San Francisco State University
School of Science and Engineering
Spring 2018

Term Project
Tank Wars
Koala-Br8

ENGR 413 Section 3
Instructor: Anthony Souza
Date Submitted: May 25, 2018

Team 26
Dennis Lin & Jessica Yip

GitHub Repositories:
https://github.com/csc413-03-sp18/csc413-tankgame-Team26
https://github.com/csc413-03-sp18/csc413-secondgame-Team26

Project Introduction and Overview

This project is two different games, Tank Wars and Koala-Br8. This document will explain how the games were developed, and guide those who wish to implement or play the games.

Tank Wars

This is a simple tank game that we will reuse some of its classes in our second game. Essentially it will be a 2-player game with a mini map. Some of the walls will be breakable from the bullets of the tank. Each player will have independent movement as well as their own health. There will be power ups located on the map to improve health as well as change the bullet that they shoot.

Koala-Br8

The evil doctor has captured a colony of koala bears for his horrible experiments. The koalas were able to escape but the evil doctor has placed a mind controlling device on them. Only with the koala's combined efforts will they be able to make their way past the many dangers of the doctor's lab and into freedom.

The arrow keys will move all the bears simultaneously, because all of them move at once, some paths or even other koala bears may block the koalas. Each level will have some danger in the maze but the goal is to move all of the koalas to the exit. If a koala touches a hazardous maze, that is in the way, it dies and the level must be restarted.

Development Environment

This assignment was completed using NetBeans 8.2 IDE with Java 8 on a Windows 10 operating system. All game resources such as images were provided by the instructor.

Assumptions

This document and program assumes the following:

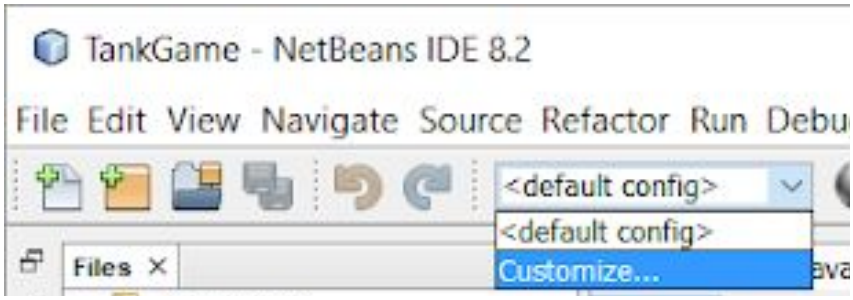
- The GitHub repositories have been cloned.
- NetBeans IDE is being used.
- The computer being used has Java 8 or later installed.
- Game resources were provided by the instructor.

Running and Playing the Games

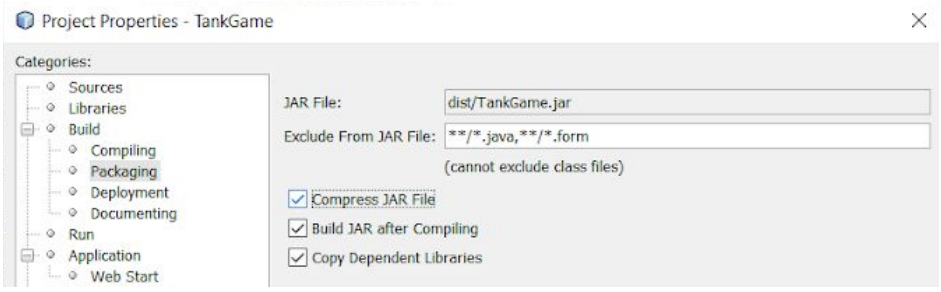
Tank Wars

In order to run the Tank Wars game, the user does not need to import the program into NetBeans or an IDE. Simply execute the executable TankGame.jar file by double clicking it in File Explorer.

The user can open the Tank War game program into NetBeans by creating a new project by going to *File >> New Project* and selecting *Java Project with Existing Sources*. After naming and choosing a directory for the new project, the following screen will ask for existing source files. Click *Add Folder* and navigate to the cloned repository file and click *Open* to add the source files to the new project. Finally, in order to create the executable JAR file in NetBeans, the user must customize the project properties by clicking on the dropdown menu in the toolbar that says <*default config*> and selecting *Customize*.



The *Project Properties* window will then pop up. Navigate under *Build >> Packaging* and mark the *Build JAR after Compiling* option. The user may also specify where to save this JAR file in order to find it easily.



After compiling the project, navigate to where the JAR file was saved and execute the JAR file to run and play the game.

Playing Tank Wars

The objective of the game is to eliminate the other player's tank by shooting them with bullets until they are out of lives. The tanks are placed on a map containing breakable and unbreakable walls, as well as power-ups. Bullets can be shot at breakable walls to clear paths, and rolling over a power-up gives that tank that power-up.

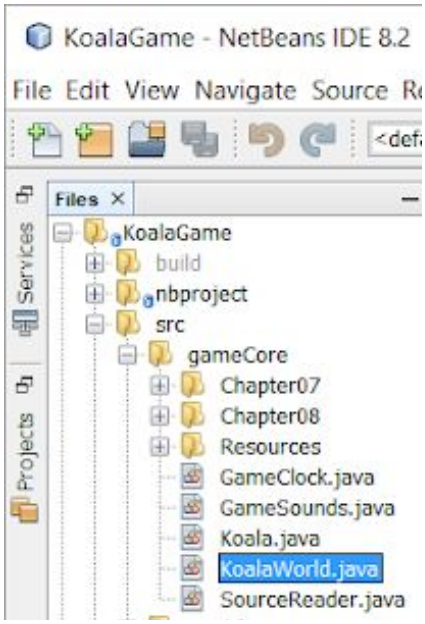
Tank 1 is controlled using the arrow keys and the "Enter" key to shoot. Tank 2 is controlled using the "WASD" keys and the spacebar to shoot.

There are two types of power-ups. One grants the player an extra life, and the other allows the tank to shoot three bullets at once instead of one. All power-ups use the same image, in order to make the game more fair.

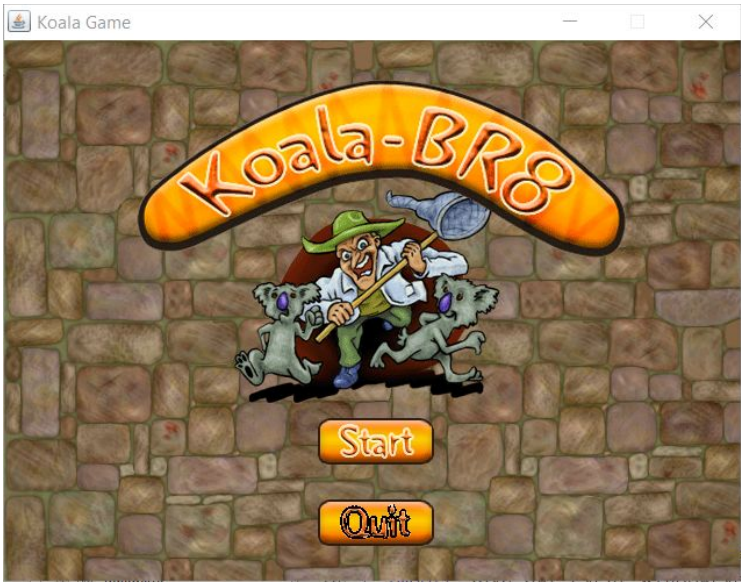
Koala-Br8

In order to open the Koala-Br8 game in NetBeans, go to *File >> Open Project*, and navigate to the directory where the cloned repository is located and click on the *csc413-secondgame-Team26* folder with the small coffee cup picture next to it. Click *Open Project* and NetBeans will open the project.

In the navigation sidebar on the left side of the IDE, open the *Files* window and navigate into the *src >> gameCore* files. Double click on KoalaWorld.java to open the file.



Run this file by going to *Run >> Run Project* or by pressing F6 in order to run and play the game.

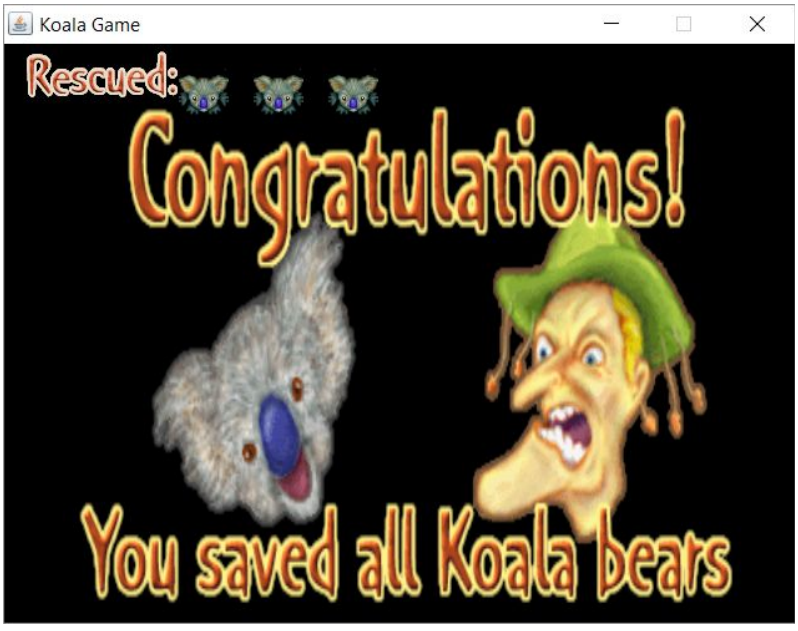


Playing Koala-Br8

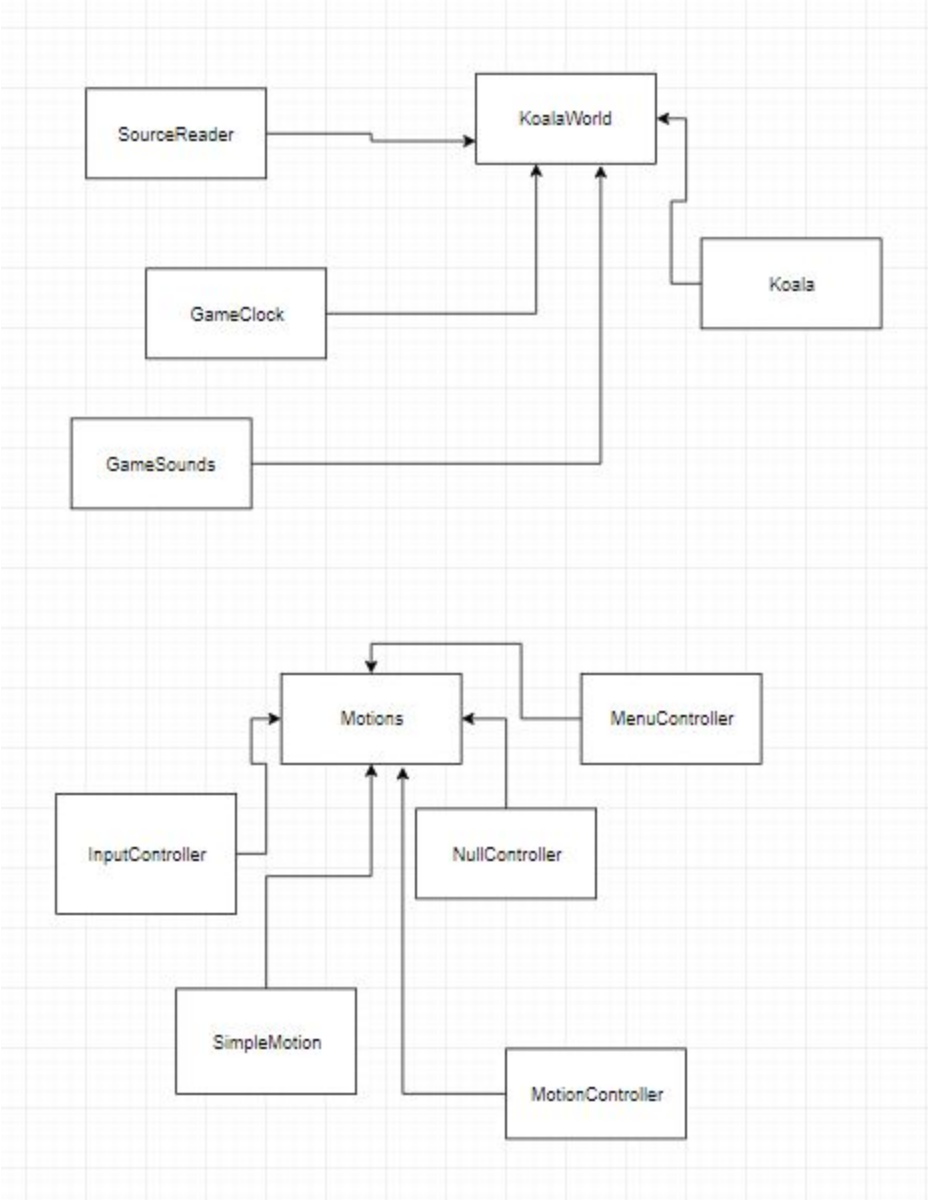
Once the above menu screen loads, press Enter to start the game. Use the arrow keys to control all the koalas at once and guide them to the exits.

There are different types of hazards throughout the map. The TNT and circular saws are fatal. The red switches can be hit to cause the TNT to explode and clear a path for the koalas to the exits.

Win the game by successfully guiding all koalas to an exit. Lose the game by killing 3 koalas.



Koala-Br8 Game Class Diagram



Shared Class Descriptions

SourceReader.java

This class reads in a source file for creating the map of each game. It creates the appropriate object at each location corresponding to the file such as a destructible or an indestructible wall.

GameSounds.java

This class reads in the appropriate audio files for the different sounds of each game.

MotionController.java
This is an abstract class that controls the movement of objects. It's able to read in the object to move and update its location accordingly.

InputController.java
This class extends MotionController and sets the controls for each game. It also uses KeyListener to set the keyboard controls.

NullMotion.java
This class is a MotionController that does nothing and is called when a motion is triggered, but no actual movement is needed.

BackgroundObject.java
This class creates the objects that appear in the background of the game. They are set at a stationary speed and the appropriate image is loaded.

Background.java
Background.java extends BackgroundObject and draws the background image. This class also draws the background as it scrolls when the tanks move around the map.

BigExplosion.java
This class is responsible for creating the explosion animation and playing the explosion sound when a tank or koala dies.

IndestructibleWall.java
This class creates the objects that are indestructible, mainly used to show the boundaries and platforms the player go be restricted on.

MoveableObject.java
This class is responsible for all moveable objects that have special behaviors.

SmallExplosion.java
This class creates the smaller explosion animation and playing the explosion sound when tank bullets hit an indestructible wall or another tank, or when TNT explodes in Koala-Br8.

InfoBar.java
This class creates the information bar at the bottom and tops of each game that display information such as lives and health.

Tank Game Class Descriptions

Tank.java

This class creates the Tank object that each player controls. It initializes the tank and sets the lives, health, initial score, strength, enables controls, and includes methods for respawn and death.

TankWorld.java

This class creates and initializes the whole map of the Tank Wars game. It creates the ArrayLists of each game item such as bullets, power-ups, and players. It also loads all the appropriate images.

IndestructibleWall.java

This class extends BackgroundObject to create and draw the indestructible walls on the map. They are set as BackgroundObject since they do not move nor cause damage.

DestructibleWall.java

This class draws the destructible walls on the map. It gives each wall a health value so it can take damage and disappear when hit by a bullet.

Bullet.java

This class extends MoveableObject and creates each Bullet object that is shot out of a tank. It keeps track of which tank the bullet came from in order to assess damage.

Koala-Br8 Game Class Descriptions

Rock.java

This class draws a rock object as a BackgroundObject and sets it to not deal damage.

Saw.java

This class creates and draws the saw objects. It sets them to move back and forth on a straight path and to deal damage when it collides with a koala. The saw does not disappear once a koala hits it.

TNT.java

This class draws the TNT objects. The object is set to deal damage when it collides with a koala, and disappear after. Each TNT object is also linked to a Detonator object.

Detonator.java

This class draws the detonator objects and links it to a TNT object. After colliding with a koala the connected TNT is set to explode.

Exit.java
This class checks for the event which is triggered by the user to exit the game.

Koala.java
This class is responsible for the koala's movement from the user input. It creates the koala objects, draws them onto the map, and sets their initial values such as health, lives, and respawn.

KoalaWorld.java
This class creates and initializes the whole map of the Koala-Br8 game. It creates the ArrayLists of each game item such as rocks, TNTs, detonators, saws, and exits. It also loads all the appropriate images.

Team Collaboration

Our term project was developed in person due to having very similar schedules that allowed us to do so. We worked together to develop a rough class diagram by hand before taking turns coding while the other stood by to do research and offer suggestions. The project was completed in a pair programming manner.

Conclusion

The goal of this project was achieved. From our time in class we were focused on the reusability of certain classes in order to make programming another game more efficiently. Our first run with the tank game was partly a success. We had some difficulty with the walls where the tank would fit through spaces that it normally should not fit through. We believe that it was how we related the reference point to the center of the tank that it does not do the right comparison when the tank collides with a wall. We reused some of the classes from the tank game for the koala game and replaced the images. The classes were a huge help in designing the Koala game and made everything flow and come together smoothly. We successfully completed the koala game and we are happy with its completion.