Tank Game Documentation

Created by

Pakin Puttanukulchai 6430282121

Nuttawash Trongsiriwat 6430119621

2110215 Programming Methodology

Semester 1 Year 2022

Chulalongkorn University

Tank Game

1. Introduction

The Tank Game is heavily inspired by battle city. You are assigned as a green tank, your objective is to kill as many enemy tank as possible. You can upgrade your tank's stat throughout the game by collecting "upgrader" but when you are stronger, the number of enemies also increase, which mean more fun and chaos.
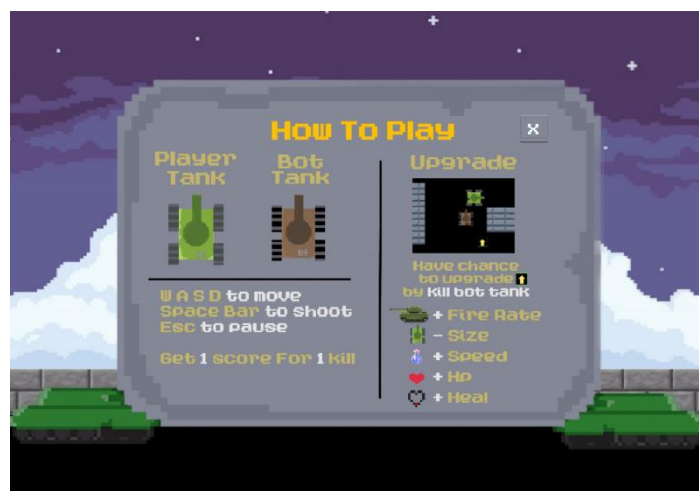
2. Game Controls

After running the game, Main Menu is the first screen of the game. Main Menu screen contains three buttons that are Play, Help and Exit button.

Main Menu Screen

You can click help button to open how to play screen. It maybe makes you understand the game.

How to play

After you click play button, you can choose map that you want to play. There are 3 maps now. By the way, you can click back button to main menu screen.
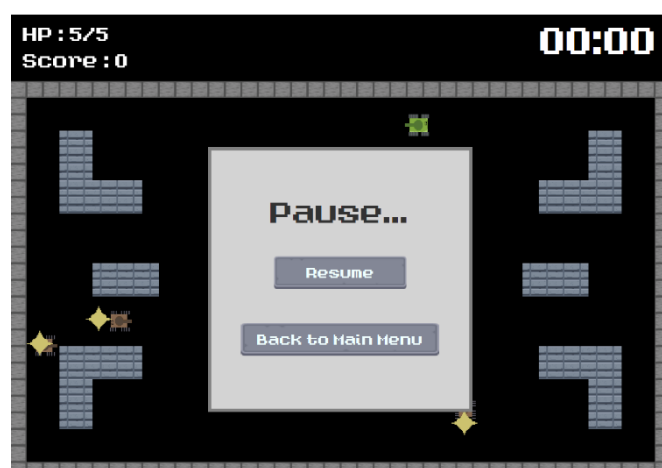

Map selection

The game will begin after you select a map. You are the green tank. The bot tank is brown tank. You can press "A" to go left , "W" to go up , "S" to go down , "D" to go right and "space bar" to shoot on keyboard.
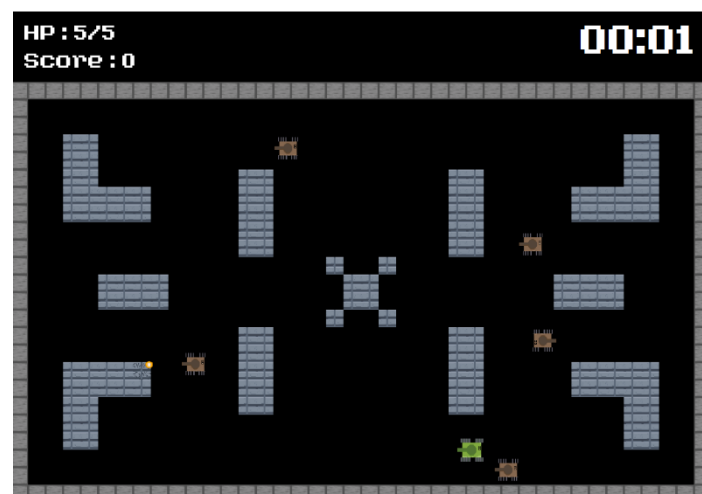

Main Game Screen

You also press "esc" to pause this game. You can choose to countinue the game or return to the main menu screen.

Pause Screen

When you kill a bot tank, your score will increase by one and you will have a chance to get an upgrader. The upgrade menu will appear when you receive the upgrader. You can select which stat to improve. You can improve your fire rate, size, speed, and health. You can also heal your tank to full health.
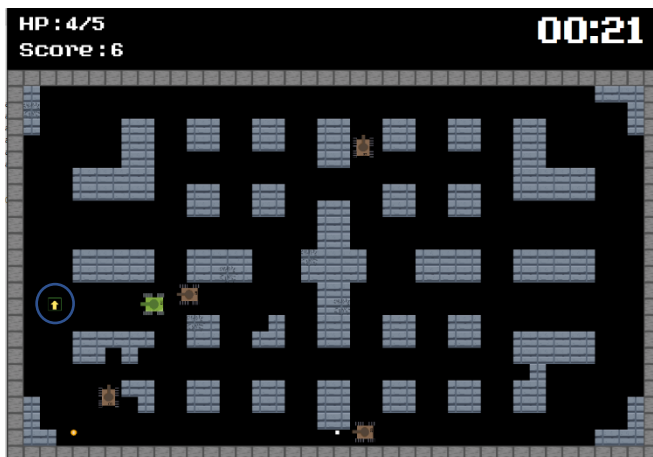
Upgrader random spawn after kill bot tank
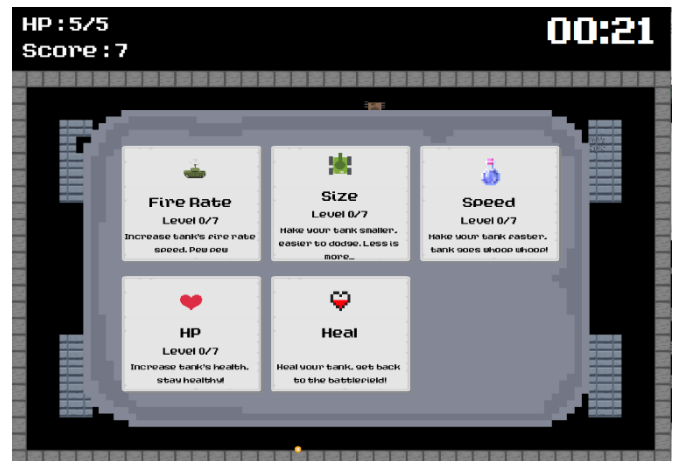
When receive upgrader

The game ends when your health reaches zero. The game's ending menu will appear. Your score is on the left side, and your total playing time is on the right. You can choose to restart the game or return to the main menu screen.
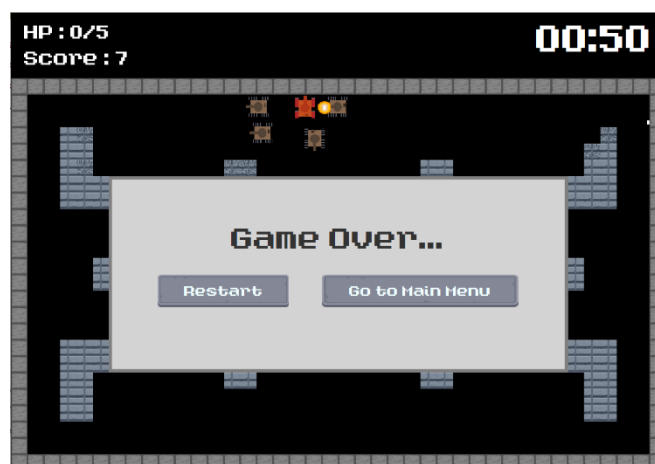
End game screen

3. Key Controls

Key	Description
W	Move your tank up
A	Move your tank left
S	Move your tank down
D	Move your tank right
Space bar	Your tank shoot
Esc	Pause your game

4. UML Diagrams

5. Class Details

1. Package application

1.1. class **Main** extends Application

1.1.1. Method

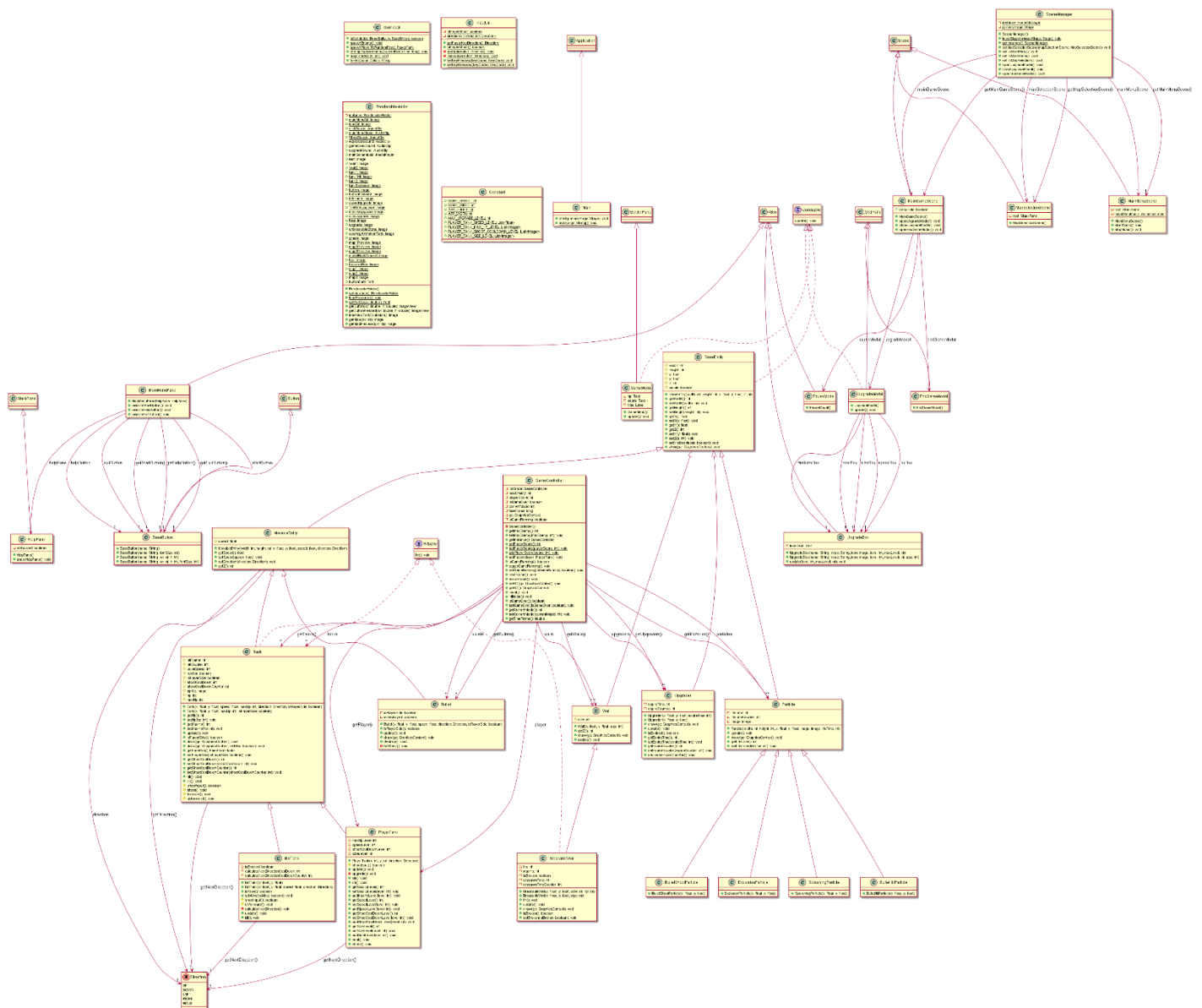| Name	Description
+ void start(Stage primaryStage)	- set primaryStage by sceneManager - set scene to main menu scene. - set title, icon and resizable. - show the stage
+ void main(String[ ] args)	Launch an application.

2. Package common

2.1. class constant

2.1.1. Fields

Name	Description
+ final int GAME_HEIGHT	A constant height of game is 600.
+ final int GAME_WIDTH	A constant width of game is 1000.
+ final int APP_HEIGHT	A constant height of application is 700.
+ final int APP_WIDTH	A constant width of application is 1000.
+ final int MAX_UPGRADE_LEVEL	A constant of max upgrade level is 7.
+ final List<Float> PLAYER_TANK_SPEED_LEVEL	Player tank speed constants for each level are 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, and 3.25.
+ final List<Integer> PLAYER_TANK_MAX_HP_LEVEL	Player tank max hp constants for each level are 5, 6, 7, 8, 9, 10, 11, 12.
+ final List<Integer> PLAYER_TANK_SHOOT_COOLDOWN_LEVEL	Player tank shoot cooldown constants for each level are 60, 40, 30, 25, 20, 15, 10.

+ final List<Integer> PLAYER_TANK_SIZE_LEVEL	Player tank size constants for each level are 40, 37, 34, 31, 28, 25, 23.

2.2 enum Direction

This enum contains every directions that tank can move.

Name	Description
– UP	Represent up direction.
– DOWN	Represent down direction.
– LEFT	Represent left direction.
– RIGHT	Represent right direction.
– HOLD	Represent hold situation.

2.3 interface Updatable

2.3.1. Method

Name	Description
+ void update()	This function that provides each entity that can be updated.

3. Package entity.base

3.1. abstract class BaseEntity implements Updatable

3.1.1. Fields

Name	Description
# int width	Width of that entity.
# int height	Height of that entity.
# float x	X-axis Position of that entity.
# float y	Y-axis Position of that entity.
# boolean visible	State that the entity is visible or not.

3.1.2. Constructor

Name	Description
+ BaseEntity (int width, int height, float x, float y)	Initialize all fields and set visible to true.

3.1.3. Method

Name	Description
+ *void draw(GraphicsContext gc)*	Draw an object, depending on the object.
+ Getter/Setter for each fields	(Getter for visible not need)

3.2. abstract class MovableEntity extends BaseEntity

3.2.1. Fields

Name	Description
# float speed	Speed value of that entity.
# Direction direction	The direction of entity is facing.

3.2.2. Constructor

Name	Description
+ MovableEntity (int width, int height, float x, float y, float speed, Direction direction)	Initialize all fields by super class.

3.2.3. Method

Name	Description
+ void setSpeed (float speed)	Setter for speed. If speed less than 0, set speed to 0.
+ Getter/Setter for each fields	

3.3. abstract class Particle extends BaseEntity

3.3.1. Fields

Name	Description
- int lifeTime	The number of frames that the particle will display.
- int lifeTimeCounter	Counter for lifeTime.
- Image image	Image of particle.

3.3.2. Constructor

Name	Description
+ Particle (int width, int height, float x, float y, Image image, int lifeTime)	- Initialize all fields. - Add this particle to ConcurrentLinkedQueue of particles

3.3.3. Method

Name	Description
+ void update ()	Decrease lifeTimeCounter by one. If lifeTimeCounter less than or equal to 0, set visible to false and remove this particle from ConcurrentLinkedQueue of particles.
+ draw (GraphicContext gc)	Draw this particle.
+ Getter/Setter for lifeTime	

3.4. abstract class Tank extends MovableEntity implements Hittable

3.4.1. Fields

Name	Description
# int hitFrame	When your tank is hit, this value is applicable. Set default to 15.
# int hitCounter	Counter for hitFrame set default to 0.
# int bulletSpeed	Speed of bullet that shoot from tank is 10.

# boolean isAlive	State that the tank is still alive or not.
# boolean isPlayerSide	State that this tank is player or not.
# int shootCoolDown	Cooldown on each shoot.
# int shootCoolDownCounter	Counter for shootCoolDown.
# Image sprite	Image of tank.
# int hp	Hp of tank at that moment.
# int maxHp	Max hp of tank.

3.4.2. Constructor

Name	Description
+ Tank(float x, float y, int maxHp, boolean isPlayerSide)	- Initialize all fields. - set width and height to 40. - set speed to 2. - set direction to UP. - set shootCoolDown to 50. - set shootCoolDownCounter to 0. - set hp to maxHp.
+ Tank(float x, float y, float speed, int maxHp, Direction direction, boolean isPlayerSide)	- Initialize all fields. - set width and height to 40. - set speed to 2. - set shootCoolDown to 50. - set shootCoolDownCounter to 0. - set hp to maxHp. - Add this tank to ConcurrentLinkedQueue of tanks

3.4.3. Method

Name	Description
+ void setHp (int hp)	Setter for hp. If hp more than maxHp, set hp to maxHp. If hp less than 0, set hp to 0.
+ void setMaxHp (int maxHp)	Setter for maxHp. If maxHp less than 1, set maxHp to 1.

+ void update ()	- If shootCoolDownCounter more than 0, decrease by shootCoolDownCounter one. - If there is push a shoot, call function shoot () - Call function forward () - If hitCounter more than 0, decrease hitCounter by one.
+ void draw(GraphicsContext gc)	Draw this tank.
+ void draw(GraphicsContext gc, boolean isHitted)	Draw this tank depend on direction and it is hitted or not. - if direction is UP don't rotate this tank image. - if direction is DOWN rotate this tank image 180 degree. - if direction is LEFT rotate this tank image 270 degree. - if direction is RIGHT rotate this tank image 90 degree. - if tank is hit, draw image tank1Hit from RenderableHolder.
+ Pair<Float, Float> getFacePos()	Return the pair of position that tank is facing depend on direction.
+ void setShootCoolDown (int shootCoolDown)	Setter for shootCoolDown. If shootCoolDown less than 0, set shootCoolDown to 0.
+ void setShootCoolDownCounter (int shootCoolDownCounter)	Setter for shootCoolDownCounter. If shootCoolDownCounter less than 0, set shootCoolDownCounter to 0.
+ void hit ()	- set hitCounter is hitFrame. - Draw image that tank is hitted. - Decrease hp by one. - If hp less than or equal to 0, call function kill () because that tank is killed.
+ void kill ()	- remove this tank from ConcurrentLinkedQueue of tanks. - set isAlive to false. - draw a explosion particle in this position.
# boolean shootInput ()	Receive an input from keyboard. It true if input is space bar.
# void shoot ()	- If the tank is on cooldown, do nothing. - Initialize bullet to the face position.
# void forward ()	Make the tank go forward according to direction.
# void unforward ()	When that entity overlaps with something, this function is called.

# Direction getNextDirection ()	Next Direction that receive an input from keyboard.
+ Getter/Setter for remain fields	

4. Package entity.interfaces

4.1. interface Hittable

4.1.1. Method

Name	Description
+ void hit ()	This function is used for entity that can be hit.

5. Package entity.Particle

5.1. class BulletHitParticle extends Particle

5.1.1. Constructor

Name	Description
+ BulletHitParticle(float x, float y)	Initialize by super class to setting value. - Width and Height is 10. - Image is image of bullet explosion to entity. - lifeTime is 40.

5.2. class BulletShootParticle extends Particle

5.2.1. Constructor

Name	Description
+ BulletShootParticle(float x, float y)	Initialize by super class to setting value. - Width and Height is 20. - Image is image of bullet explosion from tank. - lifeTime is 5.

5.3. class **ExplosionParticle** extends Particle

5.3.1. Constructor

Name	Description
+ ExplosionParticle(float x, float y)	Initialize by super class to setting value. - Width and Height is 40. - Image is image of tank explosion. - lifeTime is 55. - play explosion sound.

5.4. class **SpawningParticle** extends Particle

5.4.1. Constructor

Name	Description
+ ExplosionParticle(float x, float y)	Initialize by super class to setting value. - Width and Height is 50. - Image is image of opening animation of tank. - lifeTime is 40.

6. Package entity

6.1. class **BotTank** extends Tank

6.1.1. Fields

Name	Description
# boolean isBlocked	State that bot tank is blocked or not.
# int calculateNextDirectionCoolDown	Cooldown after call method calculateNextDirection ().
# int calculateNextDirectionCoolDownCounter	Counter for calculateNextDirectionCoolDown.

6.1.2. Constructor

Name	Description
+ BotTank(float x, float y, float speed, Direction direction)	Initialize by super class to setting value. - set isAlive to true. - set isBlocked to false. - set calculateNextDirectionCoolDown to random - set calculateNextDirectionCoolDownCounter to 0 - Initialize sprite to tank image. - set bulletSpeed to 5. - set shootCoolDown tp 80.

6.1.3. Method

Name	Description
+ boolean isAlive ()	Return this bot tank is alive or not
+ void setAlive (boolean isAlive)	Setter for isAlive
# boolean shootInput ()	Random whether this bot tank will shoot or not.
# void unforward ()	- Call method unforward() from super class and set isBlocked to true.
- void calculateNextDirection ()	Find the closest path between player tank and bot tank then go that way.
+ void update ()	Update where this bot tank should go. - Call method calculateNextDirection () - Call method update() from super class.
+ void kill ()	- Add player score by one. - Random drop upgrader from kill this bot tank. - Call method kill() from super class
# Direction getNextDirection ()	Return next direction

6.2. class BreakableWall extends Wall implements Hiitable

6.2.1. Fields

Name	Description
- int hp	Hp of this wall.
- int maxHp	Max hp of this wall. Default value is 2
- boolean isBroken	State that this wall is broken or not.

Name	Description
- int respawnTime	The time when the wall will respawn. Default value is 15*60
- int respawnTimeCounter	Counter for respawnTime.

6.2.2. Constructor

Name	Description
+ BreakableWall(float x, float y, int size, int hp)	Initialize by super class to setting value. - set maxHp to 2. - set hp to maxHp.
+ BreakableWall(float x, float y, int size)	Initialize by super class to setting value. - set maxHp to hp.

6.2.3. Method

Name	Description
+ void hit ()	Method for this wall is hit. - When this wall is hit, decrease hp by one. - If hp less than or equal 0, set isBroken to true and respawnTimerCounter to respawnTime.
+ void update ()	Updates when this wall is hit. - start counting cooldown to respawn. - If tank is in the same position as this wall, it will be counted again. - call method update() from super class.
+ void draw (GraphicsContext gc)	Draw this wall. - If this wall isn't broken and hp is max hp draw wall . - If this wall isn't broken and hp isn't max hp draw cracked wall .
+ boolean isBroken ()	Return this wall is broken or not.
+ void setBroken (boolean isBroken)	Setter for isBroken.

6.3. class Bullet extends MovableEntity

6.3.1. Fields

Name	Description
– boolean isPlayerSide	This bullet belongs to the player side or not.
– boolean isDestroyed	This bullet is destroyed or not.

6.3.2. Constructor

Name	Description
+ Bullet(float x, float y, float speed, Direction direction, boolean isPlayerSide)	Initialize by super class to setting value. - Add this bullet to ConcurrentLinkedQueue of bullets. - set isDestroyed to false. - Initialize bullet particle this position.

6.3.3. Method

Name	Description
+ boolean isPlayerSide ()	Return this bullet belongs to the player side or not.
+ void update ()	Make bullet go forward according to tank direction.
+ void draw (GraphicsContext gc)	Draw this bullet. - set color to white.
+ void destroy ()	This method will call when this bullet disappear. - remove this bullet from ConcurrentLinkedQueue of bullets. - set isDestroyed to true.
– void hitCheck ()	If the bullet hits the entity, create an explosion particle and call method hit() for that entity.

6.4. class PlayerTank extends Tank

6.4.1. Fields

Name	Description
- int maxHpLevel	Max hp according to level at that moment.
- int speedLevel	Speed according to level at that moment.
- int shootCoolDownLevel	Shoot cooldown according to level at that moment.
- int sizeLevel	Size according to level at that moment.

6.4.2. Constructor

Name	Description
+ PlayerTank(int x, int y, Direction direction)	Initialize by super class to setting value. - set maxHpLevel to 0 - set speedLevel to 0 - set shootCooldownLevel to 0 - set sizeLevel to 0

6.4.3. Method

Name	Description
# Direction getNextDirection ()	Return this bullet is player side or not.
# boolean shootInput ()	Check if player press space bar to shoot or not.
+ void update ()	When this player tank receive upgrader, the upgrade screen will be displayed.
- void upgrade ()	Open upgrade screen.
+ void hit ()	- If this player tank is hit when it still red image, do nothing. (immune) - Call method hit() from super class.
+ void kill ()	- Call method kill() from super class. - set game over to true. - open end game screen.
+ int getMaxHpLevel ()	Getter for maxHpLevel.
+ void setMaxHpLevel (int level)	Setter for maxHpLevel. - If level less than 0, level is 0. - if level more than max upgrade level, level is max upgrade level

+ void addMaxHpLevel (int level)	Increase maxHplevel by level.
+ int getSpeedLevel ()	Getter for speedLevel.
+ void setSpeedLevel (int level)	Setter for speedLevel. - If level less than 0, level is 0. - if level more than max upgrade level, level is max upgrade level
+ void addSpeedLevel (int level)	Increase speedLevel by level.
+ int getShootCoolDownLevel ()	Getter for shootCoolDownLevel.
+ void setShootCoolDownLevel (int level)	Setter for shootCoolDownLevel. - If level less than 0, level is 0. - if level more than max upgrade level, level is max upgrade level
+ void addShootCoolDownLevel (int level)	Increase shootCoolDownLevel by level.
+ int getSizeLevel ()	Getter for sizeLevel.
+ void setSizeLevel (int level)	Setter for sizeLevel. - If level less than 0, level is 0. - if level more than max upgrade level, level is max upgrade level
+ void addSizeLevel (int level)	Increase sizeLevel by level.
+ void heal ()	Set hp to max hp.
+ void shoot ()	- If player tank is on cooldown, do nothing. - play shoot sound. - call method shoot() from super class.

6.5. class Upgrader extends BaseEntity

6.5.1. Fields

Name	Description
– int expireTime	The time that this upgrader will be on the screen.
– int expireCounter	Counter for expireTime.

6.5.2. Constructor

Name	Description
+ Upgrader(float x, float y, int expireTime)	Initialize by super class to setting value. - set size to 20x20 - add this upgrader to ConcurrentLinkedQueue of upgraders.
+ Upgrader(float x, float y)	Use default constructor with expireTime is 60*10.

6.5.3. Method

Name	Description
+ void update ()	If upgrader id expired, remove it from screen. If not , decrease ExpireCounter by one.
+ void (GraphicsContext gc)	Draw this upgrader.
+ boolean isExpired ()	Check if expireCounter less than or equal 0.
+ void setExpireCounter (int expireCounter)	Setter for ExpireCounter. If ExpireCounter less than 0, set it to 0.
+ void decreaseExpireCounter ()	Decrease ExpireCounter by one.
+ Getter/Setter of remain fields	

6.6. class Wall extends BaseEntity

6.6.1. Fields

Name	Description
# int size	Width and Height of wall.

6.6.2. Constructor

Name	Description
+ Wall(float x, float y, int size)	Initialize by super class to setting value. - Width and height is size. - add this wall to ConcurrentLinkedQueue of walls.

6.6.3. Method

Name	Description
+ void draw(GraphicsContext gc)	Draw map frame with unbreakablestone image in game scene.

7. Package gui

7.1. class BaseButton extends Button

Button

Pressed Button

7.1.1. Constructor

Name	Description
+ BaseButton(String name, int w, int h, int fontSize)	Initialize with the following specifications. - Initialize text with name parameter. - set font to 8bit.ttf from RenderableHolder. - set font size with fontsize parameter. - set the color Azure as the text color. - set background to null. - set padding to empty. - set content display to center. - set graphic of button to button image from RenderableHolder. - set onMouseEntered to handle by this following : - set graphic of button to pressed button image from RenderableHolder. - set cursor to hand. - set onMouseExited to handle by this following : - set graphic of button to button image from RenderableHolder. - set cursor to default.
+ BaseButton(String name, int w, int h)	Use default constructor to set value - font size is 15
+ BaseButton(String name, int fontSize)	Use default constructor to set value - width is 200 - height is 50

8. Package gui.mainGame

8.1. class EndGameModal extends GridPane

8.1.1. Constructor

Name	Description

+ EndGameModal ()	Initializes with the following specifications.
	 - set visible to false.
	 - set Vgap and Hgap to 20.
	 - set padding to 50.
	 - set alignment to center position.
	 - set Max height to 300.
	 - set Max width to 700.
	 - set background to light gray color.
	 - set border to gray color.
	 - Initialize header label with text "Game Over…".
	 - set header label font to 8bit.ttf.
	 - Initialize buttonList Hbox by this following :
	    - set spacing to 50.
	    - set padding to 20.
	 - Initialize restart button by BaseButton class with text "Restart"
	 - set restart button OnMouseClicked to handle by this following :
	    - set visible to false.
	    - reset game.
	    - set game running to true.
	 - Initialize exit button by BaseButton class with text "Go to Main Menu"
	 - set exit button OnMouseClicked to handle by this following :
	    - set visible to false.
	    - set scene to mainmenu.
	 - add restart and exit button to buttonList.
	 - add header label and buttonList to this grid pane by position in picture.

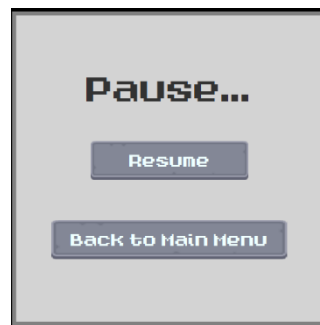**8.2. class GameMenu** extends BorderPane implements Updatable

8.2.1. Fields

Name	Description
# Text hp	Text that show your remain tank hp.
# Text score	Text that show your score.
# Label time	Time that your tank can survive.

8.2.2. Constructor

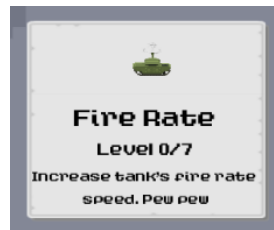| Name	Description

+ GameMenu ()	Initializes with the following specifications.
	 - set prefer size with constant value from Constant class in common package.
	 - set padding to 20.
	 - set background to black color.
	 - Initialize hp text by this following :
	     - set text to "HP : 0/0".
	     - set font to 8bit.ttf with size 20.
	     - set the color white as text color.
	 - Initialize score text by this following :
	     - set text to "Score : 0".
	     - set font to 8bit.ttf with size 20.
	     - set the color white as text color.
	 - Initialize time text by this following :
	     - set text to "00:00".
	     - set font to 8bit.ttf with size 40.
	     - set the color white as text color.
	 - Initialize left VBox pane by this following :
	     - set spacing to 20.
	     - add hp and score text to this pane.
	 - Initialize right VBox pane by this following :
	     - add time text to this pane.
	 - add left and right VBox pane to this borderpane by position in picture.

+ void update ()	- set hp, score and time text with value at that moment with the same pattern as before.

8.3. class PauseModal extends VBox

8.3.1. Constructor

Name	Description

+ PauseModal ()	Initializes with the following specifications.
	- set visible to false.
	- set alignment to center position.
	- set max height and wight to 400.
	- set padding to 20
	- set spacing to 50
	- set background to light gray color.
	- set border to gray color.
	- Initialize header label with text "Pause...".
	- set header label font to 8bit.ttf with size 30.
	- Initialize resume button by BaseButton class with text "Resume"
	- set resume button OnMouseClicked to handle by this following :
	  - play click sound.
	  - set visible to false.
	  - set game running to true.
	- Initialize back button by BaseButton class with text "Back to Main Menu"
	- set back button OnMouseClicked to handle by this following :
	  - play click sound.
	  - set visible to false.
	  - set scene to main menu.
	- add header label, resume button and back button to this VBox pane by position in picture.

8.4. class UpgradeBox extends VBox

8.4.1. Fields

Name	Description
– Text levelText	Text that show level for each type upgrade.

8.4.2. Constructor

Name	Description

+ UpgradeBox(String name, String desc, Image icon, int level, int maxLevel, int size)	Initializes with the following specifications.
	- set padding to 10.
	- set alignment to center position.
	- set spacing to 15.
	- set max width to 220.
	- set max height to 200.
	- set prefer height to 200.
	- set background to box image from RenderableHolder.
	- Initialize iconView from icon image parameter and set width and height to size parameter.
	- Initialize title text with name in parameter.
	- set title text font to 8bit.ttf with size 14.
	- Initialize levelText with value by following :
	  – if maxLevel equal to 0 , levelText is "".
	  – if maxLevel not equal to 0 , levelText is "Level " + level + "/" + maxLevel.
	- Initialize descText text by following :
	  – set text with desx in parameter.
	  – set Wrapping width to 200.
	  – set line spawing to 12.
	  – set this text to center position.
	- add iconView, title, levelText and descText to this VBox pane.
	- set this pane onMouseEntered to handle by this following :

	- resize box image.
	   - set cursor to hand.
	- set this pane onMouseExited to handle by this following :
	   - resize box image back.
	   - set cursor to default.

+ void update ()	- set levelText with value at that moment with the same pattern as before.

8.5. class UpgradeModal extends GridPane implements Updatable

8.4.1. Fields

Name	Description
+ UpgradeBox sizeBox	Upgrade box for size upgrade.
+ UpgradeBox speedBox	Upgrade box for speed upgrade.
+ UpgradeBox hpBox	Upgrade box for hp upgrade.
+ UpgradeBox fireRateBox	Upgrade box for fire rate upgrade.
+ UpgradeBox healBox	Upgrade box for heal upgrade.

8.4.2. Constructor

Name	Description
+ UpgradeModal ()	Initializes with the following specifications. - set visible to false. - Initialize sizeBox with value at that moment and description is "Make your tank smaller, easier to dodge. Less is more". - Initialize speedBox with value at that moment and description is "Make your tank faster, tank goes whoop whoop!". - Initialize hpBox with value at that moment and description is "Increase tank's health, stay healthy!". - Initialize fireRateBox with value at that moment and description is "Increase tank's fire rate speed. Pew pew". - Initialize healBox with value at that moment and description is "Heal your tank, get back to the battlefield!". - set max height to 550. - set max width to 800. - set padding to 80. - set Vgap and Hgap to 5. - set background to upgrade modal image from RenderableHolder. - set each upgrade box level up when click mouse to it. - add sizeBox, speedBox, hpBox, healBox and fireRateBox to this grid pane by position in picture.
+ void update ()	- set each upgrade box to correct context value at that moment.

9. Package gui.mainMenu

9.1. class HelpPane extends StackPane

9.1.1. Fields

Name	Description
– boolean isShowed	State that help pane is showed or not.

9.1.2. Constructor

Name	Description

+ HelpPane ()	Initializes with the following specifications.
	- set image view to how to play image from RenderableHolder.
	- set max size to 700x500.
	- set position this pane to 1000.
	- set isShowed to false.
	- Initialize close button by this following :
	   - use BaseButton class.
	   - set text to "X".
	   - set size to 40x40.
	   - when mouse click this button call function showHelppane () and play click sound.
	   - set this button to position in picture.

9.1.3. Method

Name	Description
+ showHelpPane ()	– if isShowed is false, move help pane from position x is 1000(Out of range) to x is 0 . – if isShowed is true, move help pane from position x is 0 to x is 1000 (Out of range).

9.2. class **MainMenuPane** extends VBox

9.2.1. Fields

Name	Description
- BaseButton startButton	Start button.
- BaseButton helpButton	Help button.
- BaseButton exitButton	Exit button.
- HelpPane helpPane	Show help pane when click help button.

9.2.2. Constructor

Name	Description
+ MainMenuPane(HelpPane helpPane)	Initializes with the following specifications. - set alignment to top center position. - set spacing to 30 - set padding to 200. - set help pane. - Initialize game name by this following : - set text to "Tank Game". - set text font to 8bit.ttf with size 40. - set image to be text frame. - set content display to center. - Initialize startButton from function initializeStartButton(). - Initialize helpButton from function initializeHelpButton(). - Initialize exitButton from function initializeExitButton() - add game name text, startButton, helpButton, exitButton to this VBox pane.

9.2.3. Method

Name	Description
+ void initializeStartButton()	Initialize startButton by BaseButton class. - text in button is "Play". - size is 190x50.

	- set onMouseClicked to handle by this following :
	  - play click sound.
	  - set scene to map selection.

+ void initializeHelpButton()	Initialize helpButton by BaseButton class.
	- text in button is "Help".
	- size is 190x50.
	- set onMouseClicked to handle by this following :
	  - play click sound.
	  - show help pane.

+ void initializeExitButton()	Initialize exitButton by BaseButton class.
	- text in button is "Exit".
	- size is 190x50.
	- set onMouseClicked to handle by this following :
	  - play click sound.
	  - exit this program.

+ Getter for each field.	

10. Package logic

10.1. class GameController

10.1.1. Fields

Name	Description
- GameController instance	Singleton of GameController class.
- int maxEnemy	Maximum number of bot tank.
- int playerScore	Player score at that moment.
- boolean isGameOver	State that game is over or not.
- int currentMapId	Map Id currently in use.
- long timeFrame	Count the number of frames is rendered.
- GraphicsContext gc	GraphicContext for canvas.
- ConcurrentLinkedQueue<Tank> tanks	ConcurrentLinkedQueue is used instead of ArrayList because of performance reason. All active tanks in the game are listed here.
- ConcurrentLinkedQueue<Bullet> bullets	ConcurrentLinkedQueue is used instead of ArrayList because of performance reason. All active bullet in the game are listed here.

- ConcurrentLinkedQueue<Wall> walls	ConcurrentLinkedQueue is used instead of ArrayList because of performance reason. All active walls in the game are listed here.
- ConcurrentLinkedQueue<Upgrader> upgraders	ConcurrentLinkedQueue is used instead of ArrayList because of performance reason. All active upgrader in the game are listed here.
- ConcurrentLinkedQueue<Particle> particles	ConcurrentLinkedQueue is used instead of ArrayList because of performance reason. All active particle in the game are listed here.
- PlayerTank player	A tank that player control.
- boolean isGameRunning	State that game is running or not.

10.1.2. Constructor

Name	Description
+ GameController ()	- set isGameRunning to false. - set maxEnemy to 0. - set playerScore to 0. - set isGameOver to true. - set timeframe to 0.

10.1.3. Method

Name	Description
+ GameController getInstance ()	Getter method of instance.
+ void setPlayerScore (int playerScore)	Setter for playerScore. If playerScore less than 0, set it to 0.
+ void addPlayerScore(int score)	Increase playerScore by score.
+ void nextFrame ()	If game not running, do nothing. - Increase timeframe by one. - increase the difficulty according to level upgrade. - All entities call function update () - draw all entities.
+ void drawFrame ()	If game not running, do nothing. - clear screen to default for redrawing. - draw all entities.
+ void reset ()	- clear all entities. - set playerScore to 0. - set timeframe to 0.

	- initialize new map and populate new tank.
+ void initialize ()	- create border walls. - set isGameOver to false. - load map by using mapLoader function. - Populate player and enemies tank.
+ Getter/Setter for remain fields	

10.2. class GameUtil

10.2.1. Method

Name	Description
+ boolean isCollided(BaseEntity a, BaseEntity b)	Check if entity a overlaps with entity b or not.
+ void spawnEnemy()	Spawn bot tank to random position and that position must not overlap with anything.
+ PlayerTank spawnPlayerToRandomPos()	Spawn player tank to random position at the start of the game.
+ void attemptSpawnEnemy(long currentNanoTime)	This method will try to spawn cars every second by checking if currentNanoTime divisible by 60 or not before running.
+ void mapLoader(int idx)	Check every pixel in map image if that pixel is #CCCCCC, create wall.
+ String toHex(Color color)	Convert Class color to string hex color.

10.3. class InputUtil

10.3.1. Field

Name	Description
- boolean isPlayerShoot	State that player press key to shoot or not.
- LinkedList<Direction> directions	Linked list contain all direction that player press key on that direction.

10.3.2. Method

Name	Description
<u>+ Direction getPlayerKeyDirection ()</u>	If directions list is empty, return hold direction.
<u>+ boolean isPlayerShoot ()</u>	Return player shoot or not.
<u>- void push(Direction direction)</u>	- Receive an input from keyboard. - if player press new direction, add it to directions linked list.
<u>- void remove(Direction direction)</u>	- remove direction from directions linked list.
<u>+ void setKeyPressed(KeyCode keyCodes</u>	- Add direction according to an input from keyboard. - If player press space bar, set isPlayerShoot to true.
<u>+ void setKeyReleased(KeyCode keyCodes</u>	- Remove direction that player isn't press from directions linked list - if player isn't press space bar, set isPlayerShoot to false.

11. Package scenes

11.1. class MainGameScene extends Scene

11.1.1. Fields

Name	Description
+ boolean isEscHold	State that game pause or not.
+ UpgradeModal upgradeModal	Upgrade screen that will show when player receive upgrader.
+ EndGameModal endGameModal	End game screen that will show when game is over.
+ PauseModal pauseModal	Pause screen that will show when player press esc.

11.1.2. Constructor

Name	Description
+ MainGameScene ()	- set prefer size with constant value from constant class in common package. - Initialize canvas. - Initialize gameMenu.

	- Initialize gamePane by StackPane and add canvas to it.
	- Initialize root by VBox.
	- add gameMenu and gamePane to root.
	- Initialize entity to canvas.
	- Initialize endGameModal, upgradeModal and pauseModal.
	- Add endGameModal, upgradeModal and pauseModal to gamePane.
	- Set isEscHold to false.
	-Set onKeyPressed to handle by following :
	– If player press esc show pauseModal.
	- set isEscHold to true.
	-Set onKeyReleased to handle by following :
	- If player press esc close pauseModal .
	- set isEscHold to false.
	- Play main game music.
	- Initialize timer.

11.1.3. Method

Name	Description
+ void openUpgradeModal ()	Call this function when a player receives an upgrader from a bot tank. - set upgraderModeal visible to true. - set game running to false.
+ void closeUpgradeModal ()	This function is called when the player clicks the upgrade. - set upgraderModeal visible to false. - set game running to true.
+ void openEndGameModal ()	Call this function after the player has died. - set upgraderModeal visible to true. - set game running to false.

11.2. class MainMenuScene extends Scene

11.2.1. Fields

Name	Description
- StackPane root	Root for main menu scene.
- AnimationTimer mainMenuMusic	Main menu music.

11.2.2. Constructor

Name	Description
+ MainMenuScene ()	- set prefer size with constant value from constant class in common package. - Create root with main menu image. - Create help pane - Create mainmenu pane - Initialize mainMenumusic to handle by following : if music doesn't play, play it. - Play mainMenuMusic - Add help pane and main menu pane to root. - set root

11.2.3. Method

Name	Description
+ void startMusic ()	Play background music.
+ stopMusic ()	Stop background music.

11.3. class MapSelectionScene extends Scene

11.3.1. Fields

Name	Description
- StackPane root	Root for map selection scene.

11.3.2. Constructor

Name	Description
+ MapSelectionScene ()	- Set prefer size with constant value from constant class in common package. - Initialize root with main menu image - Initialize maps pane by HBox and add map preview to this pane. - Initialize all component pane by VBox. - Create text "Map Select" with font 8bit.ttf and size 40.

	- Create BaseButton with text "Back". When click this button set scene to main menu. - Add maps pane, text and back button to all component pane. - Add all component pane to root.

12. Package sharedObject

12.1. class RenderableHolder

12.1.1. Fields

Name	Description
- final RenderableHolder instance	Singleton for RenderableHolder class
+ Image, AudioClip and Font	

12.1.2. Method

Name	Description
+ void loadResource()	Loading all resources with class loader.
+ Font getFont(double size)	Return 8bit.ttf font by size.
+ ImageView getButton(double w, double h)	Return button frame by size.
+ ImageView getButtonPressed(double w, double h)	Return pressed button frame by size.
+ Image loadNewTankExplosion()	Return new explosion gif.
+ Image getMap(int idx)	Return map image depend on idx.
+ Image getMapPreview(int idx)	Return map preview image depend on idx.

12.2. class SceneManager

12.2.1. Fields

Name	Description
- SceneManager instance	Singleton for SceneManger class

- Stage primaryStage	The main stage of the application.
- MainMenuScene mainMenuScene	Main menu scene.
- MainGameScene mainGameScene	Main game scene.
- MapSelectionScene mapSelectionScene	Map selection scene.

12.2.2. Constructor

Name	Description
+ SceneManager ()	- Initialize mainMenuScene. - Initialize mainGameScene. - Initialize mapSelectionScene.

12.2.3. Method

Name	Description
+ SceneManager getInstance ()	Getter method of instance.
+ void injectStage (Stage primaryStage)	Setter for primaryStage
+ void openEndGameModal ()	- open end game screen. - play end game sound.
+ void openUpgradeModal ()	- open upgrade screen. - play upgrade sound.
+ void setMapSelectionScene (MapSelectionScene mapSelectionScene)	Setter for mapSelectionScene.
+ void setToMainGame ()	- set scene to main game. - set game running to true. - call function reset () to reset all entities in game. - stop main menu music.
+ void setToMainMenu ()	- set scene to main menu. - set game running to false. - play main menu music.
+ void setToMainGame ()	- set scene to map selection. - set game running to false.
+ Getter for remain field	