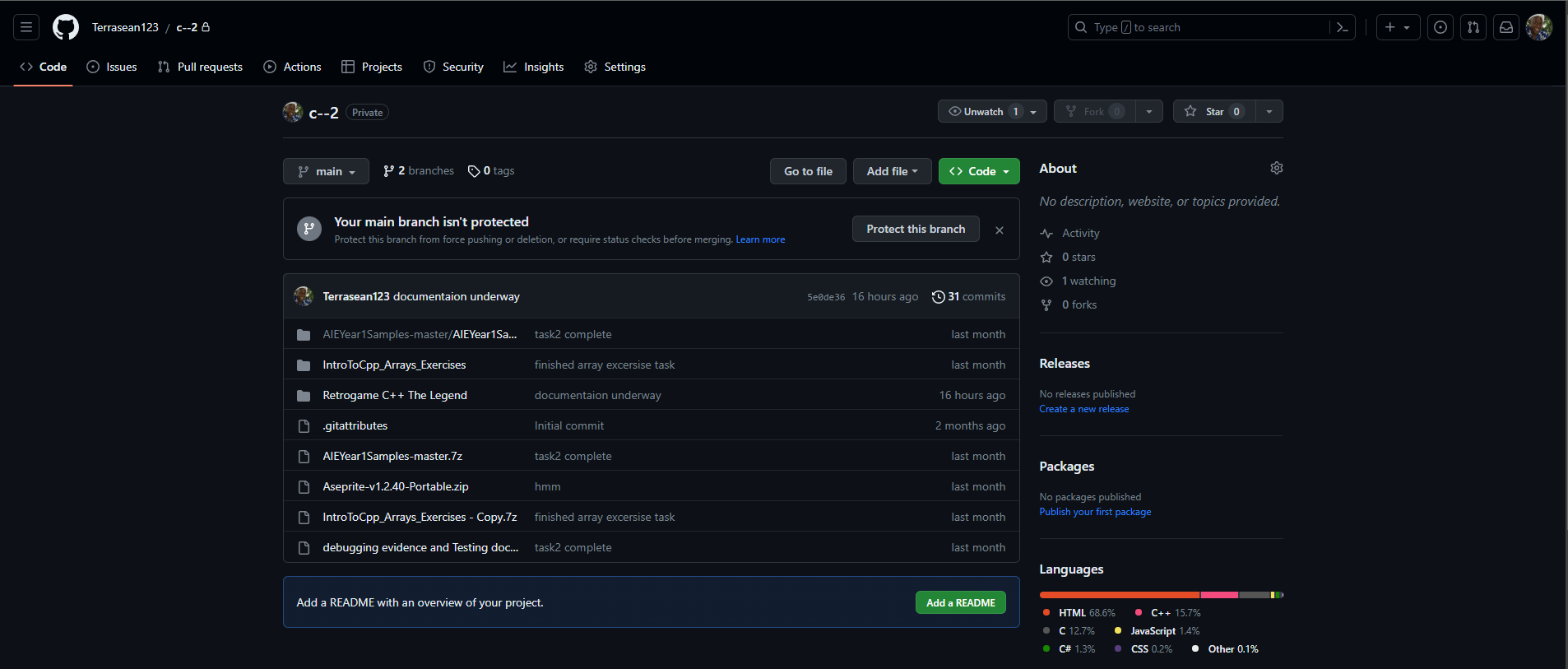


Explanation of data structures used:

The main data structure used within the assignment is a vector containing all Tiles currently being used in Game. The legend of Zata Tile map drawing system relies upon some sort of storage for the tiles. The way this works is by having a separate function increment through the Tiles stored inside and then, utilising the drawing functions from raylib , Drawing them to the screen. This Tile map storage vector is also used to monitor and manage the all tiles on screen.

Another structure that has been used for this Game is a struct known as animation Range. At first Glance this struct may appear fairly simple, it contains 2 integers, an animation start and animation end, However this struct lays the ground work for the sprite Sheet animation techniques the game uses. Evidently these two integers store the beginning and end of animation within the sprite Sheet. By using this struct it was easier to access the animation location of certain animations an allowed efficient usage without the need to calculate the sprite number manually multiple Times.

GitHub Usage:



https://github.com/Terrasean123/c--2