Tanky Shooter Documentation

Created by
Tanaphan Krupistrai 6631323621
Thukdanai Thaothawin 6631319121

2110215 Programming Methodology
Semester 2 Year 2023
Chulalongkorn University

Tanky shooter

Introduction

Tanky shooter is inspired by the 2-player tank shooter game “Battle City”. The objective of the game is to be the last player standing in order to win.

Rules

This is the control scheme for each player.

	Move	**Shoot**
Player 1	WASD keys	Space Bar
Player 2	Arrow keys	L

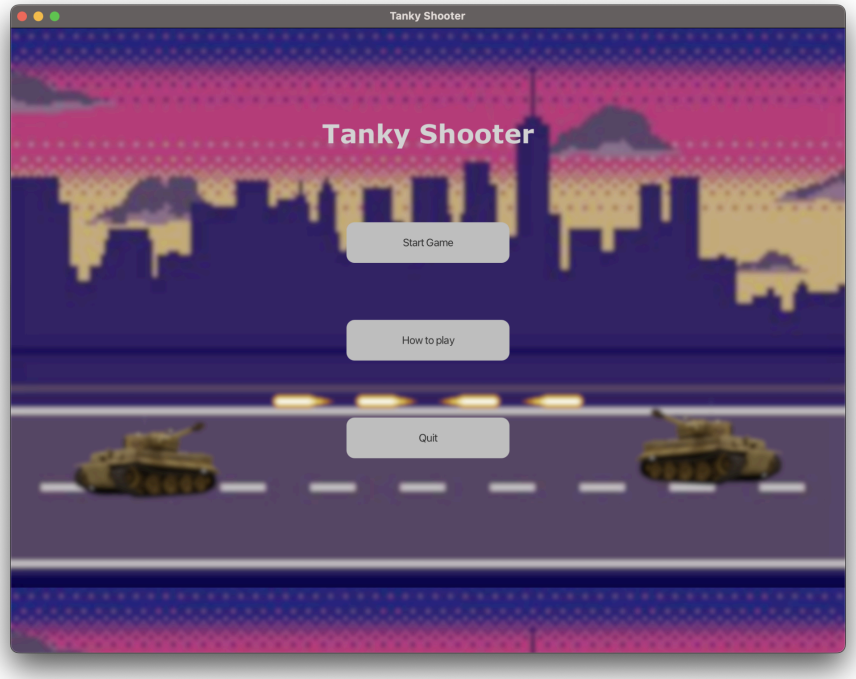
Once the game starts each player will start with 100 HP each bullet does 10 damage and bullets cannot travel through walls.

As long the game progresses, there will be power ups spawned. There are total of 4 types of power ups

	Speed Boost - Increase player movement speed for 8 seconds		Damage Boost - Increase player's bullet's damage and speed for 10 seconds
	Invincibility - Become immune to damage for 5 seconds		Healing - Heal the collected player for 50 HP

The game will continuing until a player's HP reaches 0

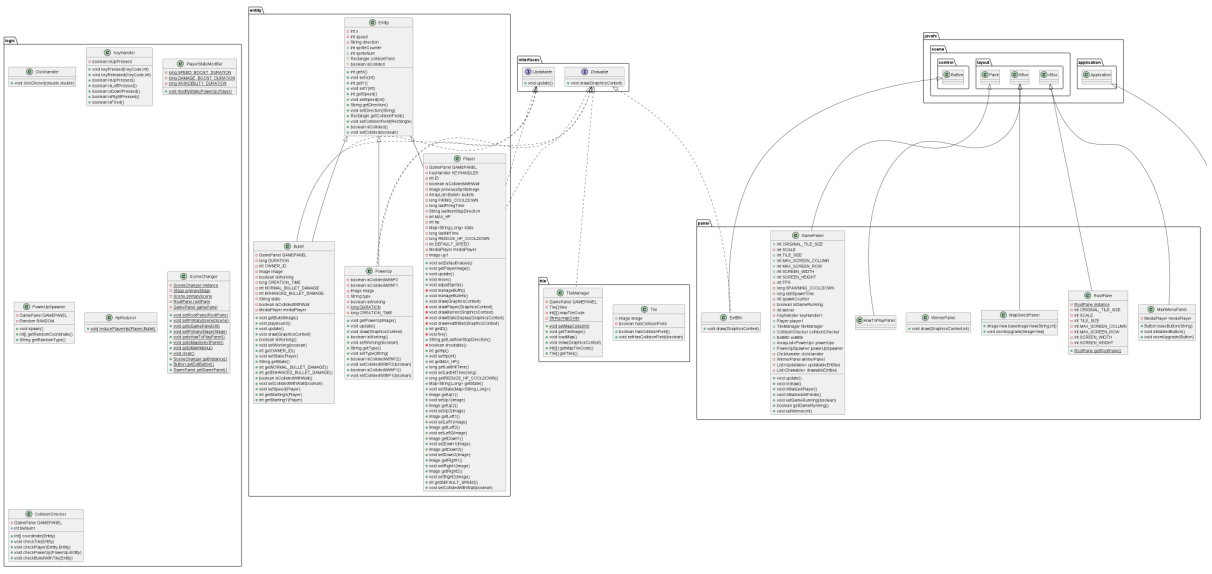
MainMenu scene



Click the “How To Play” button to see the tutorial.

Click “Start Game” to select the desired map and then click on the desired map to play the game.

Class Diagram



1.Package application

1.1 class Main extends Application

1.1.1 Methods

+ void start(Stage primaryStage)	Set Scene and Stage as RootPane's instance
+ main(String[] args)	Launch application

2.Package entity

2.1 Class Bullet extends Entity implements Drawable, Updatable

This class represents bullet entities which extends from Entity class

2.1.1 Fields

- GamePanel GAMEPANEL	GamePanel for size
- long DURATION	Duration of bullet in gamepanel
- int OWNER_ID	Owner Id (1 / 2)
- Image image	Image for display
- boolean isWorking	Boolean uses to check if bullet's still working
- long CREATION_TIME	Time the bullet was constructed (3 seconds)

- int NORMAL_BULLET_DAMAGE	Normal bullet's damage (10)
- int ENHANCED_BULLET_DAMAGE	Enhanced bullet's damage (15)
- String state	Type of Bullet (Normal / Enhanced)
- boolean isCollidedWithWall	Collision stage with wall
- MediaPlayer mediaplayer	Mediaplayer for sound

2.1.2 Constructor

+ Bullet(GamePanel GAMEPANEL, Player player)	Plat firing sound FX Set GamePanel Set bullet's properties (OWNER_ID, direction, collisionField, state, speed) Set bullet's x and y by using returned values from getStartingX() and getStartingY() Set isWorking to True Set isCollidedWithWall to false Record CREATION_TIME Set bullet Image

	corresponding to direction and state Call getBulletImage()

2.1.3 Methods

+ void getBulletImage()	Load bullet's image based on its direction and state
+ void update	Update bullet's position according to its direction Then set isWorking to false if expired Call checkBulletWithTile from CollisionChecker
+ void draw(GraphicsContext graphicsContext)	Draw bullet's GraphicsContext
+ int getStartingX(Player player)	Get starting X coordinate based on player's X coordinate and direction
+ int getStartingY(Player player)	Get starting Y coordinate based on player's Y coordinate and direction
+ void playSound()	Play sound when bullet is generate
Getters and Setters of all fields	

2.2 Class Entity

This class acts as a parent class for all entities.

2.2.1 Field

- int x, y	Entity's position
- int speed	Entity's speed
- String direction	Entity's direction
+ int spriteCounter	Use to create player moving animation
+ int spriteNum	Use to create player moving animation
# Rectangle collisionField	Entity's hitbox
# boolean isCollided	Boolean use to check collision with other entities

2.2.2 Methods

Getters and setters of all fields	

2.3 Class Player extends Entity implements Updatable, Drawable

This class represents both players which extends from Entity class

2.3.1 Fields

- GamePanel GAMEPANEL	Gamepanel for all method
- KeyHandler KEYHANDLER	KeyHandler Class for handling key press
- int ID	Player’s ID (1 / 2)
- boolean inCollidedWithWall	Use to check if player has collided with wall
- Image previousSpriteImage	Use to create player moving animation
- ArrayList<Bullet> bullets	ArrayList contained bullets fired by each player
- long FIRING_COOLDOWN	Cooldown for firing bullets (0.5 seconds)
- long lastFiringTime	Time at the last firing time
- String lastNonStopDirection	Last direction which is not “stop”
- int MAX_HP	Player’s max HP (100)
- int hp	Player’s current HP
- Map<String, Long> state	Player’s state with buff’s name as key, expiring time as value
- long lastHitTime	Latest time when player being hit
- long REDUCE_HP_COOLDOWN	Cooldown for reducing player’s HP (0.25 seconds)
- int DEFAULT_SPEED	Player’s default speed (3)
- Mediaplayer mediaplayer	Mediaplayer for hurt sound

- Image up1, up2, left1, left2, down1, down2, right1, right2	Sprite images for each direction

2.3.2 Constructor

+ Player(GamePanel GAMEPANEL, keyHandler KEYHANDLER, int ID)	Set GamePanel and KeyHandler Set player's properties (ID, HP = 100, collisionField) Call setDefaultValues() and getPlayerImage()

2.3.3 Methods

+ void setDefaultValues()	Set player's position according to player's ID {Player 1: X = 200, Y = 150 Player 2: X = 700, Y= 150} Set player's speed to DEFAULT_SPEED Set player's direction to empty String
+ void getPlayerImage()	Load images for each direction
+ void update()	Update player's direction based on key pressed Call move()

	Call adjustSprite() Call fire() bullet if specific key has been pressed and not in cooldown Check if player has collided with other entities Call update() on each bullet fired by this player Call manageBullets() and manageBuffs()
+ void draw()	Draw player's GraphicsContext and call draw() for each bullets Display player's status
+ void fire()	Initialize new Bullet with this player's properties and re-record lastFiringTime
- boolean shootable()	Return boolean indicates that the player is shootable or not
+ void move()	Update player's position and firing according to key pressed or released Check if player has collided with other entities
+ void adjustSprite()	Adjust spriteNum and spriteCounter to make player's animation

+ void manageBullets()	Call update() on each bullet fired by this player, then call HpReducer.reducePlayerHp if needed Set isWorking to false if bullet has collided Remove expired and non-functioning bullets
+ void manageBuffs	Remove expired PowerUp's effect (Set player's speed back to DEFAULT_SPEED)
+ void drawPlayer(GraphicsContext graphicsContext)	Draw player itself from GraphicsContext
+ void drawBarrier(GraphicsContext graphicsContext)	Draw player's barrier if player has invincible buff from GraphicsContext
+ void drawHealthBar(GraphicsContext graphicsContext)	Draw player's health bar from GraphicsContext
+ void drawStateDisplay(GraphicsContext graphicsContext)	Draw player's state display from GraphicsContext
Getters and setters of all fields	

2.4 Class PowerUp extends Entity implements Updatable, Drawable

This class represents power up entities which extends from Entity class

2.4.1 Fields

- boolean isCollidedWithP1	Boolean used to indicates if the PowerUp is collided with Player 1
- boolean isCollidedWithP2	Boolean used to indicates if the PowerUp is collided with Player 2
- Image image	PowerUp's image
- String type	PowerUp's type
- boolean isWorking	Boolean used to check if PowerUp's still working
- long DURATION	PowerUp's duration (10 seconds)
- long CREATION_TIME	PowerUp's time of creation

2.4.2 Constructor

+ PowerUp(GamePanel gamepanel, int x, int y, String type)	Initialize new PowerUpSpawner with GamePanel Set PowerUp's properties (collisionField, X, Y, CREATION_TIME, isWorking,

	and type) Call getPowerUpImage()

2.4.3 Methods

+ void getPowerUpImage()	Load PowerUp's image corresponding to its type
+ void update()	If PowerUp has expired, set isWorking to false
+ draw(Graphicscontext graphicscontext)	Draw PowerUp's GraphicsContext
Getters and setters of all fields	

3.Package logic

3.1 Class ClickHandler

This class handles clicking the exit button while in game

3.1.1 Methods

+ void clickClicked(double x, double y)	Handle exit pressed in gamepanel

3.2 Class CollisionChecker

This class handles collision checker between various entities

3.2.1 Field

- GamePanel GAMEPANEL	GamePanel for accessing various fields
-int tileNum1,tileNum2	Int for Tile check

3.2.2 Constructor

+ CollisionChecker(Gamepanel GAMEPANEL)	Set GamePanel

3.2.3 Methods

+ int[] coordinate(Entity entity)	Provide int[] for entity position
+ void checkTile(Entity entity)	Check if the player has collided with tiles that have collisionField or not.
+ void checkPlayer(Entity e1, Entity e2)	Check if both players have collided with each other or not
+ void checkPowerUp(PowerUp powerUp, Entity entity)	Check if the PowerUp has collided with the entity or not. Then play drinking potion SFX
+ void	Check if the bullet has

checkBulletWithTile(Entity entity)	collided with a wall or not.

3.3 Class HpReducer

This class handles reducing player's HP after has been hit with bullets

3.3.1 Method

+ void reducePlayerHP(Player player, Bullet bullet)	Reduce player's HP by considering player's state and bullet's state Play Hit SFX If any player HP reaches 0, setGameRunning to false and setWinner to another player's ID

3.4 Class KeyHandler

This class handles getting inputs from keyboards and translating

3.4.1 Fields

- boolean isUpPressed, isLeftPressed, isDownPressed,	Boolean for any pressed and released keys

isRightPressed, isFired	

3.4.2 Methods

+ void keyPressed(KeyCode e, int playerId)	Check and set direction corresponding to keys pressed
+ void keyReleased(Keycode e, int playerId)	Check and set direction corresponding to keys released
Getters of all fields	

3.5 Class PlayerStateModifier

This class handles modifying player's state according to power ups collected

3.5.1 Fields

- long SPEED_BOOST_DURATION	Duration of speed boost
- long DAMAGE_BOOST_DURATION	Duration of damage boost
- long INVINCIBILITY_DURATION	Duration of invincibility buff

3.5.2 Method

+ void modifyState(PowerUp	Modify or extend player's

powerUp, Player player)	state according to PowerUp type and player's state

3.6 Class PowerUpSpawner

This class handles randoming power ups' properties and spawning and initializing them

3.6.1 Fields

- GamePanel GAMEPANEL	GamePanel for accessing various fields
- Random RANDOM	For randoming number

3.6.2 Constructor

+ PowerUpSpawner(Gamepanel GAMEPANEL)	Set GamePanel

3.6.3 Methods

+ void spawn()	Call getRandomCoordinate() and getRandomType() Then initialize new PowerUp with returned values
+ int[] getRandomCoordinate()	Get random coordinate with does not have collision field
+ String getRandomType()	Get random type of PowerUp

3.7 Class SceneChanger

This class handles managing and injecting different scenes

3.7.1 Fields

- SceneChanger instance	SceneChanger's instance
- Stage primaryStage	RootPane's Stage
- Scene priamryScene	RootPane's Scene
- RootPane rootPane	RootPane of All Panel except GamePanel
- GamePanel gamePanel	GamePanel's instance

3.7.2 Methods

+ void gotoGamePanel(int mapCode)	Go to GamePanel according to mapCode
+ void gotoHowToPlayPanel()	Go to HowToPlayPanel
+ void gotoMapSelectPanel()	Go to MapSelectPanel
+ void gotoMainMenu()	Go to MainMenuPanel
+ void clear()	Clear all RootPane's children except Game text
+ Button getExitButton()	Initialize exit button to go to MainMenuPanel
Getters and setters of most	

fields	

4.Package panel

4.1 Class ExitBtn extends Button implements Drawable

4.1.1 Method

+ void draw(GraphicsContext graphicsContext)	Draw exit button's GraphicsContext

4.2 Class GamePanel extends Pane

4.2.1 Fields

+ int ORIGINAL_TILE_SIZE	Original tile's size
- int SCALE	For tile's size scaling
+ int TILE_SIZE	ORIGINAL_TILE_SIZE * SCALE
+ int MAX_SCREEN_COLUMN	Maximum number of screen's column (16 columns)
+ int MAX_SCREEN_ROW	Maximum number of screen's row (12 rows)
+ int SCREEN_WIDTH	MAX_SCREEN_COLUMN * TILE_SIZE
+ int SCREEN_HEIGHT	MAX_SCREEN_ROW * TILE_SIZE
+ int FPS	Time between each game

	loop
- long SPAWNING_COOLDOWN	Cooldown for spawning PowerUp
- long lastSpawnTime	Time when latest Powerup was spawned
- int spawnCounter	Numbers of PowerUp spawned
- boolean isGameRunning	Boolean that indicates if the game is over or not
- int winner	ID of winner
- KeyHandler keyhandler1, keyhandler2	KeyHandler assigned for each player
+ Player player1, player2	Each player’s instances
+ TileManager tileManager	TileManger for map drawing
+ CollisionChecker collisionChecker	CollisionChecker for collision checking
+ ExitBtn exitBtn	ExitButton to go back to MainMenuPanel
+ ArrayList<PowerUp> powerUps	ArrayList that contained all PowerUps
+ PowerUpSpawner powerUpSpawner	PowerUpSpawner for spawning PowerUps
- ClickHandler clickHandler	KeyHandler for restarting or exiting
- WinnerPanel winnerPanel	WinnerPanel to be drawn when the game ends
- List<Updatable> updatableEntities	List contained entities that implemented Updatable

	interface
- List<Drawable> drawableEntities	List contained entities that implemented Drawable interface

4.2.2 Constructor

+ GamePanel()	Set preferred size Set Background call initializedAllFields() Create and start game loop by using AnimationTimer

4.2.3 Methods

+ void update()	Spawn new PowerUp if lastSpawnTime has exceed SPAWNING_COOLDOWN Call update() on updatableEntities Call update() on each PowerUp and modifyPlayerState if collided, then delete expired ones
+ void draw(GraphicsContext graphicsContext)	Call draw() on drawableEntities Call draw() on PowerUps and

	check if any player has won
+void initializedPlayer()	Initialize new keyHandlers both both players Initialize new clickHandler Set onKeyPressed and onKeyReleased to both keyHandlers Set onMouseClicked to clickHandler Initialize new 2 players and assigned ID as 1 and 2 respectively
+void initializedAllFields()	Call initialized() Initialize new tileManager, collsionChecker, powerUpSpawner, exitBtn and winnerPanel Add both players to updatableEntities Add tileManager, both players, exitBtn to drawableEntities
Getters and setters of some fields	

4.3 Class HowToPlayPanel extends HBox

4.3.1 Constructor

+ HowToPlayPanel()	Initialize Pane for displaying tutorial Set Alignment, Spacing and Padding Add an exit button to this.

4.4 Class MainMenuPanel extends VBox

4.4.1 Fields

- Mediaplayer mediaPlayer	Media Player for background song

4.4.2 Constructor

+ MainMenuPanel()	Initialize Pane for Main menu Set Background Set Alignment Set Spacing Set Padding Provide 3 Button for - Start game - How to play - Exit

	Play background music

4.4.3 Methods

public Button baseButton(String text)	Preset of button
public void initializedButton()	Setup everybutton
public void zoomUpgrade(Button button)	Make the button interactive

4.5 Class MapSelectPanel extends HBox

4.5.1 Constructor

+ MapSelectPanel()	Initialize Pane for selecting maps

4.5.2 Methods

public ImageView baseImageView(String num,int mapcode)	Provide Button for same pattern
public void zoomUpgrade(ImageView imageView)	Make Imageview interactive

4.6 Class RootPane extends VBox

4.6.1 Fields

- RootPane instance	RootPane's instance
- int ORIGINAL_TILE_SIZE	Original tile's size
- int SCALE	For tile's size scaling
- int TILE_SIZE	ORIGINAL_TILE_SIZE * SCALE
- int MAX_SCREEN_COLUMN	Maximum number of screen's column (16 columns)
- int MAX_SCREEN_ROW	Maximum number of screen's row (12 rows)
- int SCREEN_WIDTH	MAX_SCREEN_COLUMN * TILE_SIZE
- int SCREEN_HEIGHT	MAX_SCREEN_ROW * TILE_SIZE

4.6.2 Constructor

+ RootPane()	Set preferred size Set Background Set Alignment Set Spacing Set Padding Initialize new text as "Tanky Shooter"

4.6.3 Method

+ RootPane getRootPane()	Getter for RootPane's instance

4.7 Class WinnerPanel

4.7.1 Method

+ void draw(GraphicsContext graphicsContext)	Draw WinnerPanel's GraphicsContext

5.Package tile

5.1 Class Tile

This class initializes as a Class to collect tile's property and image

5.1.1 Fields

+ Image image	Each tile's image
- boolean hasCollisionField	collisionField of each tile

5.1.2 Method

Getter and setter of hasCollisionField	

5.2 Class TileManager implements Drawable

This class handles reading, creating and drawing maps from text file

5.2.1 Fields

- GamePanel GAMEPANEL	GamePanel for accessing screen's properties
- Tile[] tiles	Tile for accessing property and image
- int[][] mapTileCode	Code for displaying tiles
- String mapCode	Path of selected map

5.2.2 Constructor

+ TileManager(GamePanel GAMEPANEL)	Set GamePanel Initialize tiles and mapTileCode Call getTileImage() and loadMap()

5.2.3 Methods

+ void setMapCode(int mapInput)	Set mapCode as file path corresponding to mapInput
+ void getTileImage()	Load each tile's image
+ void loadMap()	Read map text file and then record tile's value into mapTileCode

+ void draw(GraphicsContext graphicsContext)	Draw TileManager's GraphicsContext according to mapTileCode
Getters of some fields	

6.Package interfaces

6.1 Drawable

This interface provides draw method for implemented classes

6.1.1 Method

+ void draw(GraphicsContext graphicsContext)	Method for implementing drawing GraphicsContext

6.2 Updatable

This interface provides update method for implemented classes

6.2.1 Methods

+ void update()	Method for implementing updating entity's properties