User Guide

Welcome to C(s15)ONTRA, my spin on the beloved run-and-gun style shooter. Below I have highlighted the major features of my game followed by a complete overview of the user interface.

The central theme of my game is to get as far as possible while surviving and eliminating enemy mobs. The game uses polymorphism to add variety to the gameplay. There are different guns that the player can use upon shooting down the power-up drones. Different terrains add challenges to the platforming aspect of the game (I have also implemented the iconic bridge from the original game that falls as the player traverses it). Additionally, three types of mobs are unique in their speed and health.

The Interface

- ## Overview:

  You play as the lead character.
  You have 100 health points at the start of the game.
  You lose health if you get shot by the mobs.
  You spawn with a regular gun at the start of the game.
  There is lava below you. Falling into the lava will lead to your defeat.
  If your health points go below 0, the game will end.

  To get better guns and to get more health, shoot down the power-up drones that spawn When you reach score milestones.

  Different types of mobs spawn over the length of the game. Each mob has a target zone. If you are in the target zone you will be shot at. (Tip: The mobs only shoot you when they are looking in your direction). The mobs patrol the terrain.

  The terrain is also a central part of the game. The different terrains will impact your speed. Quick Sand will slow you down, while frozen terrain will increase your speed. The bridge uses Queues and makes use of the FIFO method when dealing with its elements. As a result, when a player steps on a bridge, the bridge begins to fall (starting with the first block)… INTO THE LAVA. It is up to you to escape the bridge before it completely falls into the lava.

  Two labels display your health and score.

Your Objective?
Defeat as many mobs as possible and survive!
Good luck Soldier!

- ## Player:

Controls:

UP - Jump **Left** - Increases Velocity Towards Left
C - Crouch **Right** - Increases Velocity Towards Right

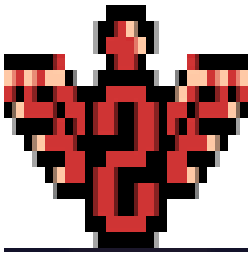
Space - Shoot

W - Move Aim Up **S** - Move Aim Down

Your Health Points: 100 HP

The game ends if the player's HP = 0 or the player falls into the lava

- ## Power-Up Drone:



The Drone appears when you reach certain score milestones.
Shooting the drone rewards the player with a random new gun.
Shooting the drone also adds 5 HP to your Health.
Don't miss them!

- **The Labels:**

The score label shows your score in the game.

The Health label displays your health points.

The game over label appears when you lose the game.

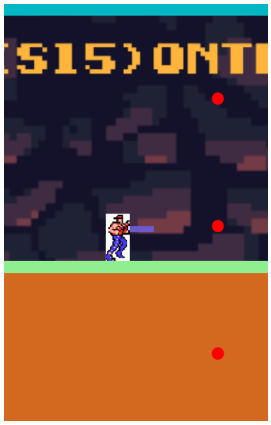
- **The Guns:**

It is here that Polymorphism was super helpful. I used three different types of guns. Regular Gun (called Gun), Cannon Gun, and the Tripple Gun.

Having the Cannon gun and Tripple gun inherit from Gun allowed me to override the Fire method so that each gun behaved differently and added variety to the game. Additionally, I override the isTripleGun() method in Triple Gun to return true. Using these two overrides I allow my Tripple Gun to shoot an array of 3 bullets instead of just one bullet.

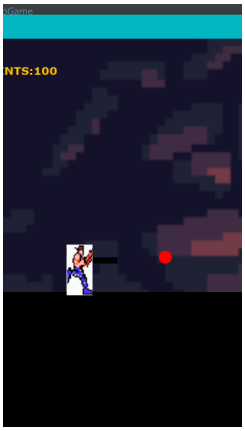
The cannon gun is the strongest of the lot. With its slow and big bullets, it is capable of inflicting high damage upon the mobs.

Triple Gun:



Cannon Gun:

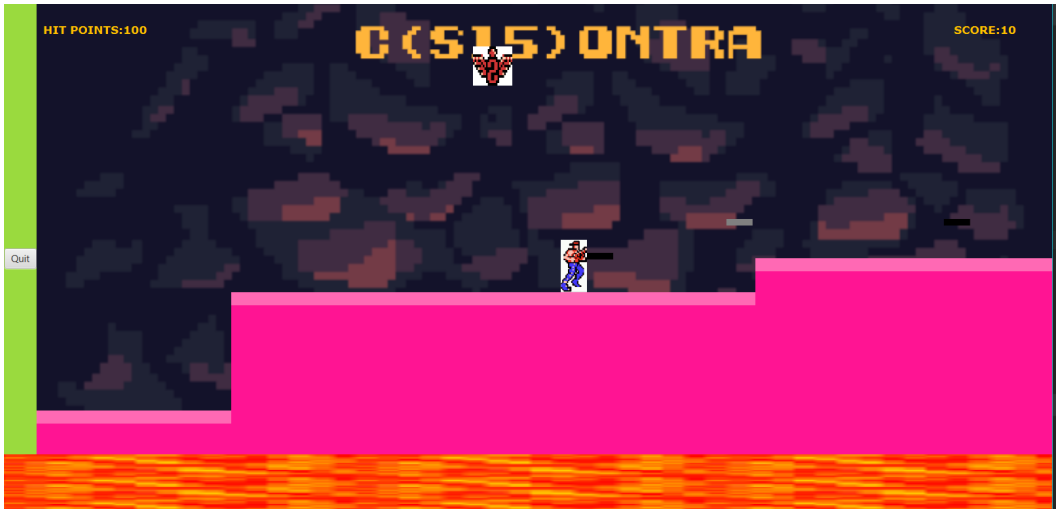
Gun:



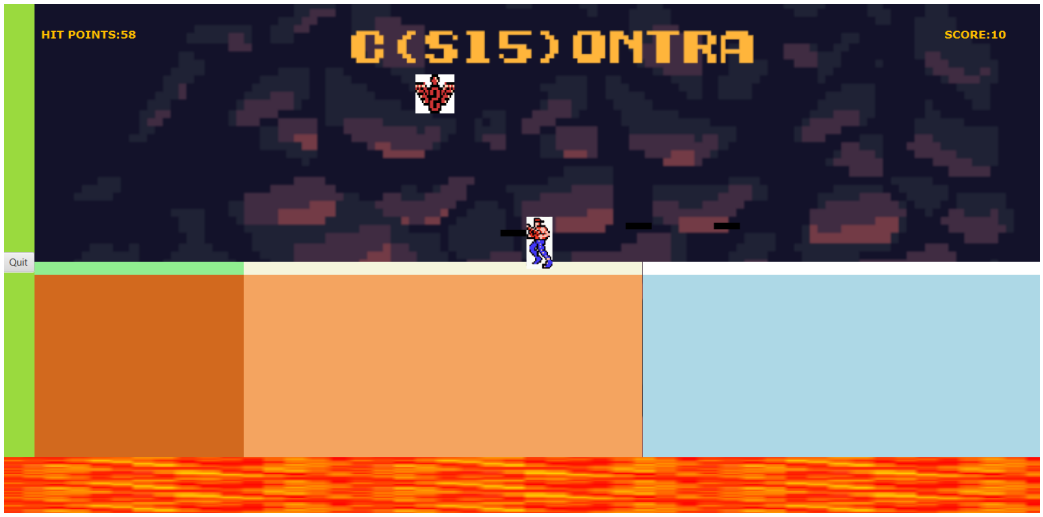
- **Terrain:**

Three different types of terrain can influence the speed of the player. The terrain is an array list of terrain blocks and other subclasses that inherit from it. We also have Bridges which are a Queue of Bridge Blocks, another subclass that inherits from the Terrain Block.

Bridge:



Three Different Terrains:



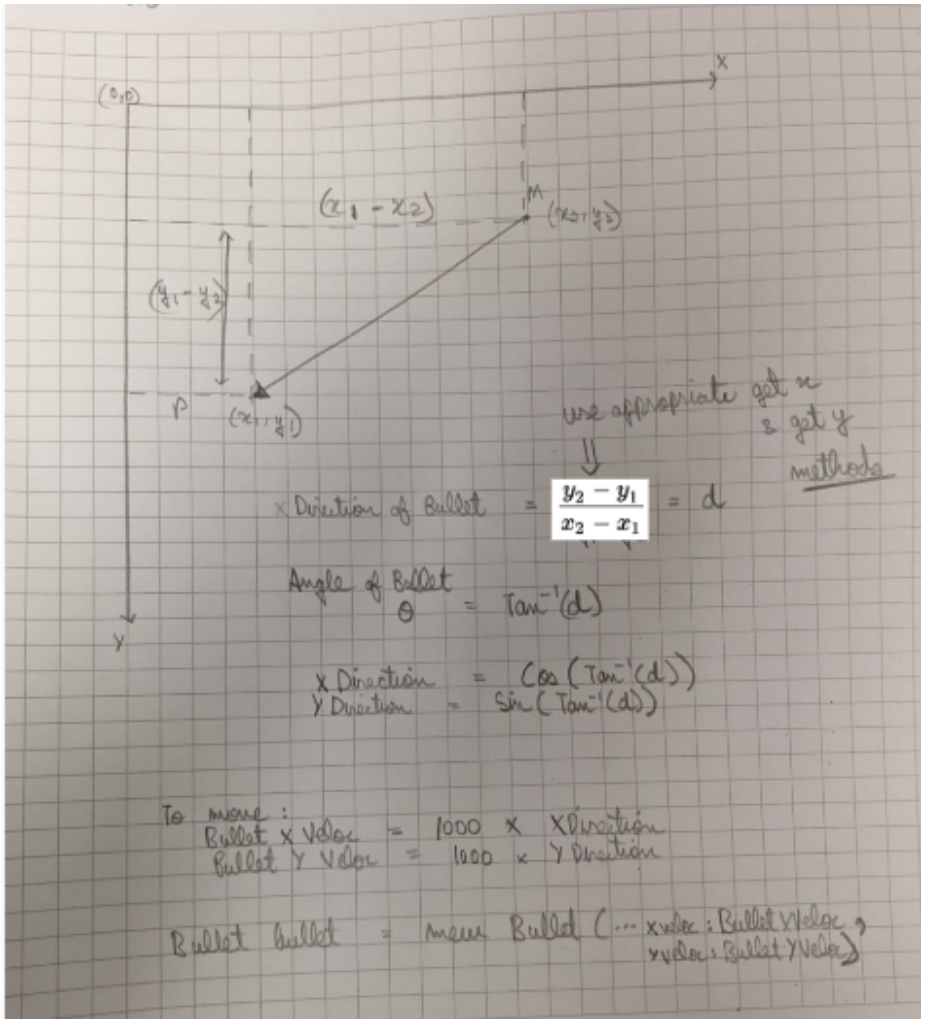
- ## Mobs:

  Mobs are another case where using Polymorphism was beneficial. The two mobs, brute, and Speed inherit from the regular mob which in turn inherits from the player class. There are three different types of mobs.

  Brute Mob: Has a slower speed but maximum hit points.
  Mob: Has regular speed and regular health.
  Speed Mob: Fastest but has the lowest health.

- ## Shooting Algorithm:

My favorite algorithm that I implemented here, is the smart shooting algorithm. In essence, the mobs all have a target circle (that is not added to the pane). Intersecting with this target circle triggers an algorithm that uses trigonometry to compute the player's direction and then passes the appropriate velocity into the bullet.



- ## The Major Classes:

Game: Game functions as my top-level logic class and handles the interactions between the player and the game. Additionally, it also instantiates the terrain generator, the class that is responsible for generating terrain and mobs.

TerrainGenerator: This class handles the “infinite” terrain generation component of the game. I have delegated the creation of the mob to another class, mob generator, which is instantiated in this class. It helps in handling the physical interactions between the player and the terrain.

Mob Generator: This class is responsible for creating the mobs and dealing with the mob’s timelines

- ## Work In Progress:

- The mobs could better interact with the environment.
- They could engage with gravity mechanics and the bridge.
- The Guns should be removed after mobs die.
- More weapons.
- Additionally, platform generation could be more interesting.