Irritating Crusader

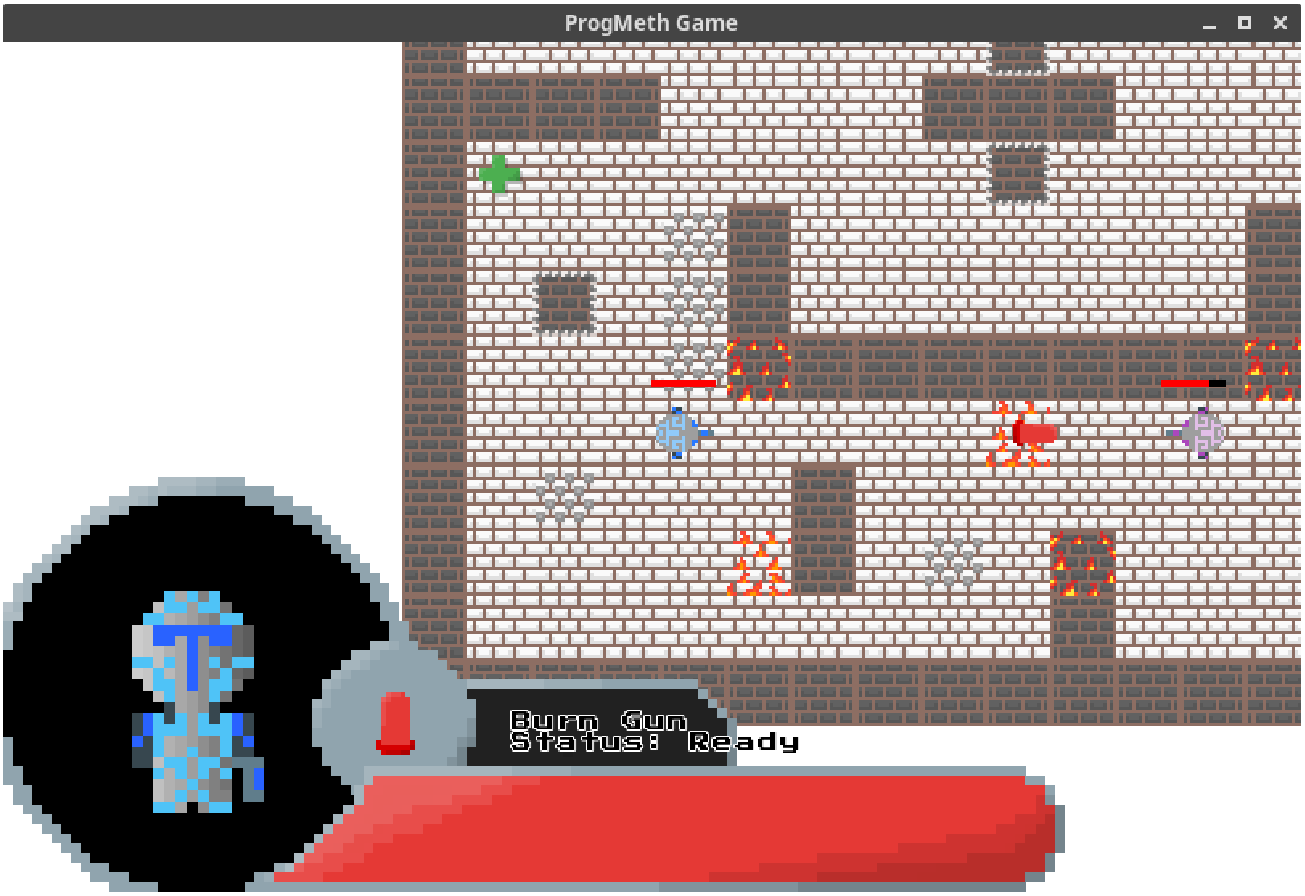
Pipat Saengow (id: 6232024821)   Siri Thammarerkrit (id: 6232029021)

May 17, 2020

Team: ProgMethMyFirstGame

Submitted for
Programming Methodology I (id: 2110215)
Semester II of 2019.
Final project presentation

Contents



1 Overview

Quick Note: Jumps to Game Manual now! if you want to play the game.

Irritating Crusader ("The game") is a two-player, top-down, shooter game where the player "disrupt" the opponent by shooting the "effect bullets", eventually causing the other player to "confuse" and get killed by the "traps".

This report describe the game's gameplay for the players ("Game Manual") and also the game's technical overview for developers ("Technical Description").

2 Game Manual

2.1 Introduction

Irritating Crusader is a simple game where there's only one goal, To kill *all* other player.

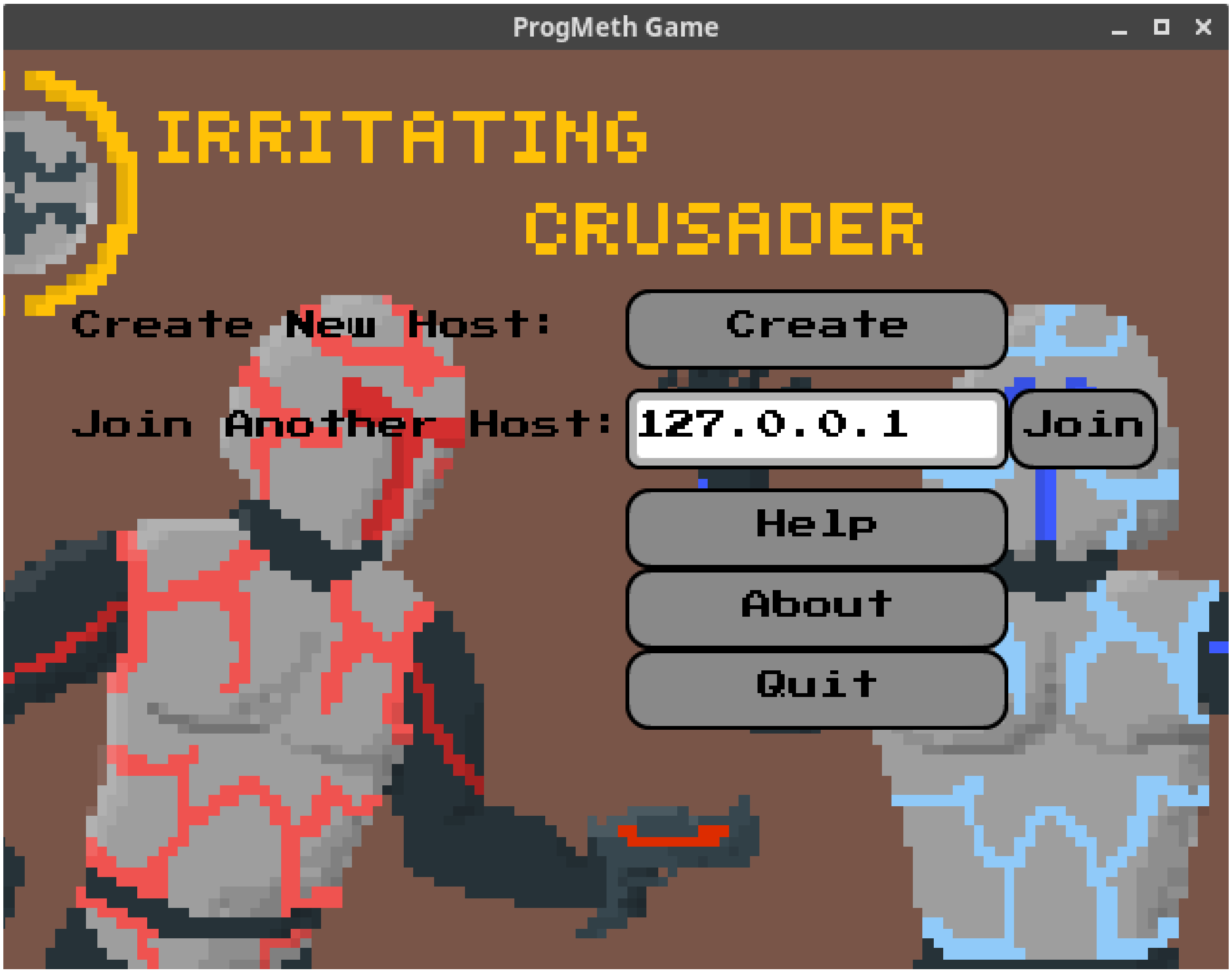
The best way to get introduced to this game is to actually start playing it. Invite your friend (or find one) and follows **Quick Start** guide.

If you played it and somehow got confused, We also provides documentations on bullets, control, etc. below.

2.2 Quick Start

Setting up

1. Grab `IrritatingCrusader.jar`.
2. Install JRE 11 from here
3. If your friends aren't connected to the same local network, Install Logmein Hamachi and follows their guide on how to setup a local network.



Join Another Host: 127.0.0.1

Playing the game

1. Launch `IrritatingCrusader.jar`.

Have one player be the host and

1. Click **Create**
2. Announce his/her IP to every other players.

Have other players who are not the host do

1. Enter the host's IP and press **Join**

Once everyone joined the game.
Press `W`, `A`, `S`, `D`, `Space`, `E` and see what it do.

2.3 Controlling the player

This game is entirely controlled by the keyboard.

Keys	Function
W	Walk north
A	Walk west
S	Walk south
D	Walk east
Space	Shoot
E	Change bullet's type

But don't walk too much though or you would fall into **traps**!

Note: There's one key we didn't mention. The `/` key opens a cheat console. If you somehow opened it, Close it using `ESC` key. Don't cheat!.

2.4 Traps

- **Burning Floor:** Make player who step on this type of floor get **Burn** status effect.
- **Burning Block:** Make player who touch this type of floor get **Burn** status effect.
- **Spike Trap Floor:** It deal little damage to players who step on this type of floor.
- **Spike Trap Block:** It deal little damage to players who touch this type of block.
- **Curing Floor:** It clear player who step on this type of floor their status effects.

So now you know how to avoid them, But wouldn't it be good if you can use them! By shooting **effect bullets** you can cause other player to fall into a trap!.

2.5 Effect Bullets

- **Burn Bullet:** It cause a **Burn** status effect to the target player.
- **Confuse Bullet:** It cause a **Confuse** status effect to the target player.
- **Slow Bullet:** It cause a **Slow** status effect to the target player.
- **Stunt Bullet:** It cause a **Stunt** status effect to the target player.
- **Hook Bullet:** Move target player to user position.
- **Teleport Bullet:** Swap location of user and target player.

2.6 Status Effects

- **Burn:** Gradually deal damage to plaer who affected by this status effect.
- **Confuse:** Make player who affected by this status effect move to the opposite direction of their intention.
- **Slow:** Make player who affected by this status effect move slower.
- **Stunt:** Make player who affected by this status effect unable to move.

2.7 HUD status

- **HP bar:** Display your health point.
- **Gun label:** Display the gun that you are holding.
- **Bullet label:** Display the bullet from your gun.
- **Character frame:** Display your character model.

3 Technical Description

3.1 Introduction

The game's engine is designed to be a general purpose, multiplayer shooter engine.
The game engine is divided into serveral components and implemented using external libraries whenever possible.

The following section describes its components and its implementation detail.

3.2 Components

The most complex functionality in the game's engine is the multiplayer client-server functionality. so the source code's structure is built around it.

The game consist of two major component. The server and the client.

The server package: `com.progmethgame.server`

The server is a component that processes the player's input and simulate the game's law including

- Physic law
- Entity interaction rules
- Entity status reporting.

The server then send the resulting calculation (graphic, sound) through the network

The network package: `com.progmethgame.network`

The networking is a component that tranparently provides the client-server communication.
It provides

- Transparent networking system
- Interface for event source and sink.
- Data serialization

The networking component then transfer server's data to the client

The client package: `com.progmethgame.client`

The client is a component that

- Render graphic.
- Play the sound and music
- Send user's keyboard input.

The server and the client are both initialized by the launcher

The launcher package: `com.progmethgame.launcher`

The launcher is a component that provides GUI for creation and disposal of the client and the server.

3.3 Implementation

The game engine is implemented using various library.

3.3.1 Graphic, Sound and Controller

LibGDX provides graphic, sound and controller library. We choose this over JavaFX because it

- Provides GL acceleration
- Have many useful utils (eg. Vector arithmetic, Asset management, TMX Map loading)
- Comes with boiler plate (eg. Game-loop, Screen management)

3.3.2 Networking

KryoNet provides object serialization and networking system.

4 API Documentation

Every classes are documented using javadoc. please open `javadoc/index.html` with your browser or if you prefer PDF files please see `refman.pdf`

5 Copyright Material

- **LibGDX** is an Apache2-licensed game engine by Bad Logic Games.
- **KryoNet** is a networking library by Nathan Sweet licensed under BSD-3-Clause.
- **PressStart2P** is a font by CodeMan38 licensed under Open Font License.
- **plain-james** is a scene2d ui skin by Raymond “Raeleus” Buckley licensed under CC BY 4.0
- **TMX Format** is a tiled map data format by mapeditor.org licensed under CC BY-SA 3.0.
- **8-bit Game Over** sound effect by Euphrosyyn via freesound.org CC BY 3.0