CSC 413 Tank Game Documentation

Fall 2019

Cody Huang

917041832

CSC 0413-01 Summer 2021

https://github.com/csc413-su21/csc413-tankgame-wenhuang415

Table of Contents

1 Introduction/Project Overview

This project is a tank game. It is two player pvp game. The players have to destroy the others tank in order to win. There are two types of walls, breakable and unbreakable walls. There are 3 types of power ups: health, moveSpeed, and raidFire power ups. Each tank has 3 lives with 10 HP per life. There are traps that will reduce tank HP if touched.

1.1 Technical Overview

This game consists many gameObject classes that are objects inside the game. A gameLoader class loads the map and instantiate every game object including the tanks. Game loader also runs the game state updating each game object and keeping track of what happens. Game load also draws the game state and render split screen and loads background music.

1.2 Summary of Work Completed

Wrote gameObject class to represent a game object with x and y coordinates and an BufferImage

Made Classes that don't move are direct descendants of gameObject(powerUps, Walls)

Made subclasses of powerUps (Speed, FireRate, HP)

Made subclass of walls BreakWall: breakable walls

Made moveable abstract class for all moveable gameObjects (bullet and tank)

Used Tank class from TRE example, refactored for shooting and detecting collisions

Made walls push tank back if they collide

Made powerUps change the Tank stats if collected

Made bullets collide with walls

Made gameLoader class to load and paint the state of game

Created background music function in gameLoader class

Added split screen in paintComponents() in gameLoader class

Added functions to return altered x and y coordinates in Tank class to resolve camera limits

Made Resource class to load all the BufferedImages from resources

2 Development Environment

IntelliJ Ultimate Version: 2021.1.2 Build: 211.7442.40

java version "1.8.0_281"

Java(TM) SE Runtime Environment (build 1.8.0_281-b09)

Java HotSpot(TM) 64-Bit Server VM (build 25.281-b09, mixed mode)

3 How to Build/Import your Project

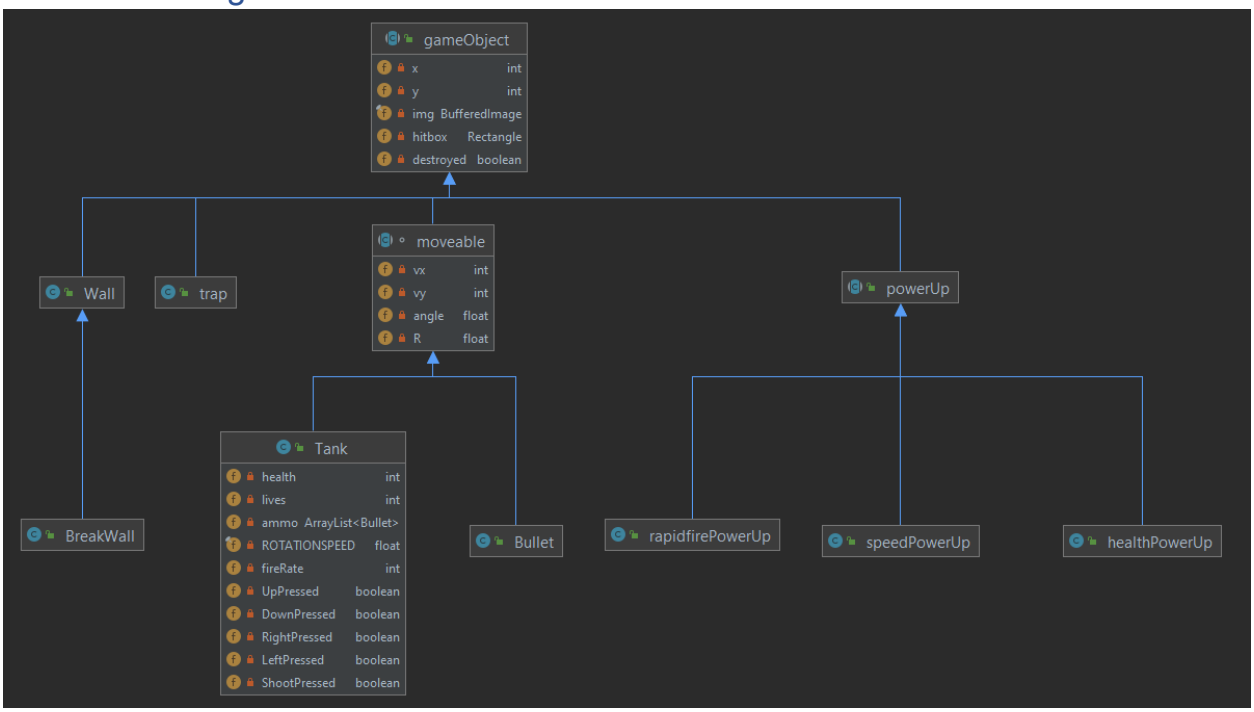
Clone files from https://github.com/csc413-su21/csc413-tankgame-wenhuang415.git. Import project to IDE using /src as main folder. Build using Launcher class with main function.

4 How to Run your Project

Run the jar or build the Launcher class and run that.

5 Implementation

5.1 Class Diagram



5.2 gameObject class

The gameObject class is an abstract class that represents all game objects that can be displayed on the screen. All gameObjects contain x and y coordinates as int, a BufferedImage to draw on screen, a rectangle hitbox to represent the hitbox based on the bufferedImage, and a boolean destroyed to show if the gameObject has been destroyed.

5.3 Wall class

The wall class is a concrete class that extends gameObject. It represents an unbreakable wall in game.

5.4 BreakWall class

The BreakWall call extends Wall class. It represents a breakable wall in game.

5.5 PowerUp class

The powerUp class extends gameObject. It is an abstract class that represents power ups. All powerUp subclasses must implement the setEffects() function that changes a status of a tank.

5.6 HealthPowerUp class

The healthPowerUp class extends powerUp. It sets the health of a tank to 10 and prints out tank health.

5.7 SpeedPowerUp class

The speedPowerUp class extends powerUp. This class increases a tank's movement speed by increasing the speed scale R in instances of tanks. Then prints out the speed value.

5.8 RapidfirePowerUp class

The rapidFirePowerUp class extends powerUp. This class increases the fire rate of tanks. Then prints the fire rate of the tanks

5.9 moveable class

The moveable class is an abstract class that extends gameObject. It represents a game object that can move around the map. It contains int vx, vy representing x and y velocities. It also has a float for the angle to change directions and float R representing the max speed the moveable object can move. It implements moveForwards() and moveBackwards() to allow the moveable object to move. It also has a checkBorder() function to prevent moveable object from moving outside the game world.

5.10 Tank class

The Tank class extends the moveable class. It represents a tank that can rotate left and right, move forwards and backwards and it can shoot bullets. Tank contains an array list of bullets called ammo, int health, and int lives.

5.11 TankControl class

The TankControl class sets up the controls for tanks. It implements KeyListener to listen for key pressed and toggles the control for the tank.

5.12 Bullet class

The Bullet extends moveable since it's the only other gameObject that can be moved. If bullet hits a tank it will subtract HP from the tank. If bullet hits a breakWall then the breakWall will be broken. Bullets are instantiated by the tanks that fire them.

5.13 Trap class

The trap class extends gameObject. It represents a trap. If a tank collides with a trap then the tank's HP will be reduced.

5.14 GameLoader class

The gameLoader class will load all the gameObjects and maintains the game state. It initializes the tanks, powerUps, and walls to create the map. It is also in charge of painting the game state. The function backgroundMusic() will start background music on a new thread.

5.15 Resource class

The Resource class loads all of the bufferedImage required to instantiate every gameObject.

5.16 Launcher class

Launcher class draws the panels for the start and end menu. This class starts the music thread and launches the game.

6 Project Reflection

This project was fun. It was my first big programming project. The hardest part was writing code that was open for extension and closed for modification. I think this project gave me a lot of experience writing code that way. I had to do a lot of refactoring of the code in order to reduce the amount of duplicate code.

7 Project Conclusion/Results

This game was pretty simple. There were things that I wanted to add but couldn't figure out how to, I.e bouncing bullets, and explosive bullets. There were somethings I wanted to add but did not have enough time. Such as different type of tanks. I know how to add different types of tanks but I did not have enough time to figure out how the player would choose the tank types in the menus. One thing I wish I had more time on was to explore the menu options. Right now I am using the menus from the TRE example. Overall I think this project was a fun one.