Cy3erPunk Manual

Introduction

This game is an adaptation of original space invader to spice thing up new game feature are invented. Shield, Boss and level were introduced in the game which kind of like mini MMO that you can level up along the stage, New AI system that will grantee that AI will never be the same because it will shoot randomly every time you start the game, so there will be no walkthrough for the game, Good luck :P.
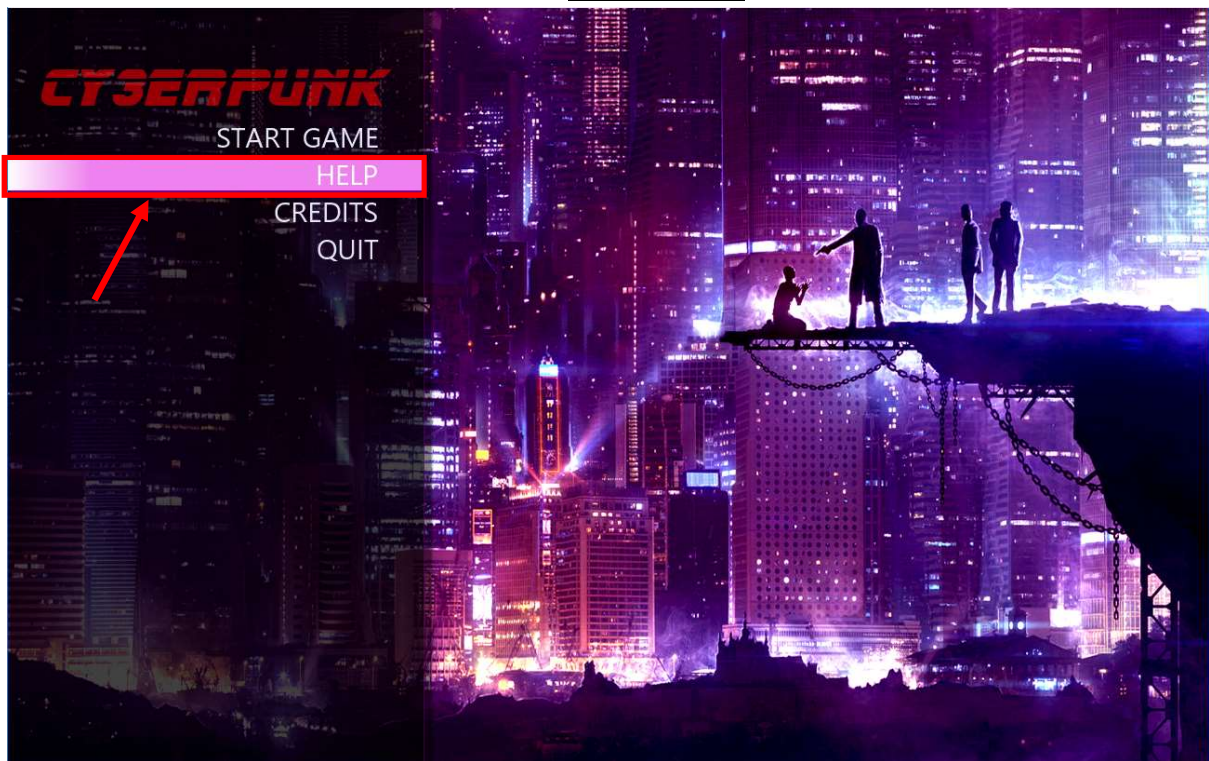
Getting Started

Main Menu
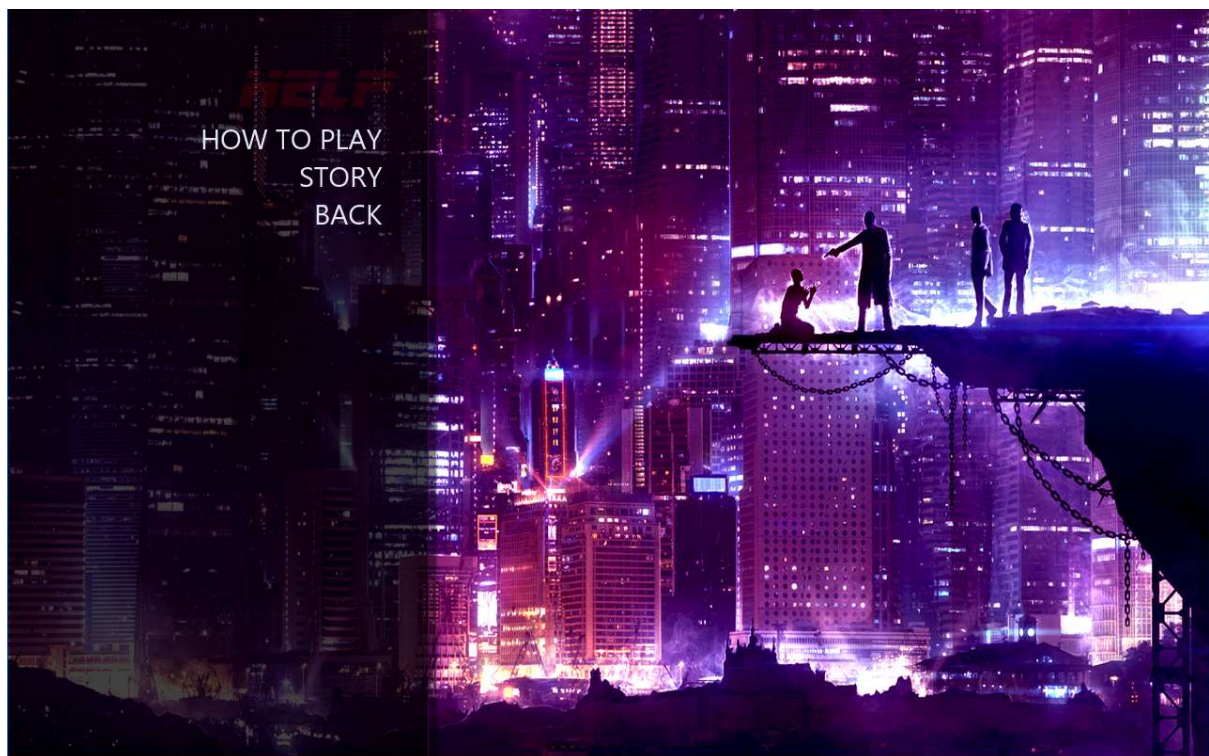

(a actual picture of a game main menu)

In main menu there are 4 menu items to be selected from each has it own function, So be aware the each one are different. In the top left logo will appear to blink to create the aesthetic of the game.
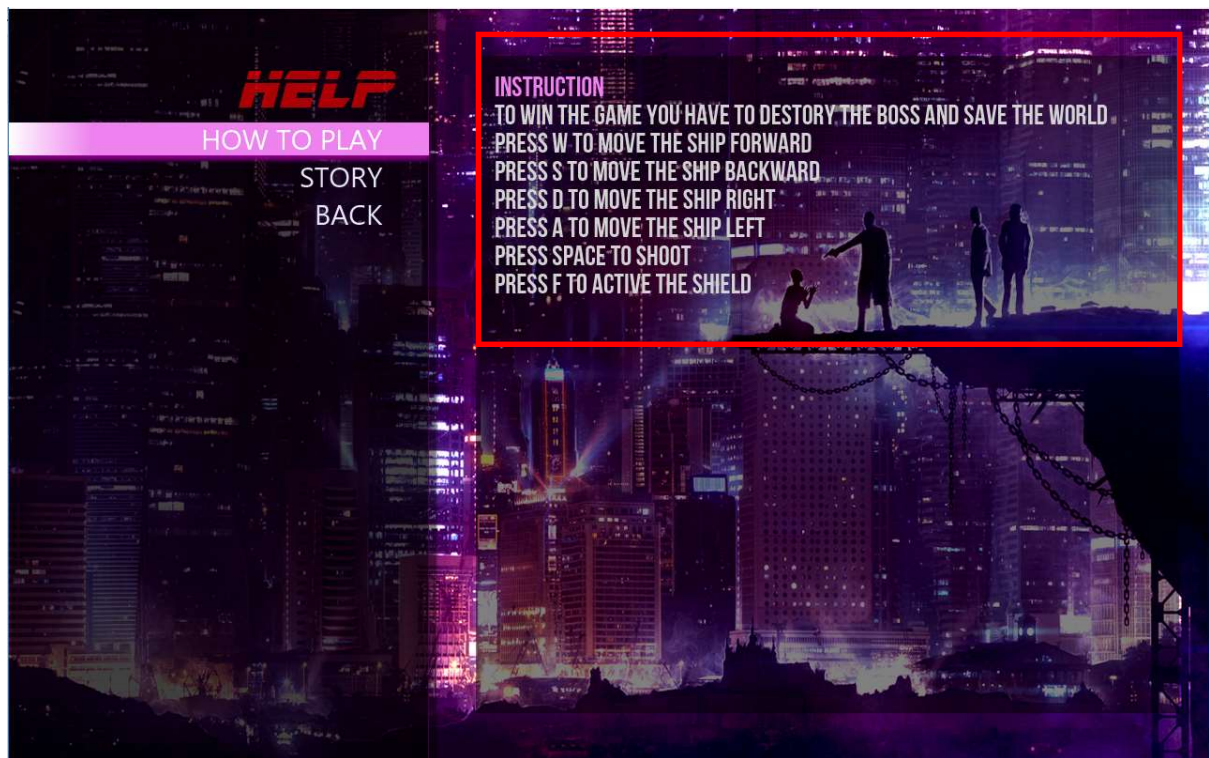
1.Help Menu

   This menu will guide you through to how to play, control your character or the main story of the game.

   when you selected help menu it will take you through the help menu you can choose the item you prefer to know. As you see ,you could notice two menu item to selected from in help menu. If you click how to play.

The instruction of the control will show up on the right as seen above. The instruction is very simple. You use w,a,s,d to move around and press f to engage the shield but in order for a shield to work it must reach the minimum shield charge which is label in red line on the shield bar.

When you click on the story menu story of the game will show up on the right similar to the how to play.

If you press back it will take you back to main menu.

Credit menu
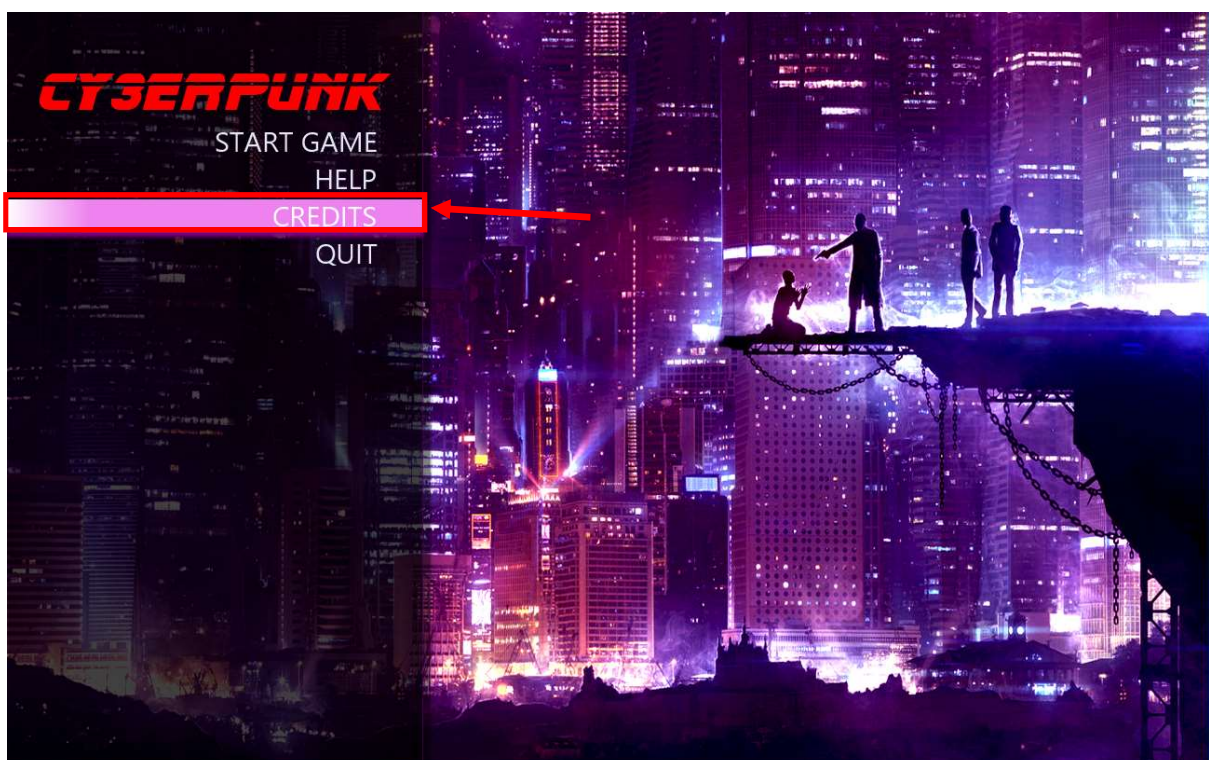
This menu will show the creator of the game .

Quit menu

Quit menu will exit the game for you if you in game you can use esc key to exit the game as well.

Start Game menu

This menu starts all the fun.
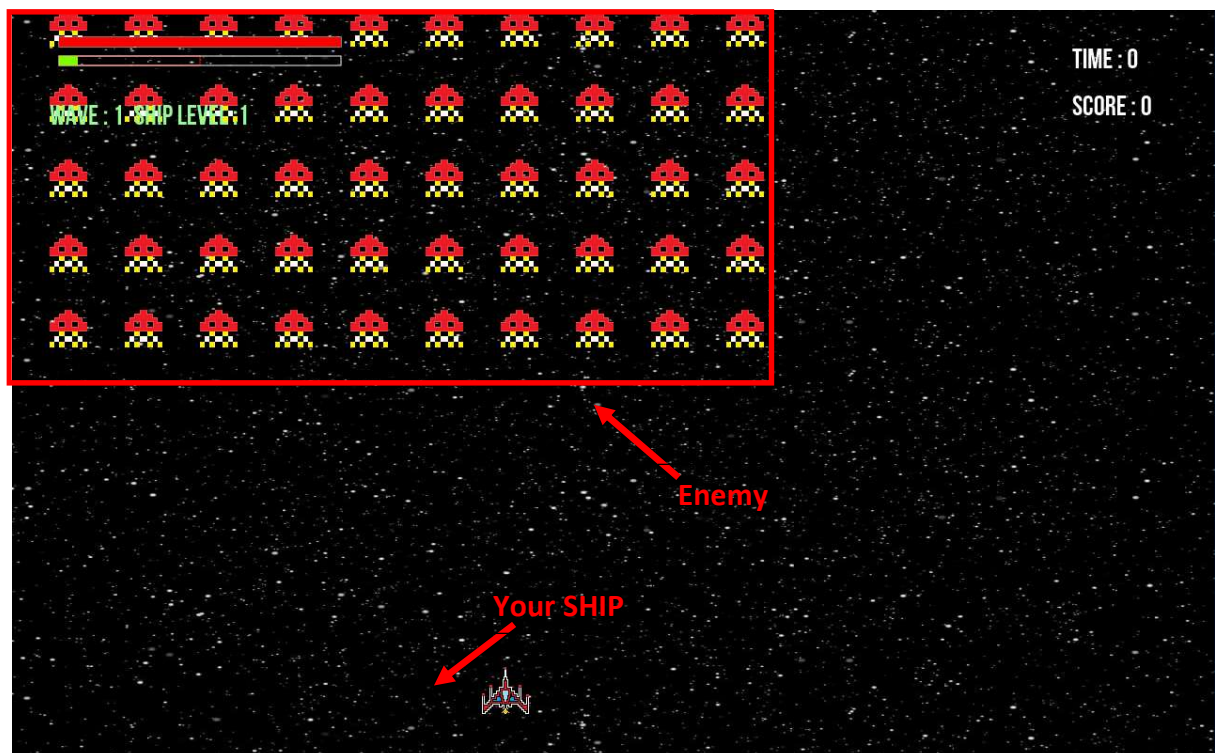
This is the start of the game you will start with wave 1 and ship level 1.

Let's go through Basic.

As you can see on the bottom of the screen your ship is there. The goal is to defeat the alien boss but to that you must go through three wave of enemy which will have different difficulty and health.

The UI
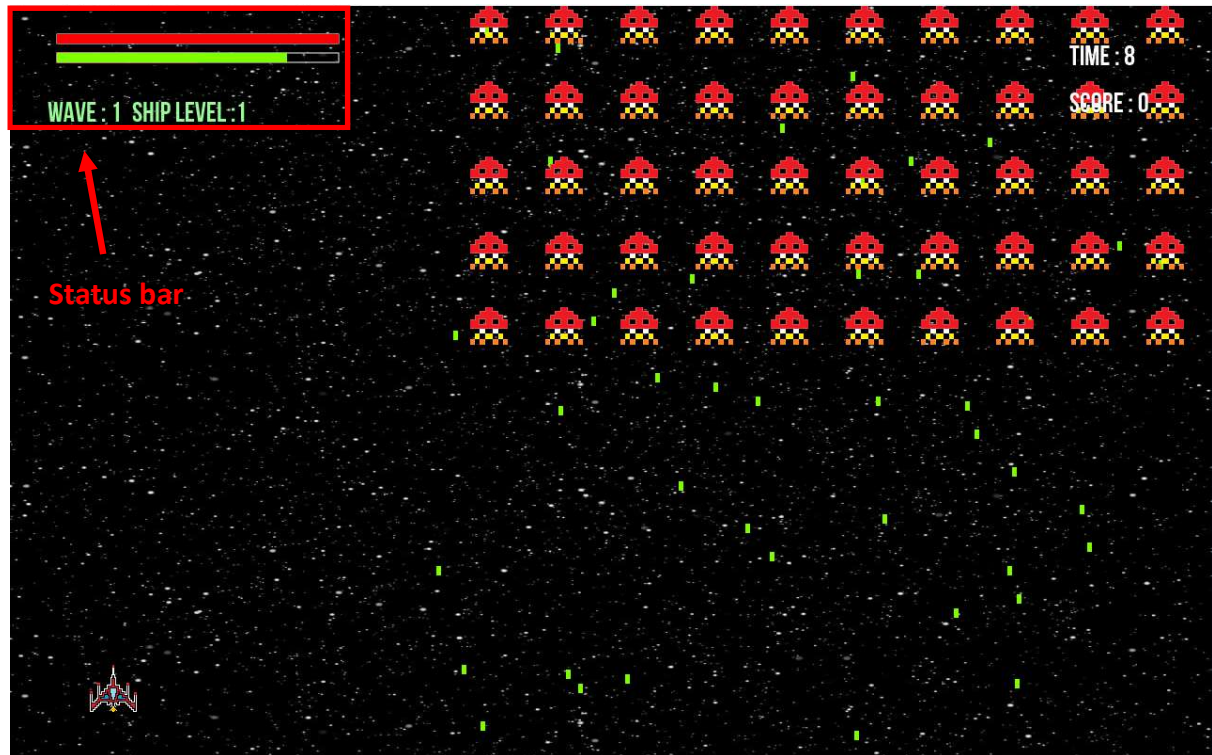
In status bar you can see your health and shield charge bar which will drain when you engage the shield deplete all that your shield will went off let you defend less from the swarm of enemy. On the status bar you can see what level is your ship and what wave are you in.
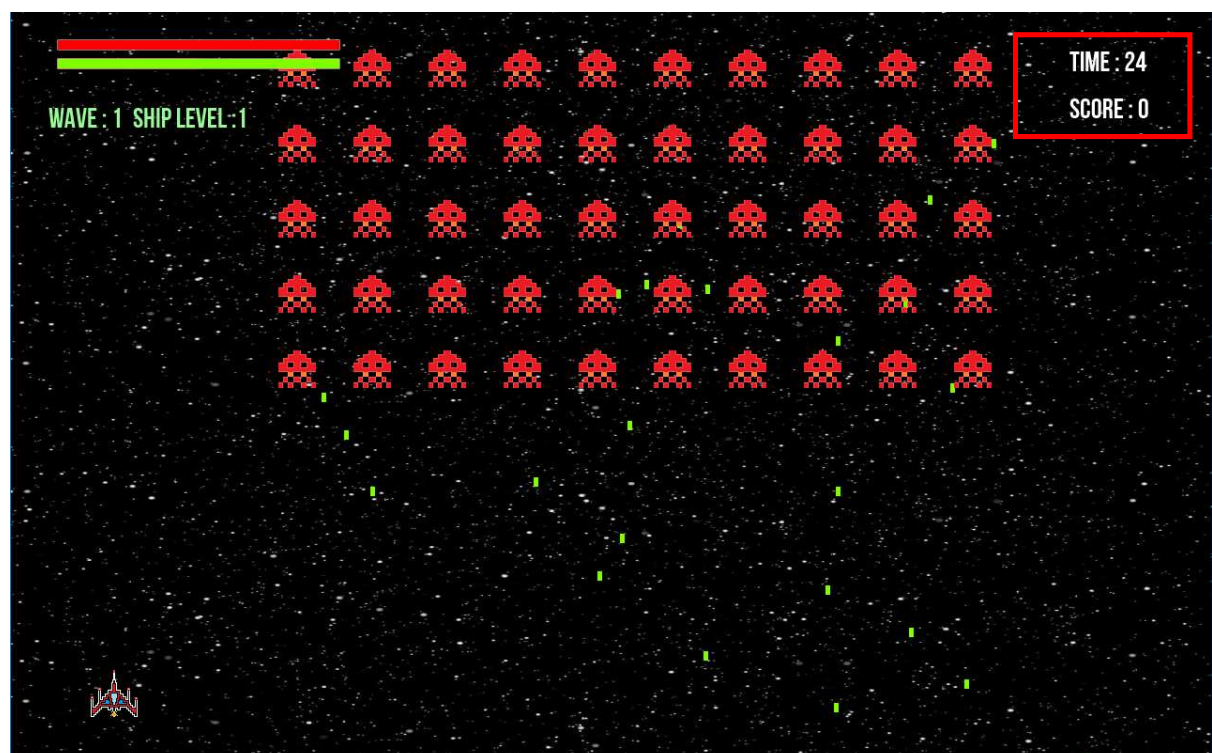
On the right you can see your time and score.
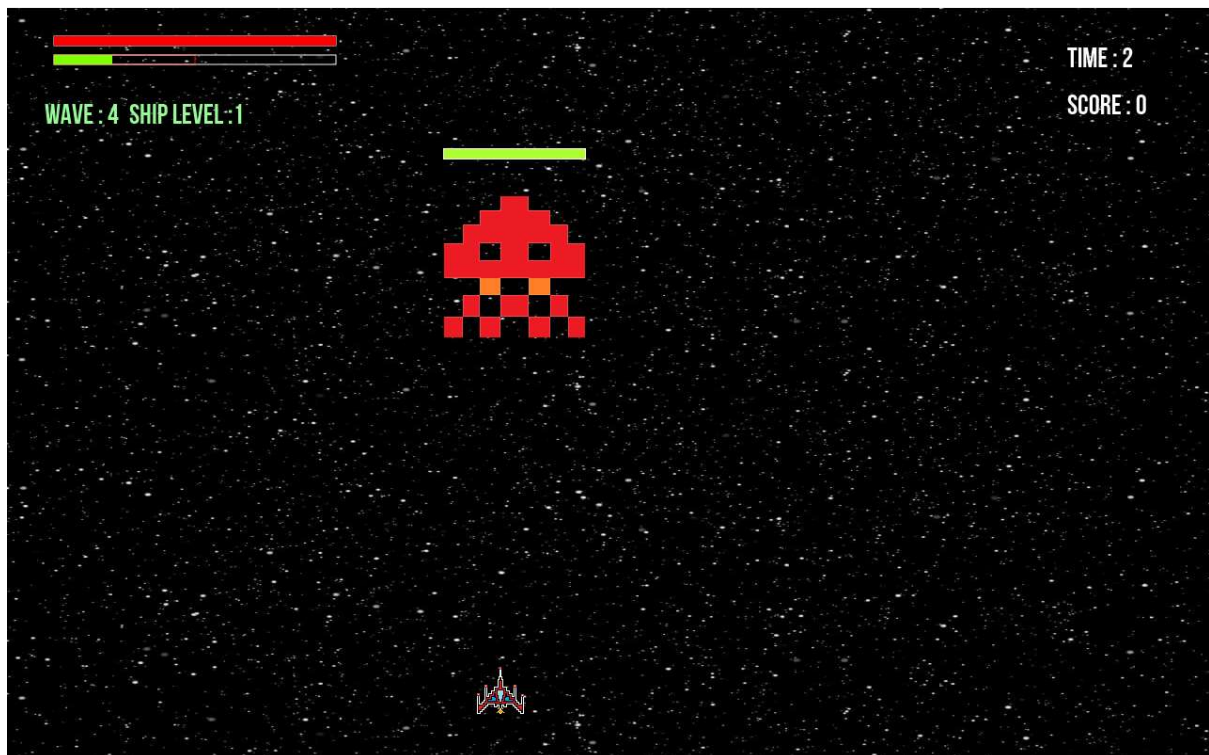
In Order to win the game, you must defeat the boss which has immense health and fire power taking him down is not an easy task, choose your tactics wisely to defeat the boss. There is a boss health bar above the boss itself to help you know how much it take to take him down.

Congratulations! Your score is 50100.

OK
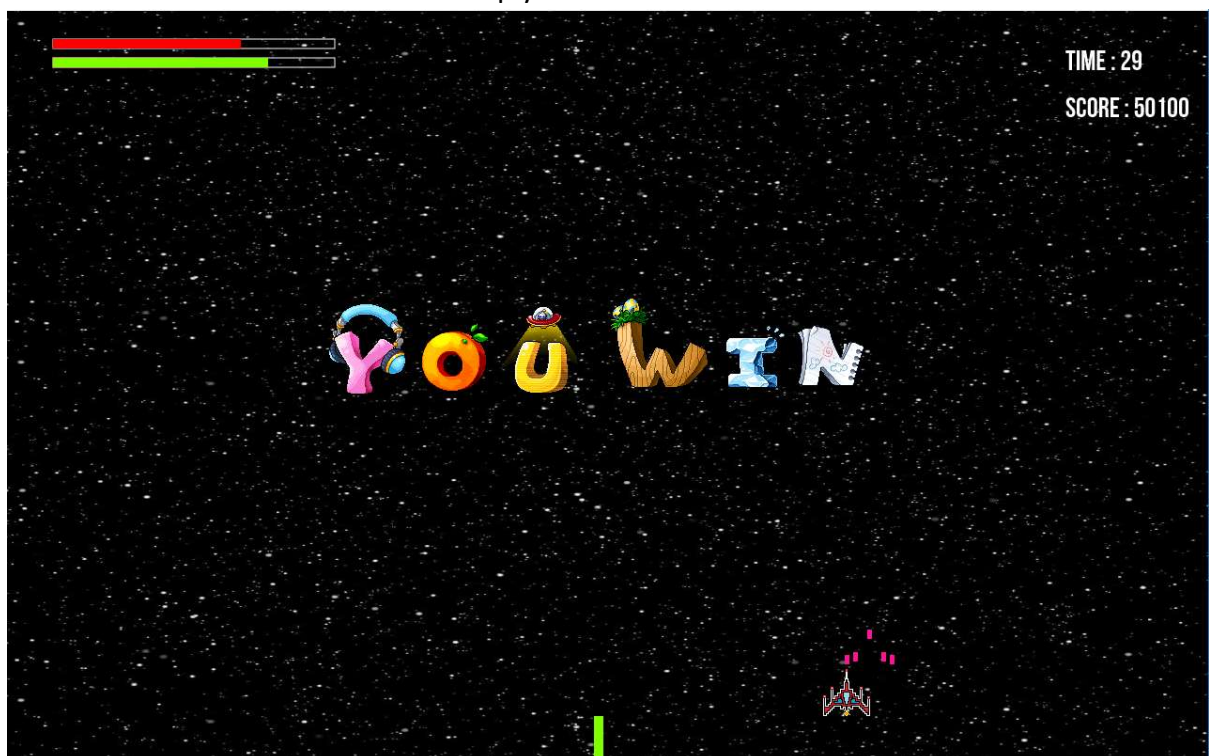
If you lucky enough to defeat the boss the following screen will appear for a moment and follow by a popup of your scores. Click OK and it will bring you back to main menu.

Implementation Detail

GameLogic
<<Java Class>>
logic

- gameObjectContainer: List<Entity>
- ship: Ship
- enemyArray: Enemy[][]
- enemycolSpace: int
- enemyrowSpace: int
- bulletCounter: int
- spawncounter: int
- enemyWave: int
- numberOfEnemy: int
- enemybodycount: int
- score: int
- isLogicRunning: boolean
- FPS: int
- LOOP_TIME: long
- gameLogic: Thread
- result: GameResult
- levelUpAnimation: LevelUpAnimation
- shipLevel: int
- startTime: int
- isStartCountTime: boolean
- isStartResultThread: boolean

- GameLogic(GameResult,LevelUpAnimation)
- spawnEnemy(int):void
- addNewObject(Entity):void
- startLogic():void
- stopGame():void
- logicUpdate():void
- shoot():void
- getScore():int
- clearGame():void
- updateLogic():void
- startCountTime():void
- countDown3():boolean

Canvas
<<Java Class>>
javafx.scene.canvas

- DEFAULT_VAL_BUF_SIZE: int
- DEFAULT_OBJ_BUF_SIZE: int
- SIZE_HISTORY: int
- current: GrowableDataBuffer
- rendererBehind: boolean
- recentvalsizes: int[]
- recentobjsizes: int[]
- lastsizeindex: int
- theContext: GraphicsContext
- width: DoubleProperty
- height: DoubleProperty

- Canvas()
- Canvas(double,double)
- max(int[],int):int
- getBuffer():GrowableDataBuffer
- isRendererFallingBehind():boolean
- getGraphicsContext2D():GraphicsContext
- setWidth(double):void
- getWidth():double
- widthProperty():DoubleProperty
- setHeight(double):void
- getHeight():double
- heightProperty():DoubleProperty
- impl_createPeer():NGNode
- impl_updatePeer():void
- impl_computeContains(double,double):boolean
- impl_computeGeomBounds(BaseBounds,BaseTransform):BaseBounds
- impl_processMXNode(MXNodeAlgorithm,MXNodeAlgorithmContext):Object
- access$000(Canvas,DirtyBits):void
- access$100(Canvas):void
- access$200(Canvas):GraphicsContext
- access$300(Canvas,DirtyBits):void
- access$400(Canvas):void

GameTime
<<Java Class>>
logic

- isTimeRunning: boolean
- seconds: int
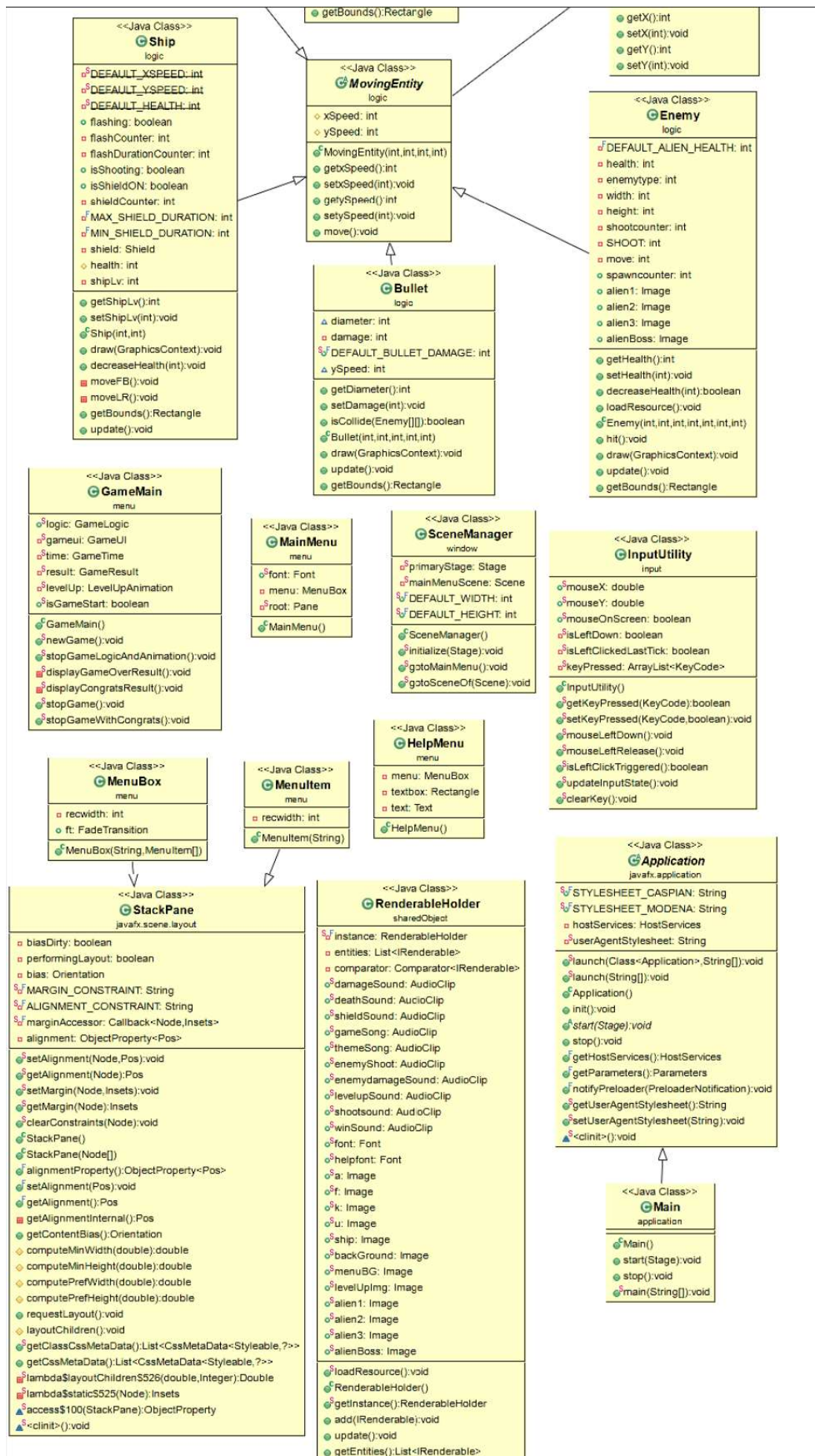- gameTime: Thread

- GameTime()
- stop():void

CreditMenu
<<Java Class>>
menu

- menu: MenuBox
- textbox: Rectangle

- CreditMenu()

GameResult
<<Java Class>>
drawing

- gameResult: Thread
- isResultRunning: boolean
- count: int

- GameResult()
- startResult():void
- stopResult():void
- updateResult():void
- drawI(GraphicsContext,double):void
- drawN(GraphicsContext,double):void
- drawO(GraphicsContext,double):void
- drawU(GraphicsContext,double):void
- drawW(GraphicsContext,double):void
- drawY(GraphicsContext,double):void

GameUI
<<Java Class>>
drawing

- FPS: int
- LOOP_TIME: long
- gameAnimation: Thread
- logic: GameLogic
- isAnimationRunning: boolean

- GameUI(GameLogic)
- addListerner():void
- paintComponent():void
- startAnimation():void
- stopAnimation():void

LevelUpAnimation
<<Java Class>>
drawing

- levelUpThread: Thread
- isLevelUpThreadRunning: boolean
- countX: int
- countY: int

- LevelUpAnimation()
- startLevelUp(Ship):void
- stopLevelUp():void
- updateLevelUp(Ship):void
- clearRect():void

IRenderable
<<Java Interface>>
sharedObject

- getZ():int
- draw(GraphicsContext):void
- isDestroyed():boolean
- isVisible():boolean

EnemyBullet
<<Java Class>>
logic

- diameter: int
- height: int
- DEFAULT_BULLET_DAMAGE: int

- EnemyBullet(int,int,int,int,int,int)
- isCollide(Ship):boolean
- draw(GraphicsContext):void
- update():void
- getBounds():Rectangle

Shield
<<Java Class>>
logic

- radius: int

- Shield(int,int)
- draw(GraphicsContext):void
- update():void
- getBounds():Rectangle

Entity
<<Java Class>>
logic

- x: int
- y: int
- z: int
- visible: boolean
- destroyed: boolean

- Entity()
- getBounds():Rectangle
- Entity(int,int)
- collideWith(Entity):boolean
- isDestroyed():boolean
- isVisible():boolean
- getZ():int
- getX():int
- setX(int):void
- getY():int
- setY(int):void

Ship
<<Java Class>>
logic

- DEFAULT_XSPEED: int

Ship
<<Java Class>>
Ⓖ Ship
logic
- ⬒ᔆDEFAULT_XSPEED: int
- ⬒ᔆDEFAULT_YSPEED: int
- ⬒ᔆDEFAULT_HEALTH: int
- ● flashing: boolean
- ■ flashCounter: int
- ■ flashDurationCounter: int
- ● isShooting: boolean
- ● isShieldON: boolean
- ■ shieldCounter: int
- ◨ᶠMAX_SHIELD_DURATION: int
- ◨ᶠMIN_SHIELD_DURATION: int
- ■ shield: Shield
- ◆ health: int
- ■ shipLv: int
- ● getShipLv():int
- ● setShipLv(int):void
- ● Ship(int,int)
- ● draw(GraphicsContext):void
- ● decreaseHealth(int):void
- ■ moveFB():void
- ■ moveLR():void
- ● getBounds():Rectangle
- ● update():void

MovingEntity
<<Java Class>>
Ⓖ MovingEntity
logic
- ◇ xSpeed: int
- ◇ ySpeed: int
- ᶠ MovingEntity(int,int,int,int)
- ● getxSpeed():int
- ● setxSpeed(int):void
- ● getySpeed():int
- ● setySpeed(int):void
- ● move():void

getBounds():Rectargle

Bullet
<<Java Class>>
Ⓖ Bullet
logic
- △ diameter: int
- ■ damage: int
- ᔆᶠDEFAULT_BULLET_DAMAGE: int
- △ ySpeed: int
- ● getDiameter():int
- ● setDamage(int):void
- ● isCollide(Enemy[][]):boolean
- ᶠ Bullet(int,int,int,int)
- ● draw(GraphicsContext):void
- ● update():void
- ● getBounds():Rectangle

getX():int
setX(int):void
getY():int
setY(int):void

Enemy
<<Java Class>>
Ⓖ Enemy
logic
- ◨ᶠDEFAULT_ALIEN_HEALTH: int
- ■ health: int
- ■ enemytype: int
- ■ width: int
- ■ height: int
- ■ shootcounter: int
- ■ SHOOT: int
- ■ move: int
- ■ spawncounter: int
- ● alien1: Image
- ● alien2: Image
- ● alien3: Image
- ● alienBoss: Image
- ● getHealth():int
- ● setHealth(int):void
- ● decreaseHealth(int):boolean
- ● loadResource():void
- ᶠ Enemy(int,int,int,int,int,int,int)
- ● hit():void
- ● draw(GraphicsContext):void
- ● update():void
- ● getBounds():Rectangle

GameMain
<<Java Class>>
Ⓖ GameMain
menu
- ○ᔆlogic: GameLogic
- ○ᔆgameui: GameUI
- ○ᔆtime: GameTime
- ○ᔆresult: GameResult
- ○ᔆlevelUp: LevelUpAnimation
- ○ᔆisGameStart: boolean
- ᔆᶠGameMain()
- ●ᔆnewGame():void
- ●ᔆstopGameLogicAndAnimation():void
- ■ᔆdisplayGameOverResult():void
- ■ᔆdisplayCongratsResult():void
- ●ᔆstopGame():void
- ●ᔆstopGameWithCongrats():void

MainMenu
<<Java Class>>
Ⓖ MainMenu
menu
- ○ᔆfont: Font
- ■ menu: MenuBox
- ○ᔆroot: Pane
- ᶠMainMenu()

SceneManager
<<Java Class>>
Ⓖ SceneManager
window
- ○ᔆprimaryStage: Stage
- ○ᔆmainMenuScene: Scene
- ᔆᶠDEFAULT_WIDTH: int
- ᔆᶠDEFAULT_HEIGHT: int
- ᶠSceneManager()
- ●ᔆinitialize(Stage):void
- ●ᔆgotoMainMenu():void
- ●ᔆgotoSceneOf(Scene):void

InputUtility
<<Java Class>>
Ⓖ InputUtility
input
- ○ᔆmouseX: double
- ○ᔆmouseY: double
- ○ᔆmouseOnScreen: boolean
- ○ᔆisLeftDown: boolean
- ○ᔆisLeftClickedLastTick: boolean
- ○ᔆkeyPressed: ArrayList<KeyCode>
- ᶠInputUtility()
- ●ᔆgetKeyPressed(KeyCode):boolean
- ●ᔆsetKeyPressed(KeyCode,boolean):void
- ●ᔆmouseLeftDown():void
- ●ᔆmouseLeftRelease():void
- ●ᔆisLeftClickTriggered():boolean
- ●ᔆupdateInputState():void
- ●ᔆclearKey():void

HelpMenu
<<Java Class>>
Ⓖ HelpMenu
menu
- ■ menu: MenuBox
- ■ textbox: Rectangle
- ■ text: Text
- ᶠHelpMenu()

MenuBox
<<Java Class>>
Ⓖ MenuBox
menu
- ■ recwidth: int
- ● ft: FadeTransition
- ᶠMenuBox(String,MenuItem[])

MenuItem
<<Java Class>>
Ⓖ MenuItem
menu
- ■ recwidth: int
- ᶠMenuItem(String)

StackPane
<<Java Class>>
Ⓖ StackPane
javafx.scene.layout
- ■ biasDirty: boolean
- ■ performingLayout: boolean
- ■ bias: Orientation
- ᔆᶠMARGIN_CONSTRAINT: String
- ᔆᶠALIGNMENT_CONSTRAINT: String
- ᔆᶠmarginAccessor: Callback<Node,Insets>
- ■ alignment: ObjectProperty<Pos>
- ●ᔆsetAlignment(Node,Pos):void
- ●ᔆgetAlignment(Node):Pos
- ●ᔆsetMargin(Node,Insets):void
- ●ᔆgetMargin(Node):Insets
- ●ᔆclearConstraints(Node):void
- ᶠStackPane()
- ᶠStackPane(Node[])
- ●ᶠalignmentProperty():ObjectProperty<Pos>
- ●ᶠsetAlignment(Pos):void
- ●ᶠgetAlignment():Pos
- ■ getAlignmentInternal():Pos
- ● getContentBias():Orientation
- ◇ computeMinWidth(double):double
- ◇ computeMinHeight(double):double
- ◇ computePrefWidth(double):double
- ◇ computePrefHeight(double):double
- ● requestLayout():void
- ● layoutChildren():void
- ●ᔆgetClassCssMetaData():List<CssMetaData<Styleable,?>>
- ● getCssMetaData():List<CssMetaData<Styleable,?>>
- ■ᔆlambda$layoutChildren$526(double,Integer):Double
- ■ᔆlambda$static$525(Node):Insets
- △ᔆaccess$100(StackPane):ObjectProperty
- △ᔆ<clinit>():void

RenderableHolder
<<Java Class>>
Ⓖ RenderableHolder
sharedObject
- ■ᔆinstance: RenderableHolder
- ■ entities: List<IRenderable>
- ■ comparator: Comparator<IRenderable>
- ○ᔆdamageSound: AudioClip
- ○ᔆdeathSound: AudioClip
- ○ᔆshieldSound: AudioClip
- ○ᔆgameSong: AudioClip
- ○ᔆthemeSong: AudioClip
- ○ᔆenemyShoot: AudioClip
- ○ᔆenemydamageSound: AudioClip
- ○ᔆlevelupSound: AudioClip
- ○ᔆshootsound: AudioClip
- ○ᔆwinSound: AudioClip
- ○ᔆfont: Font
- ○ᔆhelpfont: Font
- ○ᔆa: Image
- ○ᔆf: Image
- ○ᔆk: Image
- ○ᔆu: Image
- ○ᔆship: Image
- ○ᔆbackGround: Image
- ○ᔆmenuBG: Image
- ○ᔆlevelUpImg: Image
- ○ᔆalien1: Image
- ○ᔆalien2: Image
- ○ᔆalien3: Image
- ○ᔆalienBoss: Image
- ●ᔆloadResource():void
- ᶠRenderableHolder()
- ●ᔆgetInstance():RenderableHolder
- ● add(IRenderable):void
- ● update():void
- ● getEntities():List<IRenderable>

Application
<<Java Class>>
Ⓖ Application
javafx.application
- ᔆᶠSTYLESHEET_CASPIAN: String
- ᔆᶠSTYLESHEET_MODENA: String
- ■ hostServices: HostServices
- ○ᔆuserAgentStylesheet: String
- ●ᔆlaunch(Class<Application>,String[]):void
- ●ᔆlaunch(String[]):void
- ᶠApplication()
- ● init():void
- ● start(Stage):void
- ● stop():void
- ● getHostServices():HostServices
- ● getParameters():Parameters
- ■ notifyPreloader(PreloaderNotification):void
- ●ᔆgetUserAgentStylesheet():String
- ●ᔆsetUserAgentStylesheet(String):void
- △ᔆ<clinit>():void

Main
<<Java Class>>
Ⓖ Main
application
- ᶠMain()
- ● start(Stage):void
- ● stop():void
- ᶠmain(String[]):void

UML diagram part 2

1. Package main

1.1 Class Main extends Application

1.1.1 Method

+void start(Stage primaryStage)	The main entry point for the JavaFX applications.
	- Use SceneManager to initialize primaryStage.
	- Set title as "Cy3erPunk".
	- Set center on screen.
	- Set resizable to false.
	- Set on close request to stop game.
+void stop()	If game starts , stop game logic and game animation.
+void main(String[] args)	An entry point of the application.

2. Package drawing

2.1 Class GameResult extends Canvas

Figure 1 result when you win

2.1.1 Field

-Thread gameResult	The thread display game result animation when player wins.
-boolean isResultRunning	The flag indicate that the result animation is start and not end yet.
-int count	It used for sprite sheet.

2.1.2 Constructor

-GameResult()	• Initialize canvas. The canvas width and height is equals to SceneManager.DEFAULT_WIDTH and SceneManager .DEFAULT_HEIGHT respectively.
	• Set visible to true.

2.1.3 Method

+void startResult()	• Set isResultRunning to true.
	• Start gameResult thread by looping.
	• The loop will stop when isResultRunning is false.
	• For each loop, call updateResult().
+void stopResult()	Set isResultRunning to false.

+void updateResult()	Draw the result when player wins as shown in figure 1 • Draw each letters by calling "draw_(gc,x)" that gc is this GraphicsContext and x is position of letter that letter is located. • Increment count for animation in sprite sheet.
-void drawY(GraphicsContext gc, double x)	Draw alphabet "Y".
-void drawO(GraphicsContext gc, double x)	Draw alphabet "O".
-void drawU(GraphicsContext gc, double x)	Draw alphabet "U".
-void drawW(GraphicsContext gc, double x)	Draw alphabet "W".
-void drawI(GraphicsContext gc, double x)	Draw alphabet "I".
-void drawN(GraphicsContext gc, double x)	Draw alphabet "N".

2.2 Class GameUI extends Canvas
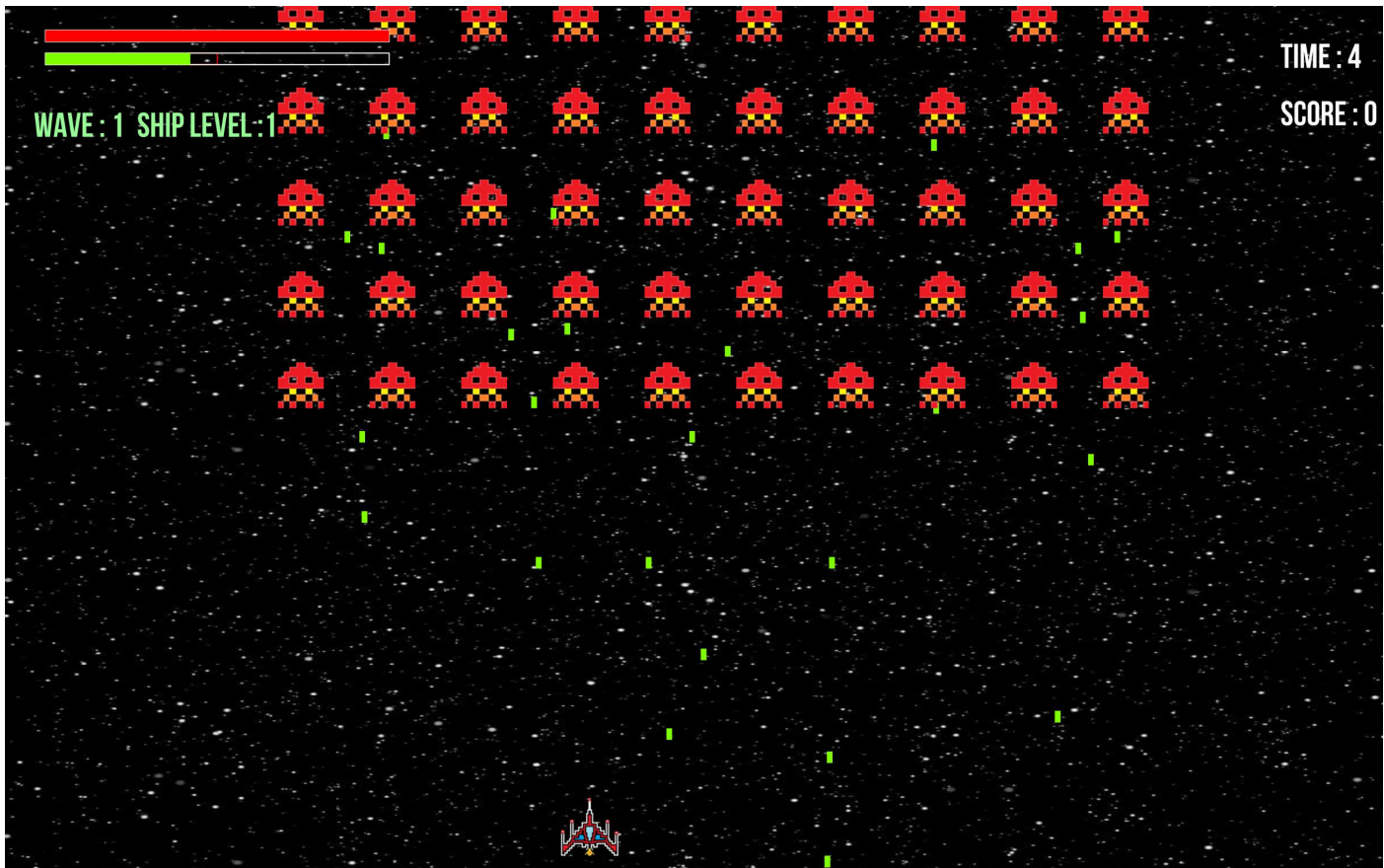
Figure 2 paintComponent

2.2.1 Field

-int FPS	Number of frame rates per second. Default is 60
-long LOOP_TIME	Time period between each update of a game animation
-Thread gameAnimation	A gameAnimation thread for draw every components
-GameLogic logic	A game logic
-boolean isAnimationRunning	The flag indicate that the game animation is start and not end yet.

2.2.2 Constructor

+GameUI(GameLogic logic)	• Initialize canvas. The canvas width and height is equals to DEFAULT_GAME_WIDTH and DEFAULT_GAME_HEIGHT respectively. • Set logic by the given parameter. • Set visible to true.

	• Add event handlers for this canvas by calling addListerner().

2.2.3 Method

+void addListerner()	Add event handler for keyboard and mouse
+void paintComponent()	Draw all components as shown in figure 2 • Draw background image by width and height is equals to DEFAULT_GAME_WIDTH and DEFAULT_GAME_HEIGHT respectively • Create list to get entities from RenderableHolder • Draw all entity that is not destroyed from list that is created and use synchronized. • Draw font ,time, score, wave and ship level. • If game end that player wins, all components - except time and score - will be deleted.
+void startAnimation()	• Set isGameAnimation to true. • Start gameAniamtion thread by looping. • The animation will update when the time has pass by a LOOP_TIME. • For each loop, call paintComponent() and use synchronized to instance from RenderableHolder. • The loop will stop when isResultRunning is false.
+void stopAnimation()	Set isAnimationRunning to false.

2.3 Class LevelUpAnimation extends Canvas
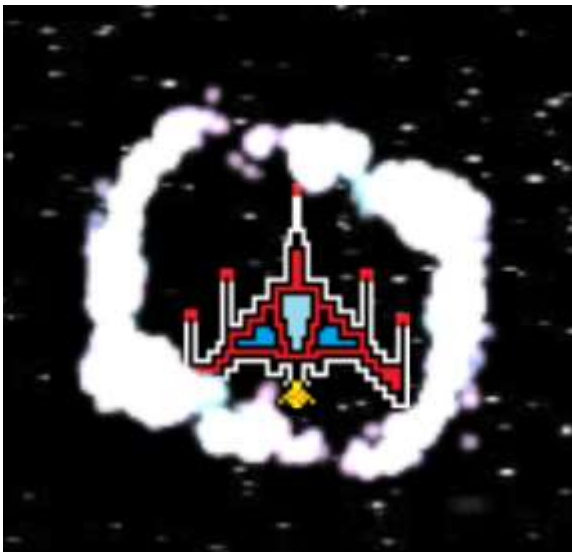
Figure 3 level up effect

2.3.1 Field

-Thread levelUpThread	A thread for animate level up effect of Ship
-boolean isLevelUpThreadRunning	The flag indicate that the level up animation is start and not end yet.
-int countX	Number is used for sprite sheet
-int countY	Number is used for sprite sheet

2.3.2 Constructor

+LevelUpAnimation()	• Initialize canvas. The canvas width and height is equals to SceneManager.DEFAULT_WIDTH and SceneManager .DEFAULT_HEIGHT respectively. • Set visible to true.

2.3.3 Method

+void startLevelUp(Ship ship)	- Set isLevelUpThreadRunning to true. - Start levelUp thread by looping. - The loop will stop when isLevelUpThreadRunning is false. - For each loop, call updateLevelUp(ship).
+void stopLevelUp()	- Set isLevelUpThreadRunning to false. - Call clearRect() for remove drawing when finish effect.
+void updateLevelUp(Ship ship)	Draw level up effect as shown in figure 3 - Always call clearRect for remove past drawing. - Draw level up effect by crop images from sprite sheet at ship position by the given parameter. - Increment countX and countY for change drawing in sprite sheet.
-void clearRect()	Clears a portion of the canvas with a transparent color value

3. Package input

3.1 Class InputUtility

3.1.1 Field

+double mouseX	Position x of mouse
+double mouseY	Position y of mouse
-boolean mouseOnScreen	Check whether mouse is on screen
-boolean isLeftDown	Check whether mouse is clicked
-boolean isLeftClickedLastTick	Check whether mouse is triggered
-ArrayList<KeyCode> keyPressed	A list contains key pressed

3.1.2 Method

+boolean getKeyPressed(KeyCode keycode)	Check whether keycode is pressed or not.
+void setKeyPressed(KeyCode keycode, boolean pressed)	Set keycode if it is pressed or not? If it is pressed, add keycode to keyPressed list Else , remove from keyPressed list
+void mouseLeftDown()	
+void mouseLeftRelease()	
+boolean isLeftClickTriggered()	Check is tricggered click
+void updateInputState()	Update input
+void clearKey()	Clear keycode from keyPressed list

4.Package logic

4.1 Class Bullet extends MovingEntity
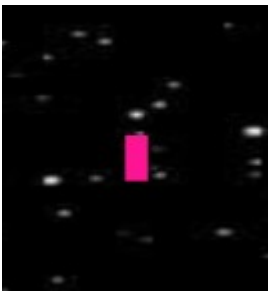
Figure 4 bullet

4.1.1 Field

-int diameter	Diameter of the bullet
-damage	Damage of the bullet
+DEFAULT_BULLET_DAMAGE	Default damage of the bullet
-ySpeed	Y-axis speed of the bullet

4.1.2 Consturctor

+Bullet(int x,int y,int xSpeed,int ySpeed,int diameter)	Initialize the element some may use parent constructor.

4.1.3 Method

+int getDiameter()	Diameter getter
+void setDamage()	Damage setter
+isCollide(Enemy[][] enemywave)	Check collide from enemywave using collidewith.Play enemeydamagesound when hit and set bullet to destroy when hit.
+void draw(GraohicsContext gc)	Draw rectangle as a bullet
+void update()	Destroy the bullet that gone off the screen to save memory.
+Rectangle getBounds()	Return rectangle as a hitbox.

4.2 Class Enemy extends MovingEnity

Figure 5 enemy

4.2.1 Field

-Int DEFAULT_ALIEN_HEALTH	Set default alien health to 1
-int health	Health of the alien
-int enemytype	Type of the alien
-int shootcounter	Counter that will certain point will cause enemy to shoot.
-int SHOOT	A number that shootcounter must aquire to shoot.
-int move	The number of alien have to move down when reach the end of the screen.
+Image alien1	Contain a picture of the alien from the render holder
+Image alien2	Contain a picture of the alien from the render holder
+Image alien3	Contain a picture of the alien from the render holder
+Image alienBoss	Contain a picture of the alienBoss from the render holder

4.2.2 Constructor

+Enemy(int x,int ,int xSpeed, int ySpeed,int enmeytype,int width,int height)	Initialize all the element.Random number for SHOOT to allow alien to shoot randomly

	Boss has extra health

4.2.3 Method

+int getHealth()	Health getter
+void setHealth(int health)	Health setter
+Boolean decreaseHealth(int damage)	Decrease health and destroy the alien when health reach 0.
+void loadResource()	Initialize load all image of the alien from RenderHolder.
+void hit()	Decreasehealth and play damage sound when ship hit the alien.
+ void draw(GraphicContext gc)	Draw each alien from enemy type
+ void update()	add Enemy bullet to gamelogic when shootcounter%SHOOT=0 and play enemyshootsound.Boss has different type of bullet check alien reach the end of the window shift down and xSpeed=-xSpeed.Boss will never got down through the middle of the screen.
+Rectangle getBounds()	Get hit box for the alien.

4.3 Class EnemyBullet extends MovingEntity
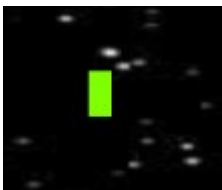
Figure 6 bullet

4.3.1 Field

-int diameter	Diameter of the enemyBullet
-int height	Determine the hight of the bullet
+int DEFAULT_BULLET_DAMAGE	Default enemyBullet damage =1

4.3.2 Constructor

+EnemyBullet(int x,int y,int xSpeed,int ySpeed,int diameter,int height)	Initialize following element

4.3.3 Method

+Boolean isCollide(Ship ship)	check if enemyBullet collideWith ship,decrease ship health and pay damage sound then set destroy=true.
+void draw(GarphicContext gc)	Draw bullet using fill rect
+void update()	Check if bullet gone off the window set destroy to true.
+Rectangle getBounds()	Return rectangle as a hitbox

4.4 Abstract class Entity implements IRenderable

4.4.1 Field

#int x	Position x of the entity
#int y	Position y of the entity

#int z	Priority of the entity
#boolean visible	Visible
#destroy	State of the entity

4.4.2 Constructor

+Entity()	
+Entity(int x,int y)	Set x,y as the given parameter Set visible true,destroy false

4.4.3 Method

+Rectangle getBounds()	Hitbox
+Boolean isDestory()	Getter for destroy
+Boolean isVisible()	Getter for visible
+int getZ()	Getter for z
+int getX()	Getter for x
+void setX(int x)	Setter for x
+int getY()	Getter for y
+void setY(int y)	Setter for y

4.5 Class GameLogic

4.5.1 Field

-List<Entity> gameObjectContainer	A list contains all object of entity
-Ship ship	A ship of game
-Enemy[][] enemyArray	Array 2 dimensions of enemy
-int enemycolSpace	Space of enemy apart from each others vertically.
-int enemyrowSpace	Space of enemy apart from each others horizontally.
-int bulletCounter	Count number of bullet
+int spawncounter	Diversify enemy shooting pace
+int enemyWave	The number wave in game
+int numberOfEnemy	The number of enemy
+int enemybodycount	Count all enemy entire game
+int score	The score of the game
-boolean isLogicRunning	The flag indicate that the logic thread is start and not end yet.
-int FPS	Number of frame rates per second. Default is 60
-long LOOP_TIME	Time period between each update of a game animation
-Thread gameLogic	A thread for update logic
-GameResult result	The game result when you win
-LevelUpAnimation levelUpAnimation	The effect animation when ship is level up
-int shipLevel	The ship level
-int startTime	Start counting time for function countDown3()
-boolean isStartCountTime	Check whether start time is counted
-boolean isStartResultThread	Check whether logic thread is started

4.5.2 Constructor

+GameLogic(GameResult result,LevelUpAnimation levelUpAnimation)	Initialize gameLogic - Set result by the given parameter - Set level up animation by the given parameter - Set score to zero - Set enemybodycount to zero - Set enemyWave to zero - Set bulletCounter to zero - Initialize gameObjectContainer with ArrayList - Initialize ship

	- Call spawnEnemy with enemywave as the parameter - Call addNewObject for add ship to gameObjectContainer list

4.5.3 Method

+void spawnEnemy(int wavenumber)	Spawn enemy - Initialize enemyArray (row is 5, col is 10) - Initialize new enemy for each row and column to enemyArray and call addNewObject with enemy as the parameter - If wave number is 4 , initialize enemy only one (it is a boss enemy) and call addNewObject with enemy as the parameter
#void addNewObject(Entity entity)	- Add new entity to gameObjectContainer list - Add new entity to RenderableHolder instance
+void startLogic()	Start logic thread - Set isLogicRunning to true. - Start logic thread by looping. - The logic will update when the time has pass by a LOOP_TIME. - For each loop, call updateLogic() and use synchronized to instance from RenderableHolder. - The loop will stop when isResultRunning is false.
+void stopGame()	Stop logic thread - Set isLogicRunning to false - Call clearGame()

+void logicUpdate()	Update the logic
	- Set ship level by every 1000 score points , ship will level up once
	- If ship level up then play level up sound and start levelUpAnimation thread
	- If all monster in that wave died, increment enemyWave and spawn enemy again then set enemybodycount to zero
	- If enemy wave Is 5, start result thread and stop game song then play win song, after that, count down 3 seconds and show pop up box with congrats and stop game
	- Reload bullet counter with increase by 1
	- Update the ship
	- If ship is shooting, call shoot() then set isShooting to false
	- Check all entities in gameObjectContainer
	  - if that entity is enemy, check whether ship is collide with enemy. If it collide ,then decrease ship health and update a enemy
	  - if that entity is bullet, check whether bullet is collide with enemy.If it collide, then remove that bullet
	  - if that entity is enemy bullet , check whether it collide with ship. If it collide, then remove that bullet
	  - else remove other entity that is destroyed

-void shoot()	Ship can shoot bullet - Every bullet counter is 0 or 20 , ship will ship shoot one time for limit excessive bullet - If ship level is more than one , initialize one bullet - If ship level is more than five , initialize more two bullets - If ship level is more than ten , initialize more two bullets - Set bullet counter to zero for reload the next shooting
+int getScore()	Get score

+void clearGame()	- Stop game song - Clear gameObjectContainer - Clear entities in RenderableHolder - Clear key in InputUtility
-void updateLogic	Update logic - Update logic by call logicUpdate() - Update instance in RenderableHolder - Update input state in InputUtility
+void startCountTime()	Count start time
-boolean countDown3()	Count down 3 seconds

4.6 Class GameTime

Figure 7 time

4.6.1 Field

-boolean isTimeRunning	The flag indicate that the gameTime thread is start and not end yet.
+int seconds	Time in second
-Thread gameTime	A thread for gameTime

4.6.2 Constructor

+GameTime()	Initialize GameTime - Set isTimeRunning to true - Start game time thread by looping - Count time

4.7 Abstract Class MovingEntity extends Entity

4.7.1 Field

#int xSpeed	X-axis speed
#int ySpeed	Y-axis speed

4.7.2 Constructor

+MovingEntity(int x,int y ,int xSpeed,int ySpeed)	Initialize parent constructor Initialize the following parameter

4.7.3 Method

+Getter setter for xSpeed and ySpeed	
+void move()	Change x and y with xSpeed and ySpeed

4.8 Class Shield extends Entity

4.8.1 Field

#int radius	Radius of a shield

4.8.2 Constructor

+Shield(int x,int y)	Use parent constructor
	- Set radius to 75
	- Set z=1000

4.8.3 Method

+void draw(GraphicContext gc)	Draw circle as a shield
+void update()	
+Rectangle getBounds()	Return a square as a hitbox

4.9 Class Ship extends MovingEntity

Figure 8 ship

4.9.1 Field

+int DEFAULT_XSPEED	Default xSpeed of the ship
+int DEFAULT_ySPEED	Default ySpeed of the ship
+int DEFAULT_HEALTH	Default health of the ship (3)
+Boolean flashing	State of flashing
-int flashCounter	For counting flashing time
-int flashDurationCounter	For duration of flashing
+Boolean isShooting	Is space key is pressed
+Boolean isShieldON	Status of a shield
-int shieldCounter	Shield charging status
-int MAX_SHIELD_DURATION	The max shield duration of the shield
-int MIN_SHIELD_DURATION	The min shield charge that will be able to engage shield
-Shield shield	Shield
#int health	Health of a ship
-int shipLv	Level of the ship

4.9.2 Costructor

+Ship(int x,int y)	Use parent contructor
	- Set health to default
	- Set z to 1000000
	- Set Ship Lv to1

4.9.3 Method

Getter and setter for shipLv	
+void draw(GraphicContext gc)	- draw health and shield bar
	- if shield ON draw shield
+void decreaseHealth(int damage)	Decrease health and set flashing to true play damagesound and will not take damage while flashing but if shield is on ON state will not take any damage but shield charge will decrease

-void moveFB()	Move forward or backward but will now surpass the window size
-void moveLR()	Move left and right but will not surpass the window size
+Rectangle getBounds()	If shieldON return shied hitbox if now will return normal ship hit box
+void update()	Check if shield on draw shield and check if shield counter is enough to engage the shield shield counter will draining while shield is on and will gone up if off check if ship is destroy ,the game will end check if flashing and create flashing while true. Check if pressed space isShooting = true check if wasd is pressed ship will move forward,left,backward,and right.if esc is pressed the game will end.

5. Package Menu

5.1 Class CreditMenu

5.1.1 Field

-MenuBox menu	Create menubox
-Rectangle textbox	For writing credit

5.1.2 Constructor

+CreditMenu()	- Initilize pane and scene - Set background image - Add text and font as seen above

5.2 Class GameMain

5.2.1 Field

+GameLogic logic	A game logic
-GameUI gameui	A game ui
-GameTime time	A game time
-GameResult result	A result game at the end if you win
-LevelUpAnimation levelUp	A level up effect
+boolean isGameStart	The flag indicate that game is start

5.2.2 Method

+void newGame()	Start new game - Set isGameStart to true - Initialize root as stack pane - Set scene with root - Initialize result, levelUp, logic, gameui, time Then add all to root - Request focus to gameui - Start gameui animation - Start logic - Play game song - Call gotoSceneof(scene)
+void stopGameLogicAndAnimation()	Stop game - Set isGameStart to false - Stop animation - Stop logic - Stop time - Stop result if you win at the end
-void displayGameOverResult()	Display pop up with game over text and go to main menu
-void displayCongratsResult()	Display pop up with congrats text and go to main menu
+void stopGame()	Call stopGameLogicAndAnimation() Call displayGameOverResult() to display later
+void stopGameWithCongrats()	Call stopGameLogicAndAnimation() Call displayCongratsResult() to display later

5.3 Class HelpMenu
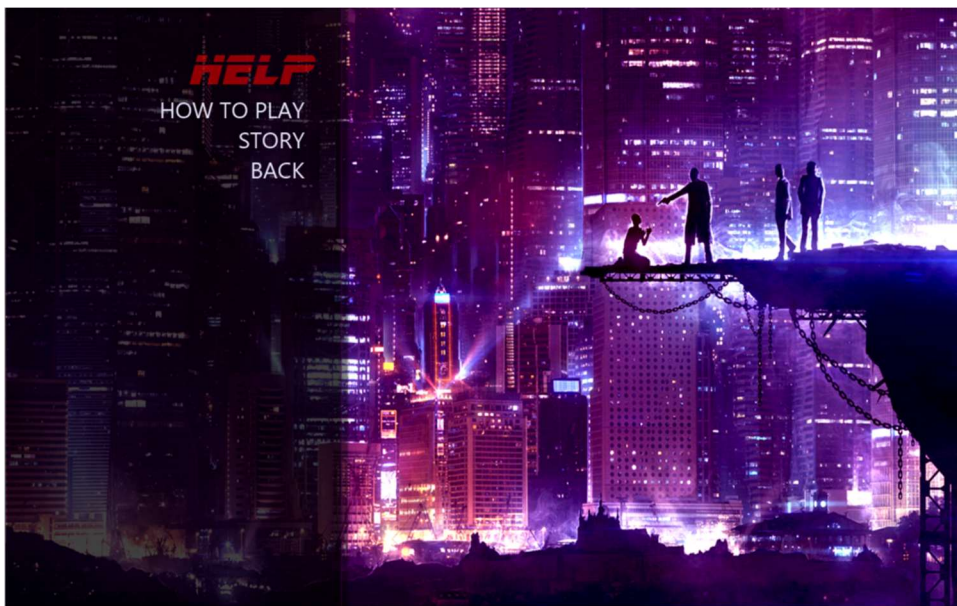
5.3.1 Field

+MenuBox menu	Create menubox
+Rectangle textbox	Rectangle to contain text
+Text text	Title of the text
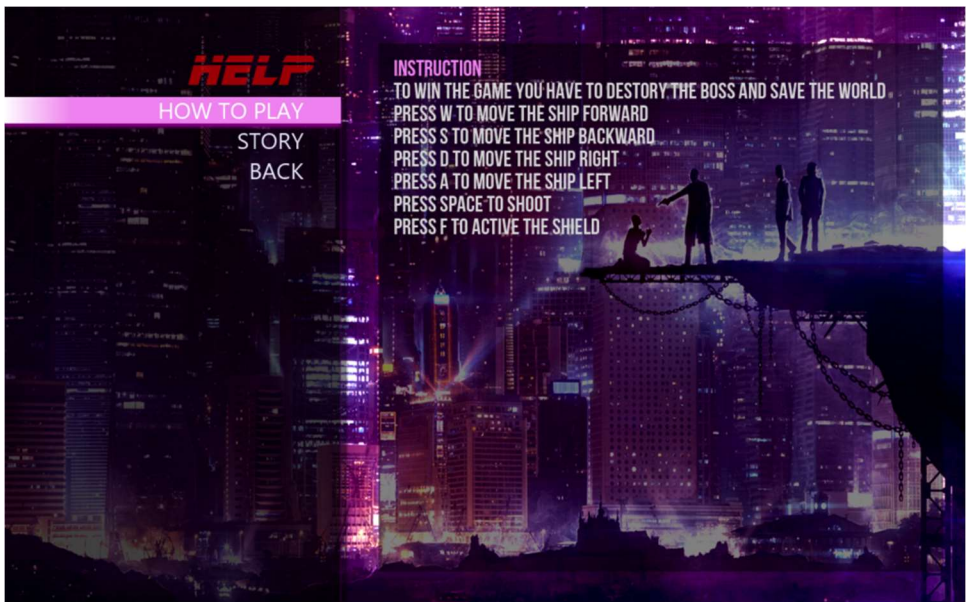
5.3.2 Constructor

+HelpMenu()	• Initilize pane and scene • Set background image • Add menuitem (howtoplay,story,back)to menu When click on mainmenu will create rectangle with opacity set to 0.5 and contain text for each menu

5.4 Class MainMenu extends scene

5.4.1 Field

+Font font	Contain font from RenderHolder
-Menubox menu	Create varaible for menuBox
-Pane root	A pane

5.4.2 Constructor

+MainMenu()	• Initialize using super set max and min size with number provided in scenemanager • Add background set fit to the window size. • Initilize quit funtion as a menuitem • Initilize helpmenu as a menuitem • Initilize credit as a menuitem • Initilize Start game as a menuItem and add eventhandler to intilize GameMain and stop themesong

5.5 Class MenuBox extends StackPane
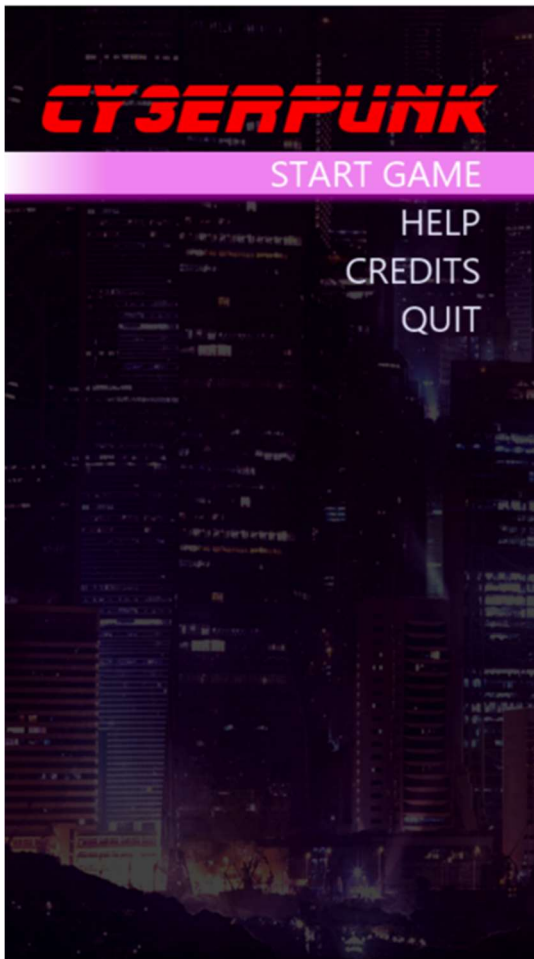
Item box will contain game title which will blink using fade trasition

5.5.1 Field

-int rectwidth	• SceneManager.DEFAULT_WIDTH * 0.35

5.5.2 Constructor

+MenuBox(String title,Menu... items)	Construct a menu box to contains multiple menuitem • Create rectangle with a width of rectwidth and hight of a window with drop shadow set spread(0.8). • Create new text for title form provided font set fade transition to 750 mills

| | • Create VBox |
| | • Add all above to VBox |

5.6 Class MenuItem extends StackPane
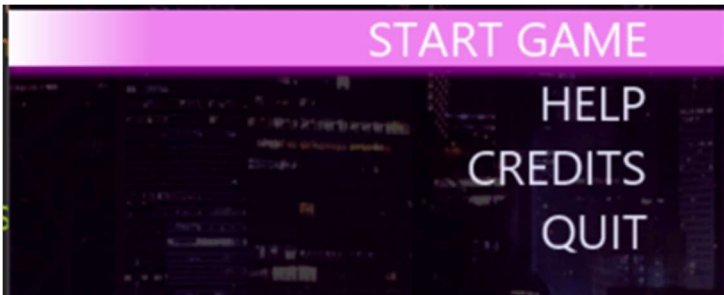
      Each menu item will change it colour to lavender with a white gradient when hovering mouse on it and change it colour to white while pressed mouse button.

-int rectwidth	• SceneManager.DEFAULT_WIDTH * 0.35
+MenuItem(String name)	Construct a menu item with a given name on it • Using Rectangle with gradient wide as a rectwidth • Set dropshadow Add eventhandler • Change rectangle to lavenderand text to white when hovering mouse above

6. Package sharedObject

6.1 Interface IRenderable

6.1.1 Method

+int getZ()	Get Z to sort
+void draw(GraphicsContext gc)	Draw canvas
+boolean isDestroyed()	The flag whether is destroyed
+boolean isVisible()	The flag whether is visible

6.2 Class IRenderableHolder

6.2.1 Field

-RenderableHolder instance	A singleton
-List<IRenderable> entities	A list contains all entities
-Comparator<IRenderable> comparator	A comparator to sort which entities is drawn first
+AudioClip damageSound	Damage sound
+AudioClip deathSound	Death sound
+AudioClip shieldSound;	Shield sound
+AudioClip gameSong;	Game song
+AudioClip themeSong;	Theme game song
+AudioClip enemyShoot;	Enemy shoot sound
+AudioClip enemydamageSound;	Hit enemy sound
+AudioClip levelupSound;	Level up sound
+AudioClip shootsound;	Bullet sound

+AudioClip winSound;	Win sound
+Font font;	Font type 1
+Font helpfont;	Font type 2
+Image a;	Image sprite
+Image f;	Image sprite
+Image k;	Image sprite
+Image u;	Image sprite
+Image ship;	Image sprite ship
+Image backGround;	Image sprite for game background
+Image menuBG;	Image sprite for menu background
+Image levelUpImg;	Image sprite for level up effect
+Image alien1;	Image sprite for alien type 1
+Image alien2;	Image sprite for alien type 2
+Image alien3;	Image sprite for alien type 3
+Image alienBoss;	Image sprite for boss

6.2.2 Constructor

+RenderableHolder()	- Initialize entities - Initialize comparator how to sort

6.2.3 Method

+void loadResource()	Load game resources
+RenderableHolder getInstance()	Get instance
+void add(IRenderable entity)	Add entity to entities and sort
+void update()	Update in entities - If entities is destroyed, remove it
+List<IRenderable> getEntities()	Get entities

7. Package window

7.1 Class SceneManager

7.1.1 Field

-Stage primaryStage	JavaFX Container
-Scene mainMenuScene	A main menu canvas
+int DEFAULT_WIDTH	Width of the application scene
+int DEFAULT_HEIGHT	Height of the application scene

7.1.2 Method

+void initialize(Stage stage)	Attempts to show this window by setting visibility to true
+ void gotoMainMenu()	Set a primaryStage's scene to mainMenuScene. Play theme game song
+ void gotoSceneOf(Canvas canvas)	Set a primaryStage's scene to the scene that containing only the given canvas as a child