**Zenith Chronicle**

**Documentation**

**Presented By**

Sirapoom Leesuravanich 6330539421

Panithi Khamwangyang 6330301321

**2110215 Programming Methodology**

**Semester 1 Year 2021**

**Chulalongkorn University**

**Zenith Chronicle – RPG hack&slash 2D-platformer Game**

**Introduction**

Zenith Chronicle is a game where player take the role of a treasure hunter to find the lost artifacts that disappeared during the old war of the old gods. With various types of artifacts and power ups found on the way, players have to make their ways to the end of the journey and collect all of the artifacts.

**Main Character**

This is a character and only character   
that player can control. He always spawn  
with 100 HP.

**Powerup**

Powerups are items found on the ground. Each one has  
their own effects and is stackable if one has multiples.  
There are 4 types of powerup.



Red potion : instantly recover some health points for player.

Apple : give player a small attack buff.

Blue fish : give player a small movement buff.

Red flower : give player a small jump height buff.

**Artifact**

By slaying enemies, player will occasionally drop artifacts which will have its own effects while holding and are also stackable. There are 6 types of artifacts.



Armor : give boosts to player’s max health points.

Sword : give boosts to player’s attack damage.

Boots : give boosts to player’s movement speed.

Ring : give boosts to player’s jump height.

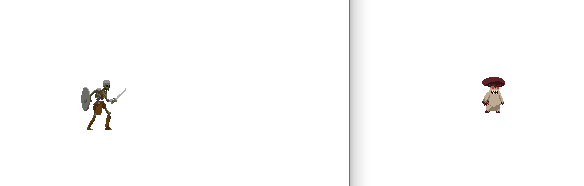
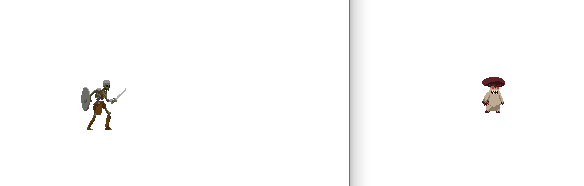
Necklace : give boosts to player’s dash speed.

Shield : give boosts to player’s health point regeneration.

**Enemy**

There are two types of enemies in this game, which are boss and monster. Boss will be the only enemy in boss stage.   
 The enemies’ strength are based on difficulty and they get stronger in every stage.

**Monster**



Monster is a type of enemy and only contain 2 subtypes : Skeleton and Mushroom. They gain HP, attack points and movement speed as the game progress.

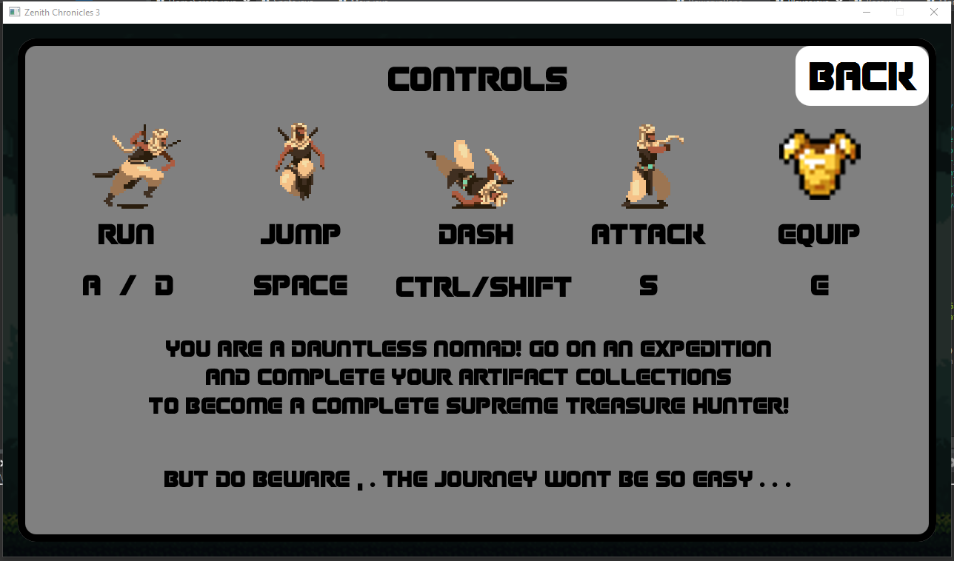
**Boss**

Boss is a type of enemy and this game has only one boss. He gains HP, attack points and attack speed as the game progress.

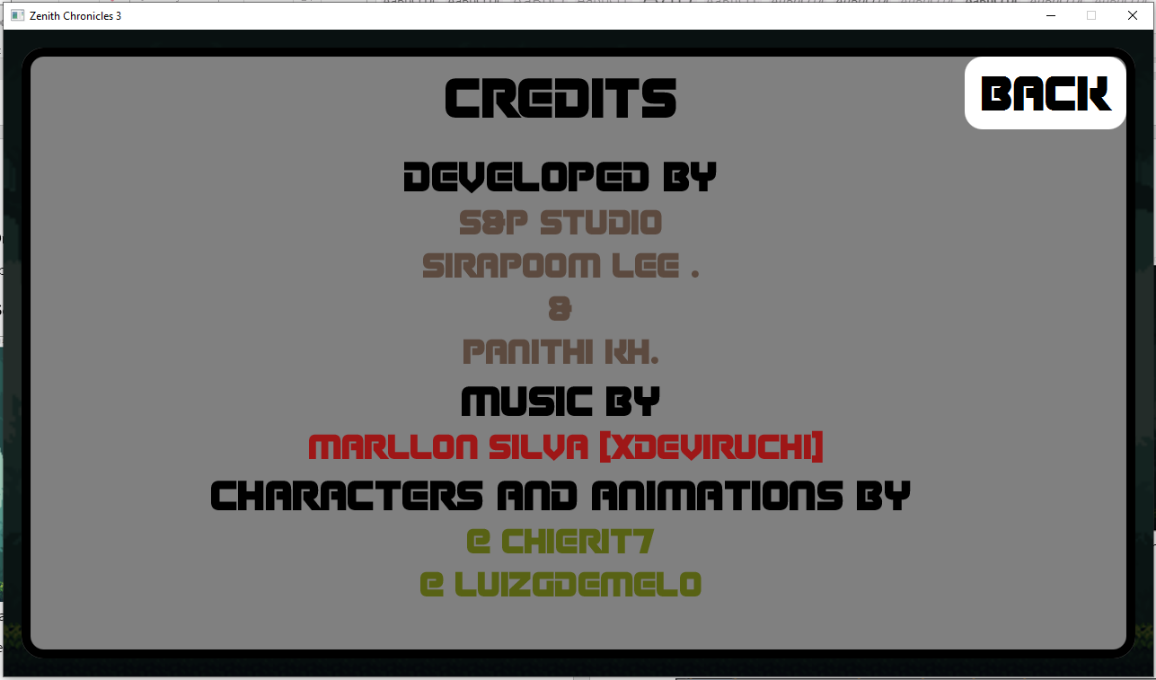
**Gameplay & Scene**

**Main Menu**

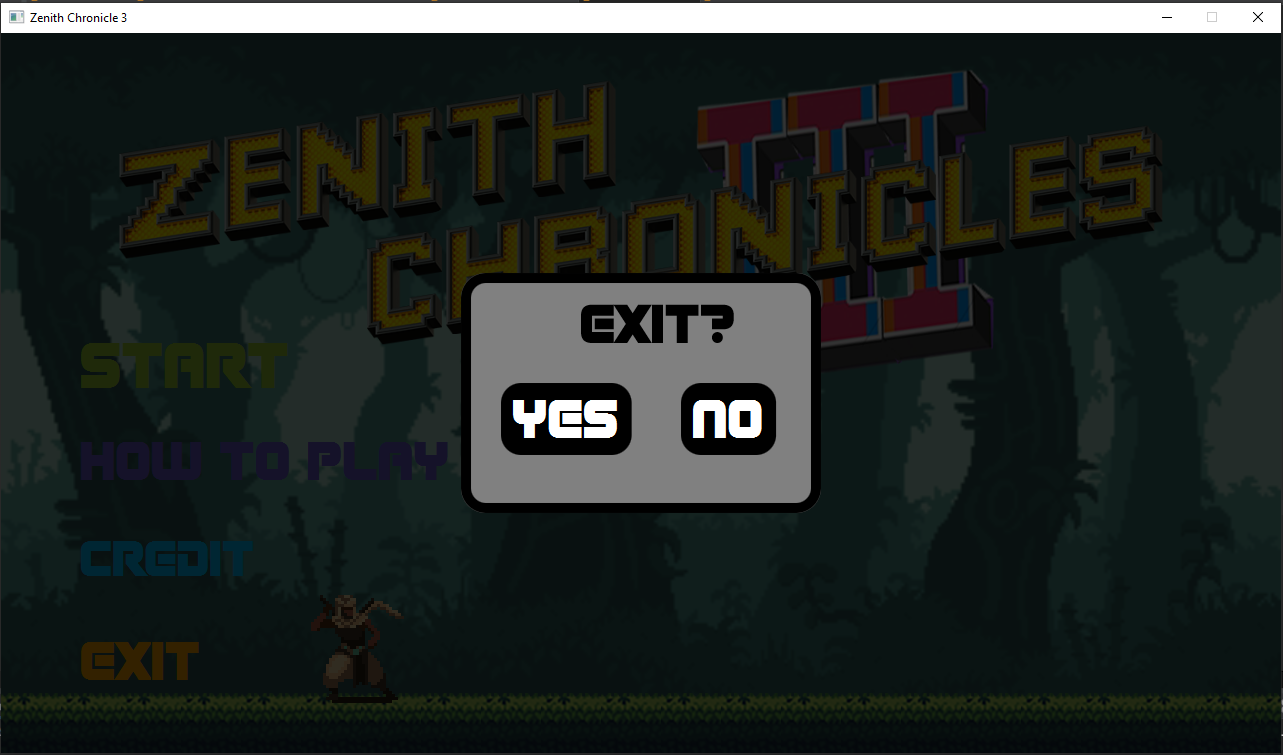
When the player open the game, they will see Main Menu. This scene has 4 buttons. Player can click “Start” to show the select game difficulty scene, “How to play”, to show how to play the game, “Credit”, to show credit, and “Quit”, which will end the game.

**How-to-play Scene**

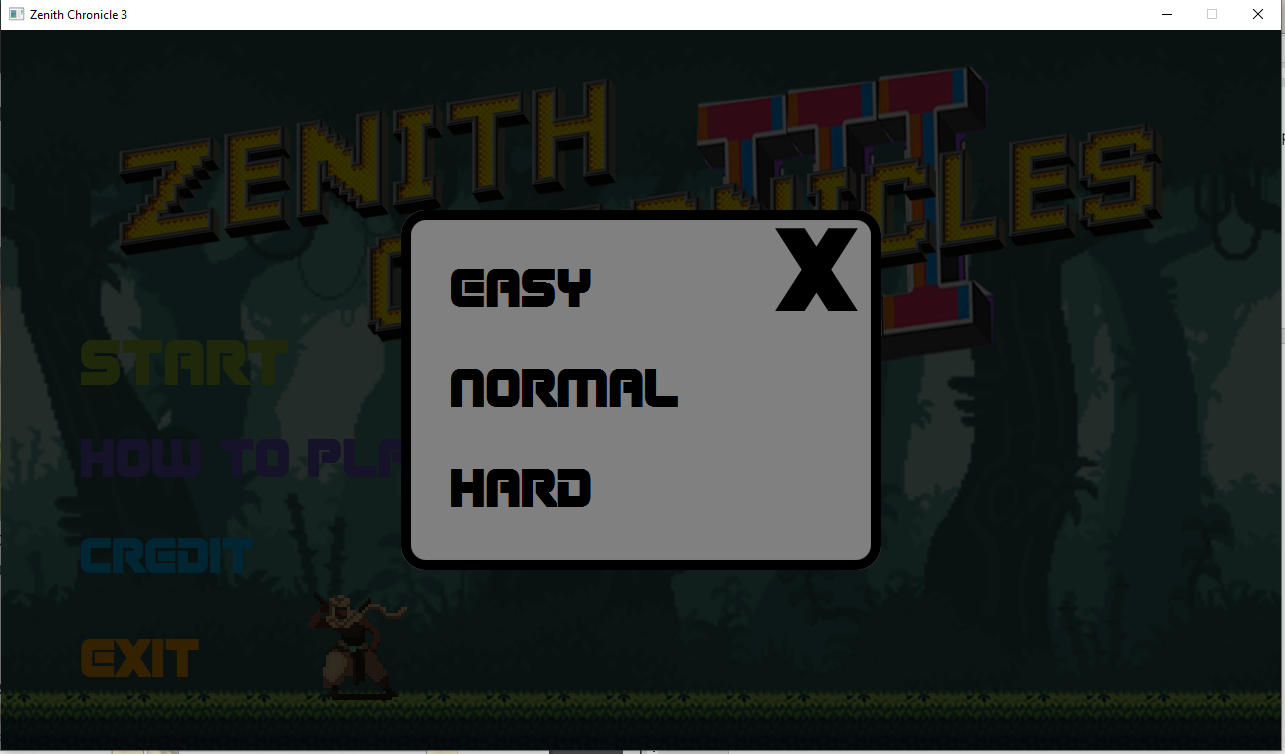
Only one option player can do is to go back to the Main Menu by press the “Back” button or press esc.

**Credit Scene**

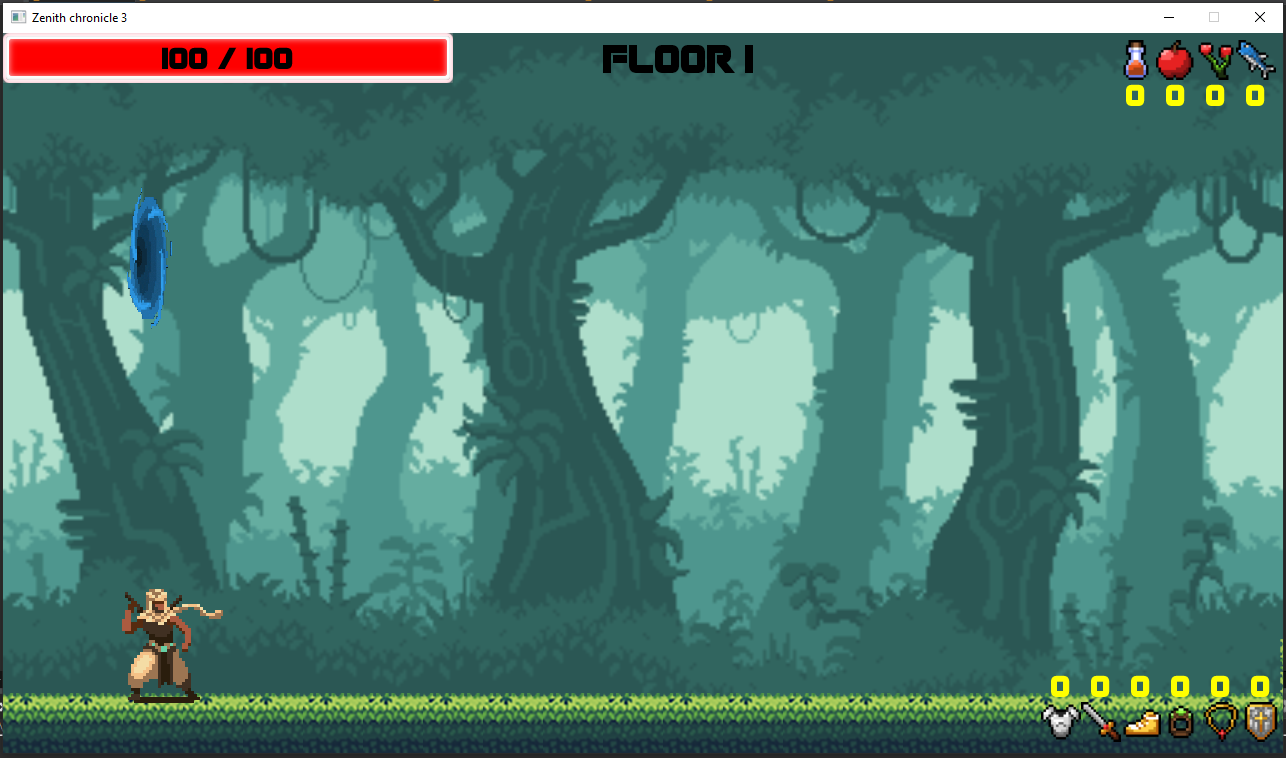
Same as the How-to-play scene, player have only one option to go back to Main Menu by press the “Back” button or press esc.

**Exit scene**

Player have two options to end the game or to go back to the main menu.

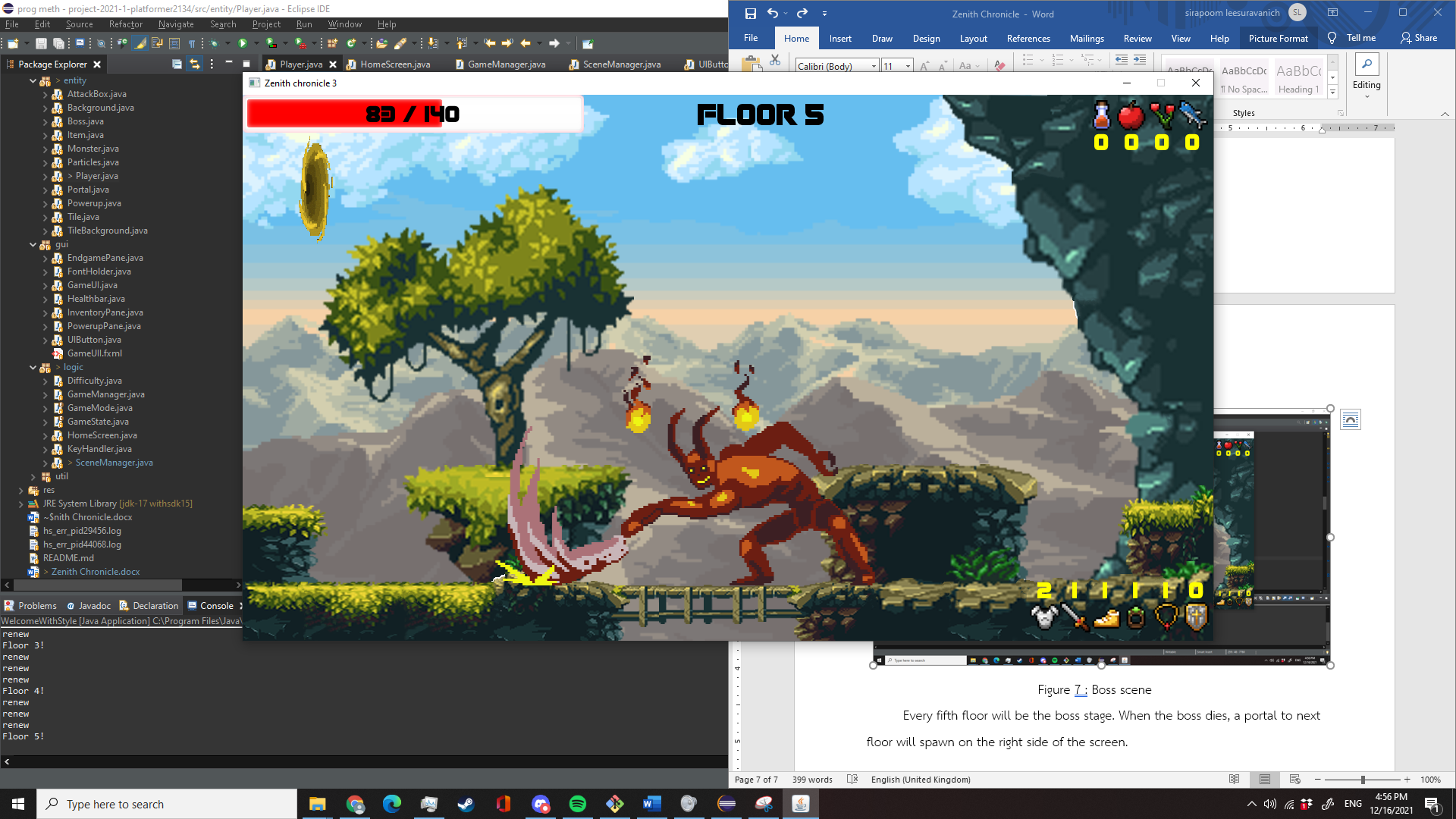
**Select difficulty scene**

Player have four options : Choose one of the three the difficulty and go to Playing Scene or they can choose to go back to main menu.

**Playing scene**

This is game scene where the game will be playing. Player will always spawn on the left side of the screen. Top-left corner show health bar of the player. Top-right corner show the temporary buffs that player currently have. And buttom-right corner will show the artifact that player had collect throughout the game.

When the player make their way to the right-end of the scene, They will find a new portal to the next floor.

**Boss scene**

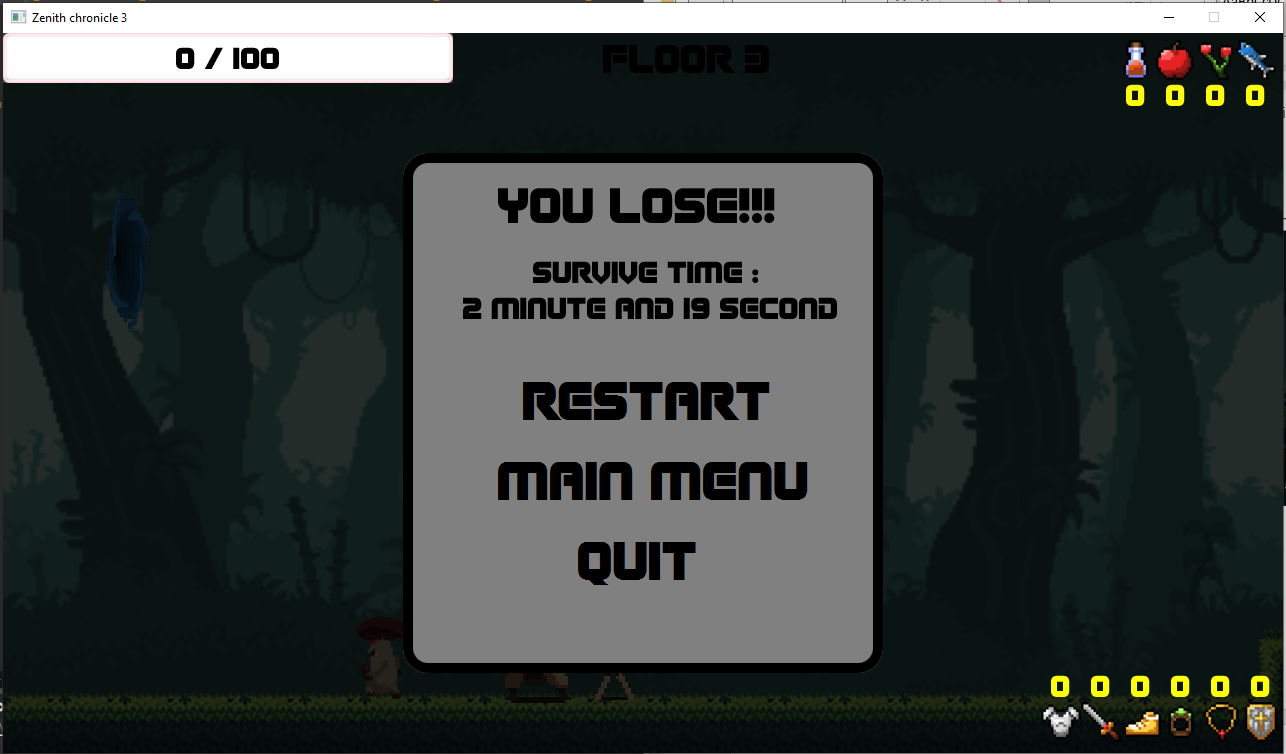
Every fifth floor will be the boss stage. When the boss dies, a portal to next floor will spawn on the right side of the screen.

**Victory scene**

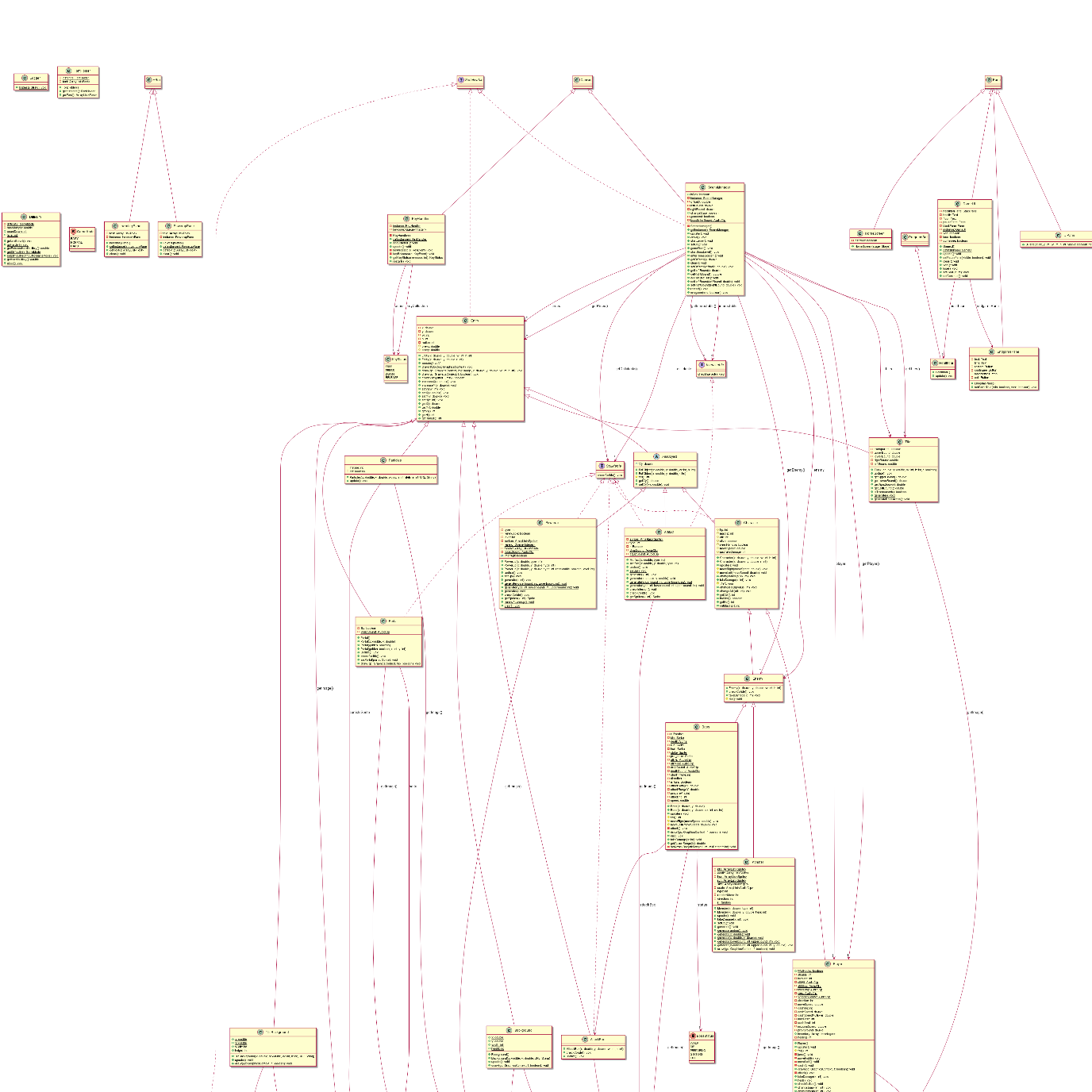
When the player have collected all types of artifacts, they win the game. They have four options, Continue : Continue playing until dies, Restart : Start over from the beginning, Main menu : Go back to the main menu, Quit : Quit the game.

**Death scene**

After winning the game, If player choose to continue fighting and die, they will see this scene showing how long they have managed to survived throughout the game. The player will have 3 options as same as Victory scene but without Continue button.

**Lose scene**

If the player die without collecting all types of the artifacts. They will see this scene and will have the same options as in Death scene.

**UML class diagram**

**1. Package component**

**1.1 class Sprite**

Contains image data for an in-game sprite.

1.1.1 Constructors

|  |  |
| --- | --- |
| + Sprite(string filepath) | Initialize these fields:  Set this.filepath as the argument filepath  Set image using new Image(ClassLoader.getSystemResource(  this.filepath).toString()) |

1.1.2 Fields

|  |  |
| --- | --- |
| -Image image | Image object contains sprite’s image |
| -String filepath | The image’s file path |
|  |  |

1.1.3 Methods

|  |  |
| --- | --- |
| +void loadImage() | Load image from filepath to image |
| Getters for each fields |  |

**1.2 Enum KeyStatus**

FREE, PRESS, DOWN, RELEASE

**1.3 Enum PlayerStatus**

IDLE, RUN, ONGROUND, GOINGUP, FALLING, RIGHT, LEFT, ATTACKING, DIE, DASHING

**1.4 Enum BossStatus**

WALK, DIE, PREPARING, STRIKING, IDLE

**2 Package entity.base**

**2.1 abstract class Entity**  
 A base structure for all entities that will exist in the game.

2.1.1 Constructors

|  |  |
| --- | --- |
| +void Entity(double x, double y, int, w, int h) | * Initialize x, y, w, h with the unput arguments * Set radius with w/2 |
| +void Entity(double x, double y, int r) | * Initialize x, y, radius with the input arguments * Set w and h with 2\*r |

2.1.2 Fields

|  |  |
| --- | --- |
| -double x | Coordinate x of the entity |
| -double y | Coordinate x of the entity |
| -int w | Entity’s width |
| -int h | Entity’s height |
| -int radius | Entity’s radius from center |
| #prevx | Coordinate x in the last frame |
| #prevy | Coordinate y in the last frame |

2.1.3 Methods

|  |  |
| --- | --- |
| *+void update()* | Update entity states and attributes each frame. |
| *+Sprite getImage()* | Returns Sprite object member of entity.  Note that an entity can have multiple sprites. |
| + draw(GraphicsContext gc, Image img, double x, double y, int w, int h) | Use graphics context to draw img on screen at coordinates (x, y) with w width and h height |
| + draw(GraphicsContext gc, boolean f) | * Call the original draw method with different arguments that depends on f argument * Will draw flipped image if f is true |
| +boolean collideWith(Entity other) | Check if with this entity and other collides |
| +void increaseX(double x) | Increase x value. Note that if x is negative then this will decrease x value. |
| +void increaseY(double y) | Increase y value |
| Getters and Setters for each field |  |

**2.2 abstract class FallObject extends Entity**

Objects of this class will fall ( increase in coordinate y ) every time fall() is called. Will stop falling once reached ground.

2.2.1 Constructors

|  |  |
| --- | --- |
| +void FallObject(double x,double y, int w, int h) | Calls super(x, y, w, h) |

2.2.2 Fields

|  |  |
| --- | --- |
| -double Vy | Determines how much y will change everytime fall() is called. |

2.2.3 Methods

|  |  |
| --- | --- |
| #int fall() | * Increases y value with Vy * Checks if entity clips through ground after increased y. If so, sets the entity on ground. * Returns 1 if entity falls downwards * Returns -1 if entity floats (y decreased) * Otherwise, returns 0 |
| Getters and Setters for each field |  |

**2.3 abstract class Character extends FallObject implements Collidable**

Class for entites that can move on terrain and can do/take damage.  
2.3.1 Constructors

|  |  |
| --- | --- |
| +void Character(double x, double y, int w, int h) | * Calls super(x, y, w, h) * Initialize these fields : * Set justTakeDamage = 0 * Set alive = true |

2.3.2 Fields

|  |  |
| --- | --- |
| #int hp | Health points of the character |
| #int maxHp | Max HP of the character |
| #int atk | Attack vale |
| #boolean alive | True if this character is alive |
| #double movespeed | Character’s movespeed (how much x will change if moveLeft() or moveRight is called. |
| #int justTakeDamage | Amount of frames left until can take damage again. |
| #boolean needRemove | True if this character needs to be remove or do something on the next update |

2.3.2 Methods

|  |  |
| --- | --- |
| +void update() | * Decrease justTakeDamage by 1 * Calls fall() |
| #void moveRight() | Increase x by movespeed and check if character clips through wall, if so, do not increase x. |
| #void moveLeft() | Same as moveRight() but decreases x |
| +void changeMaxHp(int hp) | Increase maxHp+=hp |
| +void takedamage(int atk) | Decrease hp and if hp < 0, set alive to false and call die() |
| +void changeAtk(int atk) | Increase this.atk+=atk |
| +void changeHp(int hp) | * this.hp += hp * hp can not go below zero |
| *#void die()* | Do something when a character dies |
| Getters and Setters for each field |  |

**2.4 abstract class Enemy extends Character**

Class for enemies.

2.4.1 Constructors

|  |  |
| --- | --- |
| +void Enemy(double x, double y, int w, int h) | Call super(x, y, w, h) |

2.4.3 Methods

|  |  |
| --- | --- |
| *+void checkCollide()* | Check collisions |
| +void takeDamage(int x) | * Call super.takeDamage(x) * Increase justTakeDamage by 30 |
| #void die() | Wait 0.4 sec and set needRemove to true. |

**2.5 interface Collidable**

Defines methods for Entity that collide and do something  
2.5.1 Methods

|  |  |
| --- | --- |
| *void checkCollide()* | Do something on collision |

**2.6 interface Interactable**

Defines methods for Entity that can be interacted by the player

2.6.1 Methods

|  |  |
| --- | --- |
| *void checkInteract()* | Do something on interaction |

**3. Package Entity**

**3.1 class Player extends Character**

Class for the player, the main character.

3.1.1 Constructors

|  |  |
| --- | --- |
| +void Player() | * Calls super(150 ,550, 120, 120)   (set x, y at 150, 550 and set width and height to 120x120   * Initialize these fields: * lastFrameStatus = IDLE * status = IDLE * jumpStatus = ONGROUND * face = RIGHT * direction = 0 * prevGround = 550+120 * maxDash = 1 * dashAvail = maxDash * initialize inventory with new ArrayList<Integer>(Arrays.asList(0,0,0,0,0,0)); * set alive = true * maxHp =100 * hp = maxhp * atk = 70 * moveSpeed = 7 * initJumpSpeed = 10 * dashSpeedMultiplier = 11/7 * load sprites for each action :   idle, run, jump, fall, dash, hurt, death   * start a timer thread that calls heal() every second |

3.1.2 Fields

|  |  |
| --- | --- |
| -int attackable | Frames left until can attack again |
| -int immune | Frames left until can take damage again |
| -PlayerStatus face | Direction player’s facing |
| -PlayerStatus status | Current status of player |
| -PlayerStatus lastFrameStatus | Status of player in the last frame |
| -int direction | - player’s moving direction  - -1 if moving left  - 0 if standing still  - 1 if moving right |
| -double moveSpeed | Player’s moveSpeed (in pixel) |
| -int dashing | Frames until can dash again |
| -double dashSpeed | Player’s dash length |
| -double dashSpeedMultiplier | Multiplier for moveSpeed to make dashSpeed |
| -int maxDash | Max dash player can perform before touching ground |
| -int dashAvail | Amount of dashes available left |
| -PlayerStatus jumpStatus | GOINGUP or FALLING or ONGROUND |
| -double initJumpSpeed | Initial vertical speed when player performs jump |
| -double prevGround | Last ground that player is on before going airborne (y value) |
| -Sprite idle,run,jump\_up,jump\_down,death,  attack, hurt, roll | Sprite object for each action |
| +ArrayList<integer> inventory | Player’s inventory store amount of each artifact type that player has |
| -int healing | Value that increase every second, if reaches threshold, player heals and the value reset to 0. Scale with an artifact. |
|  |  |

3.1.2 Methods

|  |  |
| --- | --- |
| +void update() | * Update player’s movement and state each frame depending on player’s input from KeyHandler and animation cooldown. * Check for collision with enemy and takeDamage(enemy’s attack) * fall() |
| #int fall() | Overrides FallObject’s fall() but add check clipping with upper Tile (collides with roof when jumping) |
| -void jump() | Set Vy to initJumpSpeed |
| -void moveRight() | - calls super.moveRight(moveSpeed)  -set offsetX in SceneManager to draw frame according to player’s coordinates |
| -void moveLeft | Same with moveRight but goes in opposite direction |
| -void dash() | moveLeft or Right with dashSpeed |
| +Sprite getImage() | Get Sprite object depending on status |
| +void draw(GraphicsContext gc, Boolean f) | Draws player with super.draw with arguments depending on status |
| -void attack() | Creates an AttackBox and check if it collides with an enemy, if so, enemy.takeDamage(atk) |
| +void takeDamage() | Take damage |
| +void heal() | Healing += 5 x amount of artifact Shield of Light |
| +void changeJumpH(int h) | Increase initJumpSpeed (which also increase jump height) |
| +changemvsp(int sp) | Increase moveSpeed |
| +void changeMaxDash(int x) | Increase maxDash |
| #protected void die() | Wait 10 seconds and set needRemove = true; |
| +boolean isimmune() | Check if player is immune to damage |
| Getters and Setters for each field |  |

**3.2 class Monster extends Character**

3.2.1 Constructors

|  |  |
| --- | --- |
| +Monster(double x,double y,int type) | Initialize each field and set type of this monster |
| +public Monster(double x,int type) | Initialize each field and set type of this monster |

3.2.1 Fields

|  |  |
| --- | --- |
| -ArrayList<Sprite> idle, run death, hurt | List of Sprite for each action of every monster type |
| -ArrayList<Integer> size | List of size for each monster type |
| -ArrayList<AudioClip> audio | List for audio effect for ech monster type |
| -int type | Monster type id |
| -int randomMove | Frames left before decide random movement again |
| -int direction | Monster’s direction 0 left, 1 right |
| -Random r | Random variable |

3.2.2 Methods

|  |  |
| --- | --- |
| +void update() | * Follows player if player is in range else do a random movement * fall() |
| +void takeDamage() | Take damage |
| +Sprite getImage | Get correct sprite depending on action |
| +void setUp() | Set up data of monsters (sprites, size, audio) |
| +void generateRandom() | Generate random monster at random x |
| +void generate(double x) | Generate random monster at x |
| +void generate(double x, double y) | Generate random monster at x, y |
| +void generate(int lowerBound, int upperBound) | Generate random monster at given x area at random y |
| +void generate(int lowerBound, int upperBound, double y) | Generate random monster at given x area and y |
| +void draw(GraphicsContext gc, boolean f) | super.draw(gc, direction==0)  draw monster on screen |
| Getters and Setters for each field |  |

**3.3 class Artifact extends FallObject implements Interactable, Collidable**

Items that can be picked up

3.3.1 Constructors

|  |  |
| --- | --- |
| +Artifact(double x,int type) | Call super constructor at x, set type |
| +Artifact(double x,double y, int type) | Call super constructor at x, y , set type |

3.3.1 Fields

|  |  |
| --- | --- |
| -ArrayList<Sprite> sprites | List of artifact sprites |
| -int type | Type id |
| -Random r | Random variable |
| -Audioclip dropSound, equipSound | sfx |

3.3.2 Methods

|  |  |
| --- | --- |
| +void update() | * fall() * checkInteract() |
| +Sprite getImage() | Get sprite |
| +void setup | Setup Sprites |
| +void generate(double x) | Generate random artifact at x |
| +void generate(double x, double y) | Generate random artifact at x, y |
| +void generate(int lowerBound, int upperBound) | Generate random artifact at given x area at random y |
| +void generate(int type, int lowerBound, int upperBound) | Generate random given type artifact at given x area |
| +void checkInteract() | Check if player interacts and add this to player’s inventory and grants effects |
| Getters and Setters for each field |  |

**3.4 class Powerup extends FallObject implements Collidable**  
3.4.1 Constructors

|  |  |
| --- | --- |
| +Powerup (double x,double y,int type) | Initialize fields  Set level = 0 |
| +Powerup(double x,double y,int type,boolean renewable,int level) | Initialize fields  Set level = 0 |
| +Powerup(double x,int type) | Initialize fields  Set level = 0 |
| + Powerup(double x,double y,int type,boolean renewable,int level) | Initialize fields |

3.4.2 fields

|  |  |
| --- | --- |
| -int type | Type id |
| -boolean renewable | If this itemcan respawm |
| -ArrayList<Sprite> sprites | List of powerups sprites |
| - Queue<Integer> renew | Queue of powerups to be renew next update |
| -ArrayList<Thread> thread | Threads of ongoing powerup effects |
| -boolean interrupt | If interrupt then don’t renew |

3.4.3 methods

|  |  |
| --- | --- |
| +Sprite getImage() | Get sprite |
| +void setUp() | Set up sprites |
| +void generate(int lowerBound, int upperBound) | Generate random powerup at given x area at random y |
| +void generate(int type, int lowerBound, int upperBound) | Generate random given type powerup at given x area |
| +void generate(x) | Generate random powerup at given x |
| +void generate(); | Generate powerups in level at exact positions |
| +void checkCollide | Check if collides with player and grant effects then start thread to count down effects cooldown |
| +void renewPowerUp() | Renew a powerup from queue |
| +void clear() | Clear threads |

**3.5 class Boss extends Enemy**

Boss of the game  
3.5.1 Constructors

|  |  |
| --- | --- |
| + Boss(double x, double y) | Initialize the boss |
| + Boss(double x, double y, int ax, int ay) | Intialize the boss with specific attack range |

3.5.2 Fields

|  |  |
| --- | --- |
| - Random r | Random |
| - Sprite idle | Sprite idle |
| - Sprite death | Sprite death |
| - Sprite run | Sprite run |
| - Sprite hurt | Sprite hurt |
| - Sprite strike | Sprite strike |
| - Sprite pre\_strike | Sprite pre strike |
| - AudioClip atkHit | Sound attack |
| - AudioClip atkMiss | Sound attack miss |
| - AudioClip dropSound | Drop item sound |
| - AudioClip deathSound | Death sound |
| - int stunImmune | StunImmune if > 0 |
| - int direction | Direction of the boss |
| - Boolean isRight | Is the boss face on right |
| - double attackRangeX | Attack range x |
| - double attackRangeY | Attack range y |
| - AttackBox attackBox | Attack box that boss can hit |
| - BossStatus status | Status |
| - int prepareAtk | Prepareatk state |
| - int attacking | Attacking state |
| - double speed | Attack speed |

3.5.3 Methods

|  |  |
| --- | --- |
| - public void update() | Update boss |
| # int fall() | Falling if on air |
| # void moveRight(double moveSpeed) | Move right |
| #void moveRight(double moveSpeed) | Move left |
| - void attack() | Attack |
| - Sprite getImage() | Return Sprite of current state |
| + void draw() | Draw the boss |
| + void die() | Call when die |
| + void takeDamage(int x) | Call when take damage |

**3.6 class Tile extends Entity**

Terrain object.

3.6.1 Constructor

|  |  |
| --- | --- |
| + Tile(double x, double y, int w, int h, boolean t) | Initialize fields |

3.6.2 Fields

|  |  |
| --- | --- |
| -boolean transparent | True if this tile is transparent (can jump through and not consider as ceiling) |
| -double upperBound | Upper bound of tile |
| -double lowerBound | Lower bound of tile |
| -double rightBound | Right Bound of tile |
| -double leftBound | Left bound of tile |

3.6.3 Methods

|  |  |
| --- | --- |
| +void generate() | Generate tiles in the level |
| +void generateBossArena() | Genrat tiles in boss arena |
| + GETTERS & SETTERS | |

**3.7 class Background extends Entity**

3.7.1 Constructors

|  |  |
| --- | --- |
| +void Background(double x, double y, String URL) | super(x,y,1280,720)  set backgroundSprite accord to URL |
| +void Background() | super(x,y,1280,720)  set default backGroundSprite |

3.7.1 Fields

|  |  |
| --- | --- |
| +DOUBLE x = 0 | x |
| +DOUBLE y = 0 | y |
| +INT WIDTH = 1280 | Width |
| +INT HEIGHT = 720 | Height |
| -Sprite | backgroundSprite |

3.7.2 Methods

|  |  |
| --- | --- |
| +Sprite getImage | Get sprite |
| +draw(GraphicsContext gc, boolean f) | Draw at constant x, y, width and height |

**3.8 class Portal extends Entity implements Collidable**

Portal to next stage

3.8.1 Constructor

|  |  |
| --- | --- |
| + Portal() | Initialize portal |
| + Portal(double x, double y) | Initialize portal at x,y |
| + Portal(boolean golden) | If golden make the portal golden |
| + Portal(boolean golden, int x, int y) | Initialize portal at x,y  If golden make the portal golden |

3.8.2 Fields

|  |  |
| --- | --- |
| - Sprite portal | Sprite |
| - Boolean flip | If need to flip when draw |
| - AudioClip warpSound | Sound play on collide |

3.8.3 Methods

|  |  |
| --- | --- |
| + void update | check collide |
| + public Sprite getImage() | Return portal |
| + void checkCollide() | If collide with player go to next floor |
| + void setPortal(Sprite portal) | Set sprite |
| + void draw(GraphicsContext gc,boolean flip) | Draw portal |

**3.9 class Particles extends Entity**

Class for particles that just used for effects

3.9.1 Constructors

|  |  |
| --- | --- |
| +Particles(double x, double y, int w, int h, int lifetime, String URL) | Initialize fields  Set sprite accord to URL  If lifetime < 0 , inf = true |

3.9.2 Fields

|  |  |
| --- | --- |
| -int lifetime | Particle lifetime |
| -boolean inf | Is particle stay forever |
| -Sprite particleSprite | sprite |

3.9.3 Methods

|  |  |
| --- | --- |
| +void update() | decrease lifetime  if lifetime <= 0 and not inf then remove this from scene |
| +Sprite getImage() | Get sprite |

**3.10 class TileBackground extends Entity**

3.10.1 Constructors

|  |  |
| --- | --- |
| +void TileBackground(double x, double y, int w, int hString URL) | super(x,y,w, h)  set tileBackgroundSprite accord to URL |

3.10.1 Fields

|  |  |
| --- | --- |
| +DOUBLE x = 0 | x |
| +DOUBLE y = 0 | y |
| +int height | Width |
| +int width | Height |
| -Sprite tileBackGroundSprite | sprite |

3.10.2 Methods

|  |  |
| --- | --- |
| +Sprite getImage | Get sprite |
| +draw(GraphicsContext gc, boolean f) | Super.draw(gc, sprite , x, y, width height) |

**4 GUI**

**4.1 class UIPane extends Pane**

A structure for Panes that exist in the game  
4.1.1 Constructors

|  |  |
| --- | --- |
| + UIPane(int x, int y, int w, int h, boolean visible, boolean fade) | Initialize the Pane with the following specification:  Set layout to x,y.  Set width and height to w,h.  Set visible to boolean visible.  Set opacity to 0.8 if fade else set border width to 10. |

**4.2 class UIButton extends Button**

A structure for Button that exist in the game  
4.2.1 Constructors

|  |  |
| --- | --- |
| + UIButton(String t, int x, int y, int fontSize, Color color, boolean transparent) | Initialize the Button with the following specification:  Set text = t.  Set layout to x,y.  Set Font to font “Alien Eclipse”, set font size t= fontSize and set font color to color.  Set background to be transparent if “transparent”. |
| + UIButton(String t,int x,int y) | Initialize the Button with the following specification:  Set text = t.  Set layout to x,y.  Set Font to font “Alien Eclipse”, set font size t= 55 and set font color to Black.  Set handle on action. |

4.2.2 Fields

|  |  |
| --- | --- |
| + AudioClip clickSound | Sound that play when the button is clicked. |

**4.3 class Healthbar extends ProgressBar**

Health bar for the player

4.3.1 Constructors

|  |  |
| --- | --- |
| + Healthbar() | Initialize the Healthbar  Set min size to 450,50.  Set glow. |

4.3.2 Methods

|  |  |
| --- | --- |
| + update() | Update the healthbar to be equal to player’s health. |

**4.4 class EndgamePane extends Pane**

Pane to be show when the game end.  
4.4.1 Constructors

|  |  |
| --- | --- |
| + EndgamePane() | Initialize the end game pane. |

4.4.2 Fields

|  |  |
| --- | --- |
| - Text text | Text that show the game result |
| - Text time | Show play time |
| - Button restart | Button for restarting the game |
| - Button continue | Button for continue after pause |
| - Button mainmenu | Button for go back to main menu |
| - Button exit | Button for exit the game |

4.4.3 Methods

|  |  |
| --- | --- |
| + setGameText(boolean win,boolean conti) | Set game text base on the two boolean.  Set pane visible to true.  Pause the game. |

**4.5 class HomeScreen extends Pane**

Home screen in the main menu.  
4.5.1 Constructors

|  |  |
| --- | --- |
| + HomeScreen(Stage stage) | Initialize the homescreen with pictures, buttons and panes. |

4.5.2 Fields

|  |  |
| --- | --- |
| - boolean escfree | Boolean if esc was holding or not |

**4.6 class InventoryPane extends HBox**

Inventory pane showing the artifacts that player had collect  
4.6.1 Constructors

|  |  |
| --- | --- |
| + InventoryPane() | Initialize the inventory pane. |

4.6.2 Fields

|  |  |
| --- | --- |
| - ArrayList<Text> text | List of text for each item in inventory pane |
| - InventoryPane instance | Singleton pattern |

4.6.3 Methods

|  |  |
| --- | --- |
| + InventoryPane getInstance() | Singleton partern |
| + ArrayList<Text> getText() | Return text(filed varieble) |
| + void clear() | Set instance to null |

**4.7 class PowerupPane extends HBox**

Powerup pane showing the powerup that currently have effects.  
4.7.1 Constructors

|  |  |
| --- | --- |
| + PowerupPane() | Initialize the powerup pane. |

4.7.2 Fields

|  |  |
| --- | --- |
| - ArrayList<Text> text | List of text for each powerup in powerup pane |
| - InventoryPane instance | Singleton pattern |

4.7.3 Methods

|  |  |
| --- | --- |
| + InventoryPane getInstance() | Singleton partern |
| + ArrayList<Text> getText() | Return text |
| + void clear() | Set instance to null |

**4.8 class GameUI extends Pane**

UI for the game that will show on pause  
4.8.1 Constructors

|  |  |
| --- | --- |
| + GameUI() | Initialize the GameUI pane |

4.8.2 Fields

|  |  |
| --- | --- |
| - Healthbar healthbar | Healthbar |
| - StackPane healthbarPane | Healthbar pane for healthbar and health text |
| - Text health | Health of the player |
| - Text floor | Text show the floor in the current game state |
| - Pane pausePane | Pane show on pause |
| - Pane blackPane | Pane with 0.8 opacity to make the background black |
| - EndgamePane endgamePane | Pane show when the game end |
| - GameUI endgamePane | Singleton |
| - boolean win | Is the game win or not |
| - boolean lose | Is the player died or not |
| - boolean continuee | If the game win and player chose to continue fighting |

**4.9 class FontHolder**

Font holder for fonts use in the game.  
4.9.1 Constructors

|  |  |
| --- | --- |
| + InventoryPane() | Initialize the fields |

4.9.2 Fields

|  |  |
| --- | --- |
| - ArrayList<Font> font | List storing fonts |
| - InventoryPane instance | singleton |

4.9.3 Methods

|  |  |
| --- | --- |
| + FontHolder getInstance() | Singleton |
| + ArrayList<Font> getFont() | Return font |

**5 Package logic**

**5.1 class DIfficulty**

Difficulty of the game  
5.1.2 Fields

|  |  |
| --- | --- |
| - GameMode difficulty | Game mode ; easy normal hard |
| - double hardMultiply | hardMultiply for the enemy |
| - countDown | countDown before boss stage |
| - int level | Current floor |

5.1.3 Methods

|  |  |
| --- | --- |
| + void goNextLevel() | Call when go to next floor |
| + void goHarder() | Call when go to next floor after killing the boss |
| + double getHardMultiply() | Return hardMultiply |
| + double getExtremeHardMultiply() | Return a bigger version of hardMultiply |
| + void setDifficulty(GameMode difficulty) | Set difficulty |
| + GameMode getDifficulty() | Return difficulty |

**5.2 class KeyHandler extends Canvas**

Key handler for gameplay  
5.2.1 Constructors

|  |  |
| --- | --- |
| + KeyHandler() | Initialize the fields and call addlistener |

5.2.2 Fields

|  |  |
| --- | --- |
| - Keyhandler instance | singleton |
| - Queue<KeyStatus> action | Queue for updating keystatus |
| - Queue<Integer> keycode | Queue for updating keystatus |
| - KeyStatus[] keyCollection | List storing all keystatus |

5.2.3 Methods

|  |  |
| --- | --- |
| + KeyHandler getInstance() | Singleton |
| + void addListener() | Add listener |
| + void update() | Update keycollection |
| + void keyPressed(KeyEvent e) | Call on key press |
| + void keyReleased(KeyEvent e) | Call on key release |
| + KeyStatus getKeyStatus | Get keystatus |
| + void restart() | Reset key handler |

**5.3 class SceneManager extends Canvas implements Serializable**

Scene manager for the game level  
5.3.1 Constructors

|  |  |
| --- | --- |
| + SceneManager() | Initialize the fields |

5.3.2 Fields

|  |  |
| --- | --- |
| + boolean isDev | For dev playtesting |
| - SceneManager Instance | singleton |
| - offsetX | offsetX for Camera |
| - ArrayList<Enemy> enemy | List of enemy |
| - ArrayList<Entity> props | List of props |
| - ArrayList<Tile> tiles | List of tiles |
| - ArrayList<Collidable> collidable | List of collidable |
| - ArrayList<Interactable> interactable | List of interactable |
| - Player player | Player |
| - double leftBound | Left bound |
| - double rightBound | Right bound |
| - boolean changeState | Change when go on next floor |
| - boolean gameend | Is game end? |
| - AudioClip bossEnterSound | Sound play when boss enter |

5.3.3 Methods

|  |  |
| --- | --- |
| + SceneManager getInstance() | Singleton |
| + update() | Update every frame and call draw() |
| + draw() | Draw every object needed to be draw |
| + void startLevel() | Call when go next floor |
| + void setUp | Setup powerup,item,monster |
| + void gameStart() | Call when game start and go next floor |
| + void startBossLevel() | Call when go in boss stage |
| + void enterHomeScreen() | Call when enter home screen |
| + double getOffsetX() | Return offsetX |
| + void clear() | Clear things in fields |
| + void addTile(Tile tile) | Add tile |
| + void restart() | Set instance to null |
| + void endgame(boolean end) | Set gameend = end |
| +Getters & Setters | |

**5.4 class GameManager**

Game manager for the game  
5.4.1 Constructors

|  |  |
| --- | --- |
| + GameManager() | Initialize the fields |

5.4.2 Fields

|  |  |
| --- | --- |
| - SceneManager Instance | Singleton |
| - boolean Gameend | Gameend? |
| - GameState state | State of the game |
| - GameState musicState | State of music |
| - boolean escPress | Helper for pause the game |
| - boolean escPress | Helper for pause the game |
| - AnimationTImer animation | Animation that run the game |
| - boolean continuee | Helper for pause the game |
| - boolean pause | Helper for pause the game |
| - int playtime | Playtime in milisecond |
| - int playtimem | Playtime in minute |
| - Thread timerThread | Thread for get timing |
| - long lastLevelTime | Collect time for each stage |
| + int SCREENWIDTH | Screen width |
| + int SCREENHEIGHT | Screen height |
| + double GRAVITY | Gravity of the game |
| + AudioClip titleBGM | Bgm |
| + AudioClip levelBGM | Bgm |
| + AudioClip bossBGM | Bgm |
| + AudioClip victoryBGM | Bgm |
| + AudioClip gameoverBGM | bgm |

5.4.3 Methods

|  |  |
| --- | --- |
| + SceneManager getInstance() | Singleton |
| + update(ActionEvent e) | Update gamestate |
| + void gameStart(ActionEvent e) | Startgame |
| + void appStart() | Start application |
| + void stopBGM() | Stop bgm |
| + void playBGM(AudioClip bgm) | Play bgm |
| + void clear() | Clear things in fields |
| + void restart() | Stop animation, end the game, set instance to null |
| + void pause() | Set the game to be pause |
| + void continue() | Set the game to be continue playing |
| + String getplaytime() | Return play time |
| + Gettet & setter | |

**5.5 enum GameMode**

EASY, NORMAL, HARD

**5.6 enum GameState**

TITLE, LEVEL, BOSS, PAUSE, GAMEOVER, VICTORY

**6 Package util**

**6.1 class Logger**

Show log

6.1.1 Method

|  |  |
| --- | --- |
| + void log(String msg) | log |

**7 Defualt package**

**7.1 class Main extends Application**

Main

7.1.1 Constructors

|  |  |
| --- | --- |
| + void main(String[] args) | Start main |

7.1.2 Method

|  |  |
| --- | --- |
| + void start(Stage stage) | Start application with stage |