Magic Slinger Documentation

Created by

Krit Cholapand 6030070521

Navabhorn Pandee 6030324021

2110215 Programming Methodology

Semester 1 Year 2018

Chulalongkorn University

Introduction

Magic Slinger is an arcade action game that provides you an amazing fun time. The goal of this game is for you to beat the boss monster. In order to get there, you have to fight with the bowling monsters first. You can not only shoot the monsters but also use cool special skill to kill them. Moreover, along with these sounds, effects and animations we implemented will increase the fun to the upper level.

User Manual

Figure 1. Shows how the game looks like when it starts. This is a resource-loading scene.

Figure 1. The loading scene of the game

After the resources are completely loaded, the player will be taken to the start scene of the game as shown in Figure 2. A start button, that is below the bottom right of the game banner, is ready to be clicked.

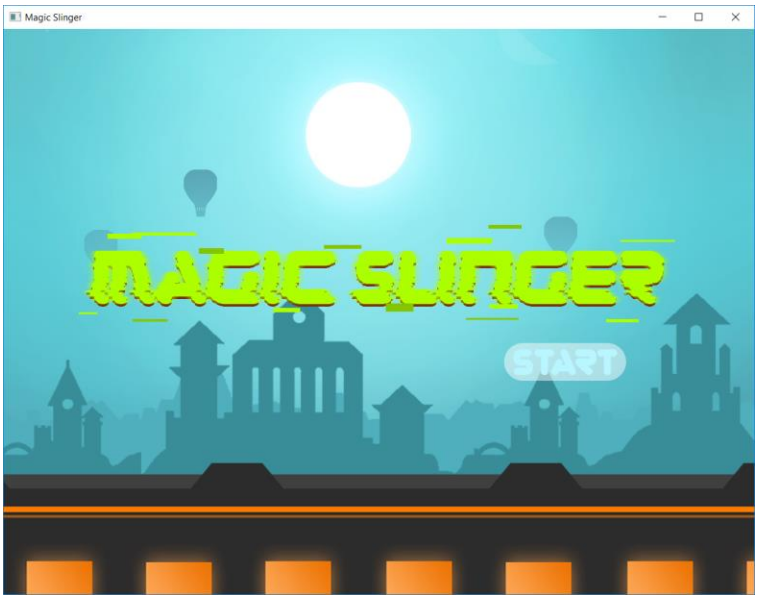

Figure 2. The start scene of the game

When the player clicks start button, then, the game scene will be appeared with a character-dropping animation at first. There are character's hp bar and score counter at the bottom left corner of the screen as shown in Figure 3. Prepare yourself to play the best game of this semester!

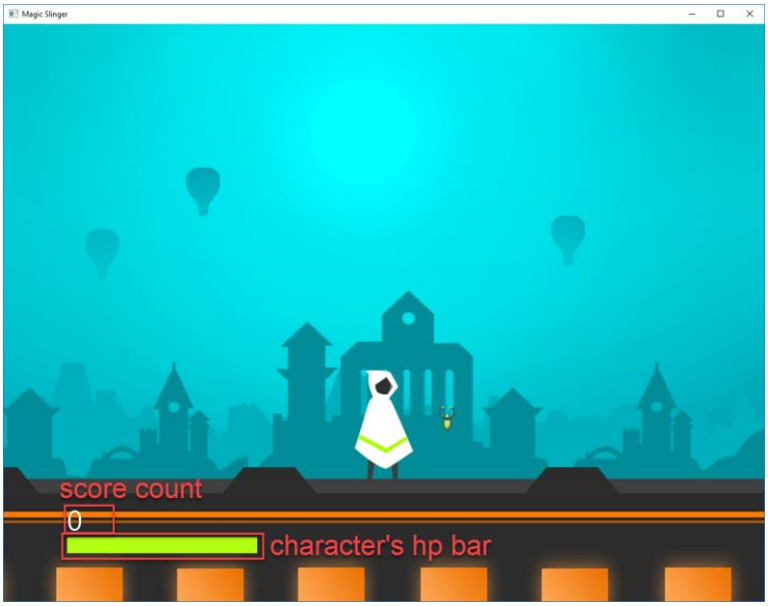

Figure 3. The game scene of the game

Game Control

- Movement

  Move right : D

  Move left : A

  Jump : W (press twice to double jump)

  Float : S

- Other controls

  Shooting direction : Mouse

  Shoot : Left click

  Skill part1, part2 : Right click

Monsters

	Name	HP	Attack Damage	Value	Behavior
	Bowling	1600	100	150	randomly float around player.
	Gunner Bowling	1000	100, 70	250	randomly float around player but keep some distance and shoot projectile.
	BowlingXL	3000	200	450	randomly float around player with low speed.
	Boss	50000		5000	HP > 25000 • float at low altitude while shooting projectile HP <= 25000 • float at high altitude while spawning other enemy

Skill

The skill is enabled when the monster is attacked and its hp is lower than 1000. Then, it will be marked with red mark which means skill is ready to use as shown in Figure 4. We can use the skill by pressing right to shoot “Weapon” to marked monster. Character’s weapon will be shoot to that monster. If miss, player will need to wait for weapon to return, If it hit marked monster, weapon will attach to target and player can also shoot from that position.
At the same time, the mark will change the color from red to green. Next, the player can press right click again in order to blink to kill and our character will gain hp.

Figure 4.

Figure 5.

Figure 6.

Gameplay

After the player clicked the start button, the player will be presented in the game scene. You can control a character by using the guide above. As time passes, the monsters are being to spawn. A simple way to attack the monsters is to press and hold left click and move the mouse to control shooting direction. You can choose between only shooting or using the skill but we suggest both. Once the monster is attacked, 10 points will add to the score. Then, if the monster is dead, its value will add to the score as well. Don’t let the monsters and the orange bullets touch you, they will decrease your hp. Also, if you want to gain your hp back, just use the special skill as much as possible. In addition, you can press and hold S button in order to float in the air. When the sky fully turns into red color, the boss will appear. Beat the boss before you lose all your hp and you will be the winner!

- private # protected + public

underline static *italic abstract* **bold final**

1. Package main

1.1 Class Main

1.1.1 Field

- Stage stage	Store the stage when the game starts

1.1.2 Method

+ void main(String[] args)	Main method that run the application
+ void start(Stage primaryStage) throws Exception	The main entry point for all JavaFX applications. The start method is called after the init method has returned, and after the system is ready for the application to begin running. [Ref. https://docs.oracle.com/javase/8/javafx/api/javafx/application/Application.html]
+ void stop() throws Exception	Called when the application stop
Getter and setter of stage	

2. Package exception

2.1 Class CharacterOutBoundException extends Exception

Throw this exception when the player use Warp out of the stage.

3.Package controller

3.1 Class GameLogic

3.1.1 Field

- GameLogic instance	Singleton of GameLogic class
- int frame	Number of frame the game has been run
- int killCount	Count enemy kill by player
- int spawnTick	inner variable for spawnMonster()
- int state	state of game (0 -> no boss 1 -> boss)
- int score	score
- BackGround bg	contain backGround object

3.1.2 Constructor

- GameLogic()	initialize bg

3.1.3 Method

+ void mainUpdate(GraphicsContext gc)	
- void updateEntity()	Loop through SharedEntity.entities and update it
- void updateDamageAttack()	Loop through SharedEntity.getDamageableEntities with every IAttackables if they collide invoke DamageableEntity.attackedBy() with attacker parameter
- void updateDraw(GraphicsContext gc)	Loop through SharedEntity.getDrawable() and draw it
- void monsterSpawn()	if state = 0 (no boss) --- if spawnTick is 0 and entity not exceed Const.MAX_ENTITY • spawn random monster at random edge of game room

	• (Bowling 70% BowlingXL 20% GunnerBowling 10%) • set spawnTick to 60-260 else if spawnTick <= 0 and entity exceed • set spawnTick to 600 else • spawnTick-- if killCound exceed Const.KILLTHRSH • set state to 1 • change bgm to Sounds.bgm_boss • create Boss at center of the room if there is no Boss • set Boss for sharedEntity • destroy all other monster
+ void reset()	Set instance.frame = 0 instance.frame = 0 instance.state = 0 instance.score = 0 instance.killCount = 0
+ GameLogic getInstance()	Singleton of GameLogic
Getters and setters of score frame and killCount	

3.2 Class SharedEntity

This class hold active entity in the game and provide access to specific type of entity.

3.2.1 Field

- SharedEntity instance	Singleton of SharedEntity
- List<Entity> entities	Lst of all active entities

- Player player	Game’s player
- Weapon weapon	Player’s weapon
- Boss boss	Game’s boss

3.2.2 Contructor

- SharedEntity()	Construct player Construct weapon

3.2.3 Method

+ void add(Entity e)	Add new entity to entities
+ void remove(Entity e)	Remove entity from entities
+ void clear()	Clear entities list
+ List<IDrawable> getDrawable()	Get list of drawable entities
+ List<Entity> getEntities()	Get list of all entities
+ List<Projectile> getProjectiles()	Get list of all Projectile entity
+ List<Particle> getParticles()	Get list of all Particle entity
+ List<Monster> getMonsters()	Get List of all Monster entity
+ List<DamageableEntity> getDamageableEntities()	Get List of all entity that can be damage by other entity
+ List<IAttackable> getIAttackables()	Get List of all entity that can attack Damageable entity
+ void reset()	clear() reinitialize player, weapon add player, weapon to entities

Getters and Setters of instance, player, weapon and boss	

3.3 Class View extends Rectangle implements IUpdateable

This class operate which part of game's stage to be draw.

3.3.1 Field

- View instance	
- int speed	Speed of view to follow player
- int MAXSHAKETICK	Duration of view shaking
- int shakeTick	Current tick of shaking
- double desX	X axis destination of the view
- double desY	Y axis destination of the view
- double tt	Period of shaking
- double tx,ty,ct	Inner variable for shake algorithm, set to 0

3.3.2 Constructor

- View()	Set x to 600 Set y to 0 Set width to Const.WINDOW_WIDTH Set height to Const.WINDOW_HEIGHT

3.3.3 Method

- void processShake()	Do the shaking
+ void update()	Set x, y to follow player processShake() Gauruntee that View will not show thing out of the scene

+ void shake()	Tell View to shake
+ void reset()	Set shakeTick = 0 desX = 0 desY = 0 ct = 0 tt = 10 tx = 0 ty = 0
Getters and setters of instance	

3.4 Class IDrawable (Interface)

This interface is a template of object that can be draw in game.

3.4.1 Method

+ int getZ()	Return depth of the drawable object, if the object return greater it draw first
+ void draw(GraphicsContext gc)	Call when draw drawable object
+ boolean isVisible()	Return drawable object's visibility if false it will not be drawn

3.5 Class IUpdateable (Interface)

This interface is a template of object that can be update.

3.5.1 Method

+ void update()	Call when object get update

4. Package entity

4.1 Package player

This class process main Player position, status by input taken when destroy gameOver.

4.1.1 Class Player extends DamageableEntity

4.1.1.1 Field

- double WALKSPEED	Speed of player walking
- double JUMPSPEED	Initial speed of player jump
- double GRAVITY	Gravity act on player when jumping
- double MAXYSPEED	Terminal velocity of player when falling
- **int MAXIMMORTALTICK**	Howlong player be immortal after collide with enemy
- int MAXJUMPCOUNT	Howmuch player can jump (2 is double jump)
- boolean isImmortal	When Immortal player will not take body hit damage.
- int immortalTick	Current immortal time left
- boolean isGround	true when player on the ground false otherise
- int jumpCount	Current jump left
- int warpTick	Inner variable to operate warp animation

4.1.1.2 Constructor

+ Player(double x, double y)	Set sprite.image to Sprites.p_idleR Set x to x Set y to y Set width to Const.PLAYER_WIDTH Set Height to Const.PLAYER_HEIGHT set hp to Const.MAX_HP set sprite speed to 0.2

4.1.1.3 Method

+ void update()	Check on ground condition Check immortal condition

	Check whether player is hover if not do • processWalk() • processJump() • processGravity() • move() ProcessWarp()
- void move()	Move the player by speedX speedY Gaurantee player to not move out bound
- void processWalk()	Check input and set speed corresponding to walk direction
- void processJump()	Check input if player command to jump Chen check jumpCount if it's not 0 then • set y speed corresponding to jump • create jump particle • play jump sound
- void processGravity()	If player not on the ground do decrease y speed due to GRAVITY but not exceed MAX_SPEEDY
- void processWarp()	If warpTick is not zero or less do warp animation and decrease warpTick
- boolean hover()	If S is pressed do hover • set speedY to 0 • create HoverParticle below player
+ void setOnWarp()	Set player on warp animation
+ int getZ()	return 1
+ void onDestroy()	Call GameScene.stopMainTimeline() Call GameScene.playOverAnim()
+ void attackedBy(IAttackable e)	If e is Monster and player is not immortal • play fx_onhitp sound • create PHitParticle on player

	• call damaged(e.attck(this)) • set Immortal to true else if e is MonsterBullet • play fx_onhitp sound • create PHitParticle on player • call damaged(e.attck(this))
+ void setX(double x) throws CharacterOutBoundException	Set player.x if x < 0 or x > 2400 • throws CharacterOutBoundException
+ void setY(double y) throws CharacterOutBoundException	Set player.y if y > 630 • throws CharacterOutBoundException
+ void heal(int amount)	heal player.hp by amount but not exceed Const.MAX_HP
Getters and setters of isImmortal, isGround, jumpCount and hp	

4.1.2 Class Weapon extends Entity

Weapon of player handle shooting and skill use

4.1.2.1 Field

- double DIS	Distance of weapon from player
- int fireCD	Bullet firing interval
- int NORMAL	0
- int FIRE	1
- int ATTACH	2
- int state	NORMAL
- Monster attach	What monster is attaching

4.1.2.2 Constructor

+ Weapon()	set sprite.image to p_weapon

4.1.2.3 Method

+ int getZ()	return 2
+ void update()	processPos() processFire()
- void processPos()	if NORMAL state • calculate angle weapon-player • set x y to hover around player by DIS with angle if in ATTACH state • calculate angle weapon-monster • set x y to hover around monster by DIS with angle • if right mouse is trigger set player x, y to current position and set state to NORMAL
- void processFire()	if left mouse is pressed and fireCD = 0 and in state NORMAL or ATTACH • play fx_fire sound • set fireCD to 5 • create PlayerBullet at x,y if right mouse is presed and state NORMAL • play fx_skill sound • set visible to false • create WeaponProjectile • set state to FIRE if right mouse is pressed and state ATTACH • play dash_sound • reset Player.jumpCount • set player x y to this.x this.y call attach.destroy() • set state to NORMAL
+ void onDestroy()	-

Getters and setters of state and attach	

4.2 Package monster

4.2.1 Class Boss

Boss of the game have predictable move at first it float near ground and shoot bullet but when go lunatic it spawn monster and float high

4.2.1.1 Field

- int MAXWALK	300
- int walkTick	current walk tick
- int bosState	0 - normal 1 - lunatic

4.2.1.2 Constructor

+ Boss(double x, double y)	set sprite.image to m_boss Set x to x Set y to y Set hp to Const.BOSS_MAX_HP Set value to 5000 set width, height, speedX, speedY to 100, 200, 1.2, 1.2

4.2.1.3 Method

+ int attack(Entity e)	return 200
# void subUpdate()	if state = 0 ---------- if walkTick > 200 • create MonsterBullet if walkTick > 0 • walkTick -- else

	• walkTick = MAXWALK • set new destination if hp less than half • set bossState to 1 • create ExplosionParticle • shake() View • set speedX, speedY to 6 if state = 1 ---------- - randomly spawn 4 monster at rate (Bowling 90% GunnerBowlin 5% BowlingXL 5%)

4.2.2 Class Bowling extends Monster

Common Monster that float around

4.2.2.1 Field

# int MAXWALK	250
# int walkTick	current walk tick

4.2.2.2 Constructor

+ Bowling(double x, double y)	Set Sprite.image to m_mob1R Set x to x Set y to y Set hp to 1600 Set value to 150 Set width, height, speedX, speedY to 65, 65, 3, 3

4.2.2.3 Method

+ int attack(Entity e)	retun 100
# void subUpdate()	spriteUpdate(); if walkTick > 0

	• walkTIck -- else • walkTick = MAXWALK • set new destinationX, destinationY
# void spriteUpdate()	Update sprite corresponding to facing

4.2.3 Class BowlingXL extends Bowling

Upgrade version of Bowling with more health and damage but move slow

4.2.3.1 Constructor

+ BowlingXL(double x, double y)	Set Sprite.image to m_mob3R Set x to x Set y to y Set hp to 3000 Set value to 450 Set width, height, speedX, speedY to 100, 100, 1.5, 1.5

4.2.3.2 Method

+ int attack(Entity e)	return 200
# void spriteUpdate()	Update sprite corresponding to facing

4.2.4 Class GunnerBowling extends Bowling

Upgrade version of Bowling that can shoot bullet move fast and stay away from player but have less health

4.2.4.1 Constructor

+ GunnerBowling(double x, double y)	if walkTick > 200 && walkTick%20 == 1
	• create MonsterBullet
	spriteUpdate();
	if walkTick > 0
	• walkTIck --
	else
	• walkTick = MAXWALK
	• set new destinationX, destinationY

4.2.4.2 Method

# void spriteUpdate()	Update sprite corresponding to facing

4.2.5 Class Monster extends DamageableEntity implements IAttackable

4.2.5.1 Field

# int NORMAL	0
# int MARK	1
# int DMARK	2
# int state	NORMAL
# int value	value of the monster when killed

4.2.5.2 Constructor

+ Monster(Image[] images, double x, double y, int maxHp)	Set sprite.image to images Set x to x Set y to y Set hp to maxHp set value to value

4.2.5.3 Method

+ void update()	move() processMark() subUpdate()
# void subUpdate()	For each sub class to have own update
# void processMark()	if hp < 1000 and not marked • state = MARK if marked • create MarkParticle at x, y else if state = DMARK

	• create DMarkParticle at x, y
+ int getZ()	return 2
+ void attackedBy(IAttackable e)	if e is PlayerBullet • play fx_onhit sound • create HitParticle • call damaged(e.attack(this)) else if e is WeaponProjectile and in MARK state • create HitParticle • call damaged(e.attack(this)) • set state to DMARK
+ onDestroy()	pla fx_explosion sound increase GameLogic.score by value increase GameLogic.killCount by 1 shake() View create ExplosionParticle create ScoreParticle with value parameter

4.3 Package particle

4.3.1 Class DMarkParticle extends Particle

4.3.1.1 Constructor

+ DMarkParticle(double x, double y)	Set sprite.image to fx_dMark set age to 1

4.3.2 Class ExplosionParticle extends Particle

4.3.2.1 Constructor

+ ExplosionParticle(double x, double y)	Set sprite.image to fx_explosion Set sprite speed to 0.13 Set age to 30

4.3.3 Class HitParticle extends Particle

4.3.3.1 Constructor

+ HitParticle(double x, double y)	Set sprite image to fx_hit Set age to 3

4.3.4 Class HoverParticle extends Particle

4.3.4.1 Constructor

+ HoverParticle(double x, double y)	Set sprite image to fx_hover Set age to 1

4.3.4.2 Method

+ int getZ()	return 0

4.3.5 Class JumpParticle extends Particle

4.3.5.1 Constructor

+ JumpParticle(double x, double y)	Set sprite image to fx_jump Set age to 5

4.3.6 MarkParticle extends Particle

4.3.6.1 Constructor

+ MarkParticle(double x, double y)	Set sprite image to fx_mark Set age to 1

4.3.7 Class Particle extends Entity (Abstract)

4.3.7.1 Field

- int age	duration of particle

4.3.7.2 Constructor

# Particle(Image[] images, double x, double y, int age)	Set sprite image to images Set x to x

	Set y to y Set age to age

4.3.7.3 Method

+ int getZ()	return 100
+ void update()	age -- if age == 0 • destroy() move()
+ void onDestroy()	•

4.4 Package projectile

4.4.1 Class MonsterBullet extends Projectile

4.4.1.1 Constructor

+ MonsterBullet(double x, double y)	Set sprite image to fx_mBullet Set x to x Set y to y Set destinationX to player.x Set destinationY to player.y Set speed to 3

4.4.1.2 Method

+ int attack(Entity e)	destroy() return 70

4.4.2 Class PlayerBullet extends Projectile

4.4.2.1 Constructor

+ PlayerBullet(double x, double y)	Set sprite image to fx_bullet Set x to x

	Set y to y Set destinationX to mouse.x Set destinationY to mouse.y Set speed to 10 Set width to 15 Set height to 15

4.4.2.2 Method

+ int attack(Entity e)	GameLogic.score += 10 create ScoreParticle destroy() return 80

4.4.3 Class Projectile extends Entity implements IAttackable (Abstract)

4.4.3.1 Field

# double speed	Speed of projectile

4.4.3.2 Constructor

+ Projectile(Image[] images, double x, double y, double destinationX, double destinationY, double speed)	Set sprite image to images Set x to x Set y to y Set destinationX to destinationX Set destinationY to destinationY Set speed to speed

4.4.3.3 Method

+ int getZ()	return 99
+ void update	move()

	if out of game stage • destroy()
+ void move()	increase x y by speedX speedY
+ onDestroy()	
Getter and Setter of speed	

4.4.4 Class WeaponProjectile extends Projectile

4.4.4.1 Field

- int NORMAL	0
- int ATTACH	2
- int ageTick	120

4.4.4.2 Constructor

+ WeaponProojectile(double x, double y)	Set sprite to p_weapon Set x to x Set y to y Set destinationX to mouseX Set destinationY to mouseY Set width, height to 100

4.4.4.3 Method

+ void update()	move() if age <= 0 • set Weapon state to NORMAL • set Weapon visible to true else • ageTick -= 1
+ int attack(Entity e)	Play fx_targetLock sound

	Set Weapon state to ATTACH Set Weapon visible to true Set Weapon attach to e

4.5 Class DamageableEntity extends Entity (Abstract)

This abstract class is a subclass of class Entity. It adds functionality to kill and to be damaged.

4.5.1 Field

# int hp	current hit point of entity
# int maxHp	max hp of entity

4.5.2 Constructor

+ DamageableEntity(Image[] images, double x, double y, double w, double h, int maxHp)	Set sprite image to images Set x, y, width, height to x, y, w, h Set hp to maxHp Set maxHp to maxHp
+ DamageableEntity(Image[] images, double x, double y, int maxHp)	Set sprite image to images Set x, y to x, y, Set hp to maxHp Set maxHp to maxHp

4.5.3 Method

+ void attackedBy(IAttackable e)	Call when get attack by attackable entity
# void damaged(int atk)	Call when to calculate damage hp -= atk if hp <= 0 • hp = 0 • destroy()
Getters and setters of hp and maxHp	

4.6 Class Entity extends Rectangle implements IDrawable, IUpdateable (Abstract)

4.6.1 Field

+ int LEFT	-1
+ int RIGHT	1
# Sprite sprite	sprite of entity
# double speedX	speed in x axis
# double speedY	speed in y axis
# double destinationX	x axis destination
# double destinationY	y axis destination
# int facing	direction facing
# boolean visible	visibility of entity

4.6.2 Constructor

+ Entity(Image[] images, double x, double y, double width, double height)	Set sprite image to images Set x to x Set y to y Set speedX to 0 Set speedY to 0 Set destinationX to x Set destinationY to y Set facing to RIGHT Set visible to true Set width, height to width, height
+ Entity(Image[] images)	Call super() Set sprite image to images Set speedX to 0

	Set speedY to 0 Set destinationX to x Set destinationY to y Set facing to RIGHT Set visible to true
+ Entity()	Call super() Set speedX to 0 Set speedY to 0 Set destinationX to x Set destinationY to y Set facing to RIGHT Set visible to true

4.6.3 Method

+ void draw(GraphicsContext gc)	Call when drawn default is • draw sprite image at calculated x y corresponding to view • animate sprite
+ void move()	x, y += speedX, speedY Set facing corresponding to direction moving
+ void create()	add entity to SharedEntity
+ void destroy()	remove entity from SharedEntity call onDestroy()
+ void onDestroy()	called when destroy()
Getters and setters of visible, speedX, speedY, destinationX, destinationY, facing and sprite	

4.7 Class IAttackable (Interface)

This interface is the template for entity that can attack DamageableEntity

4.7.1 Method

+ int attack(Entity e)	call when attacking return amount of damage inflick

4.8 Class Rectangle (Abstract)

This class provide fundamental property of rectangle and funtion to check collision to other object

4.8.1 Field

# double x	x position
# double y	y position
# double width	width of the rectangle
# double height	height of the rectangle

4.8.2 Constructor

+ Rectangle()	set x, y, width, height to 0
+ Rectangle(double x, double y, double width, double height)	Set x to x Set y to y Set width to width Set height to height

4.8.3 Method

+ boolean isCollideWidth(Rectangle b)	Check collision with b
+ boolean isCollideWidth(double x, double y, double width, double height)	Check collision with rectangle with x = x y = y

	width = width height = height
Getters and setters of width, height, x and y	

5. Package constants

5.1 Class Const

5.1.1 Field

+ int WINDOW_WIDTH	Store the value of window width (1200)
+ int WINDOW_HEIGHT	Store the value of window height (900)
+ int GROUND_POS	Store the value of y axis of the ground position (720)
+ double PLAYER_HEIGHT	Store the value of the character height fot hit area (150 actually 180)
+ double PLAYER_WIDTH	Store the value of the character width fot hit area (50 actually 92)
+ double FRONT_CITY_SPEED	Store the speed of the front city animation (20)
+ double BACK_CITY_SPEED	Store the speed of the back city animation (2)
+ double STAGE_WIDTH	Store the value of the stage width (2400)
+ int MAX_HP	Store the value of the character's hp (1000)
+ int BOSS_MAX_HP	Store the value of the boss's hp (10000)

6. Package input

This class provide other class to check input from user

6.1 Class Input

6.1.1 Field

- Set<KeyCode> activeKeys	Collection of current active key
- Set<KeyCode> triggerKeys	Collection of key triggering

- boolean leftMouseActive	Wheter left mouse is active
- boolean rightMouseActive	Wheter right mouse is active
- boolean leftMouseTrigger	Wheter left mouse is triggering
- boolean rightMouseTrigger	Wheter right mouse is triggering
- double mouseX	Mouse x position
- double mouseY	Mouse y position

6.1.2 Method

- void addKey(KeyCode code)	When press key • if key is not in active key add key to triggerKeys • add key to activeKeys
- void removeKey(KeyCode code)	remove keys frome activeKeys
+ void clear(()	clear all active keys
+ void bindScene(Scene scene)	Bind input to listen to scene
+ void unBindScene(Scene scene)	Unbind scene
+ boolean isKeyActive(KeyCode code)	Check wheter key is active
+ boolean isKeyTrigger(KeyCode code)	Check wheter key is triggering
+ boolean isLeftMouseTrigger()	Check wheter left mouse is trigger
+ boolean isRightMouseTrigger	Check wheter right mouse is trigger
+ boolean isLeftMouseActive()	Check wheter left mouse is active
+ boolean isRightMouseActive()	Check wheter right mouse is active
Gettes of mouseX and mouseY	

7. Package resource

7.1. Package scene

7.1.1 Class LoadingScene

This class is the first scene set to stage handle loading animation and load all resource into game.

7.1.1.1 Field

- Pane root	Root of the scene
- Canvas canvas	Canvas of the scene
- Thread loadThread	Thread use to loading resources
- int animationFrame	Intialize with 0,
- boolean animFinished	Initailze with FALSE, to indicate that the fade-in animation is already finished or not

7.1.1.2 Constructor

+ loadingScene()	• Initialize the loadindScene with the size 1200x900. • Create 1200x900 canvas and add it to the scene. • Sets the background color to white and set padding to 0. • initialize canvas and graphic context gc • define and start loadingThread • playLoopAnimation(gc) • Wait for loadingThread to finish and animation to finish then playFadeOutAnimation • add banner componet to root

7.1.1.3 Method

- void playFadeOutAnimation(GraphicsContext gc)	Use transition to fadeout root When finish set game scene to StartScene play bgm on StartScene
- void playLoopAnimation(GraphicsContext gc)	Use timeline to fade in banner

	Use transition to fade out banner
Getter and Setter of loadThread	

7.1.2 Class StartScene

This class set to stage after loading scene play as start scene of the game.

7.1.2.1 Field

- Pane root	Root of the scene
- Canvas canvas	Canvas of the scene
- ImageView nameBanner	Banner of game name
- ImageView startBtn	Button to start game
- MediaPlayer bgm	Main bgm of the scene
- Timeline animBg	Timeline of background animation
- int bgFrame	Frame played of background animation
- double backPos	Background back city position
- double frontPos	Background front city position
- Timeline animFade	Timeline of fade animation use in cut scene
- int fadeFrame	Frame of fade animation
- Timeline animDrop	Timeline of player drop animation
- int dropFrame	Frame of drop animation
- int NORMAL	0
- int CUTSCENE	1
- int state	set to NORMAL

7.1.2.2 Constructor

+ StartScene()	• Initialize the loadindScene with the size 1200x900 • Create 1200x900 canvas and add it to the scene. Sets the background color to white and set • initialize canvas and graphic context gc • set bgm to play infinite • set Cursor to ui_coursor[0] • initialize nameBanner and set position to (120,300) • initialize startBtn and set position to (800,500) • add all componet to root addStartEventHandler() playBgAnimation(gc)

7.1.2.3 Method

- void playBgAnimation(GraphicsContext gc)	set bgFrame to 0 initialize animBg • calculate backPos and frontPos corresponding to bgFrame Const.BACK_CITY_SPEED Const.FRONT_CITY_SPEED • clear old drawn • draw bg_sky bg_backCity bg_frontCity bg_train corresponding to backPos and frontPos • bgFrame++ set animBg to play INDEFINITE play animBg
- void playStartAnimation(GraphicsContext gc)	Set fadeFrame to 0 Set dropFrame to 0 This method play start cutscene Initialize animFade • like bg animation • draw black rectangle with opacity corresponding to fadeFrame

	• fadeFrame++ • bgFrame++ Set animFade to play half second Play animFade when finsih play animDrop Initialize animDrop • draw black rectangle as background • draw player droping corresponding to drop frame • dropframe ++ Set animDrop to play 21 frame On finish animDrop set Main stage' scene to gameScene
- void addStartEventHandler()	Add EventHandler to startBtn When mouse hover set button to ui_startH Set button to ui_start otherwise On clicked set startBtn and nameBanner visible to false stop animBg and playStartAnimation((GraphicsContext)canvas.getGraphicsContext2D());

7.1.3 Class GameScene

This class is main game scene set to stage when in main game. Updating game logic and provide play pause stop function also provide animation when win or game over.

7.1.3.1 Field

- Pane root	Root of the scene
- Canvas canvas	Canvas of the scene
- MediaPlayer bgm	Main bgm of the scene
- Timeline tl	Main timeline of the game

7.1.3.2 Constructor

+ GameScene()	• Initialize the loadindScene with the size 1200x900 • Create 1200x900 canvas and add it to the scene.

	Sets the background color to white and set • initialize canvas and graphic context gc • set bgm to bgm_game and play INDEFINITE • set Cursor to ui_coursor[0] • bind Input with scene • addMainTimeline(gc)

7.1.3.3 Method

- void addMainTimeline(GraphicsContext gc)	initialize main timeline of the game • update GameLogic set tl to play INDEFINITE
- void playMainTimeline()	fade in canvas and play tl
- void pauseMainTimeline()	pause tl
- void stopMainTimeline()	stop tl and bgm
+ void gameReset()	call • GameLogic reset() • SharedEntity reset() • View reset() • re initialize startScene and gameScene set Main stage to startScene

7.1.4 Class Scene

This class provide access to LoadingScene StartScene and GameScene

7.1.4.1 Field

- Scene loadingScene	= new loadingScene()
- Scene startScene	for store StartScene
- Scene gameScene	for store GameScene

7.1.4.2 Method

+ void loadResource()	Call when game load • initialize startScene and gameScene
Getters of loadingScene, startScene and gameScene	

7.2 Class Sprite

This class provide interface for javafx.Image including

- function to animate sprite
- function to set sprite animation speed
- function to set sprite frame
- funtion to set Image

7.2.1 Field

# Image[] sprite	Image of sprite
# double frame	frame sprite are playing
# double speed	speed of sprite animation

7.2.2 Constructor

+ Sprite()	Set sprite to blank Image Set frame to 0 Set speed to 1
+ Sprite(Image[] image)	Set sprite to image Set frame to 0 Set speed to 1
+ Sprite(Image[] image, int frame, int speed)	Set sprite to blank Image Set frame to frame Set speed to speed

7.2.3 Method

+ void animate()	Animate sprite • increase frame by speed • guaruntee frame to not exceed sprite length
+ Image getImage()	Return image of current frame
+ void setSprite(Image[] images)	Set sprite to images
+ int getFrame()	Get current frame in integer
+ void setFrame(double frame)	Set current frame
getter and setter of speed and getter of sprite	

7.3 Class Sprites

This class store all Image and have loadResource method called when game load

7.3.1 Field

+ Image[] loading_banner	Store the image of loding-scene banner
+ Image[] ui_nameBanner	Store the image of name banner on start scene
+ Image[] ui_start	Store the image of start button on start scene
+ Image[] ui_startH	Store the image of start button with highlight on start scene
+ Image[] bg_sky	Store the image of sky background
+ Image[] bg_backCity	Store the image of city at the back
+ Image[] bg_frontCity	Store the image of city at the front
+ Image[] bg_train	Store the image of the train
+ Image[] p_jumpL	Store the image of the left jumping character
+ Image[] p_jumpR	Store the image of the right jumping character
+ Image[] p_walkL	Store the images of the left walking character

+ Image[] p_walkR	Store the images of the right walking character
+ Image[] p_dashL	Store the images of the left dashing character
+ Image[] p_dashR	Store the images of the right dashing character
+ Image[] p_idleL	Store the image of the left idle character
+ Image[] p_idleR	Store the image of the right idle character
+ Image[] p_weapon	Store the image of character's weapon
+ Image[] fx_bullet	Store the image of character's bullet
+ Image[] fx_mBullet	Store the image of monster's bullet
+ Image[] fx_jump	Store the image of the jumping effect
+ Image[] fx_hit	Store the image of the hit effect
+ Image[] fx_mark	Store the image of the red mark
+ Image[] fx_dMark	Store the image of the green mark
+ Image[] fx_explosion	Store the images of the monster's explosion
+ Image[] fx_hover	Store the image of the hover base
+ Image[] m_mob1L	Store the image of bowling monster on the left side
+ Image[] m_mob1R	Store the image of bowling monster on the right side
+ Image[] m_mob2L	Store the image of gunner bowling monster on the left side
+ Image[] m_mob2R	Store the image of gunner bowling monster on the right side
+ Image[] m_mob3L	Store the image of big bowling monster on the left side
+ Image[] m_mob3R	Store the image of big bowling monster on the right side
+ Image[] m_boss	Store the image of the boss

7.2.2 Method

+ void loadResource()	load all Image[]

7.3 Class Sounds

This class store all AudioClip and have loadResource method called when game load

7.3.1 Field

+ AudioClip bgm_boss	Store the mp3 of background music of boss scene
+ AudioClip bgm_game	Store the mp3 of background music of game scene
+ AudioClip bgm_gameOver	Store the mp3 of background music of game over scene
+ AudioClip bgm_train	Store the mp3 of background music of start scene
+ AudioClip bgm_victory	Store the mp3 of background music of victory scene
+ AudioClip fx_fire	Store the mp3 of shot sound effect
+ AudioClip fx_onhit	Store the mp3 of monster on hit sound effect
+ AudioClip fx_onhitp	Store the mp3 of player on hit sound effect
+ AudioClip fx_jump	Store the mp3 of jump sound effect
+ AudioClip fx_skill	Store the mp3 of skill-using sound effect
+ AudioClip fx_targetLock	Store the mp3 of target-locking sound effect
+ AudioClip fx_dash	Store the mp3 of dash sound effect
+ AudioClip fx_explosion	Store the mp3 of explosion sound effect

7.3.2 Method

+ void loadResource()	load all Sound
+ AudioClip getBgm_train()	getter of bgm_train

8. Package background

8.1 Class BackGround implements IDrawable, IUpdateable

This class handle animation of background

8.1.1 Field

- double backPos	Background back city position
- double frontPos	Background front city position

8.1.2 Method

+ int getZ()	return -9999
+ void update()	update the animation • calculate backPos frontPos corresponding to Const.BACK_CITY_SPEED Const.FRONT_CITY_SPEED and current GameLogic frame
+ void draw(GraphicsContext gc)	draw bg_sky bg_backCity bg_frontCity bg_train corresponding to backPos and frontPos with amount of red color corresponding to GameLogic.killCount
+ boolean isVisible()	return true

9. Package gui

9.1 Class GUI implements IDrawable

This class animate GUI of game including score player health bar and boss health bar.

9.1.1 Field

- int playerHp	Player current hp
- double BAR_L	Player health bar lenght
- double BAR_H	Player health bar height
- int pX	Player health bar x

- int pY	Player health bar y
- double SPEED	Health bar speed
- double curL	current lenght of health bar
- double desL	actual lenght of health bar
- double BOSS_BAR_L	Boss health bar lenght
- double BOSS_BAR_H	Boss health bar height
- int bX	Boss health bar x
- int bY	Boss health bar y
- double bdesL	Boss health bar current lenght
- GUI instance	Singleton of GUI object

9.1.2 Constructor

- GUI()	Set playerHp to Const.MAX_HP Set CurL to 0

9.1.3 Method

+ int getZ()	return 9999
+ boolean isVisible	return true
+ void draw(GraphicsContext gc)	We update this class within draw because it take merely no calculation in updation • Set desL corresponding to player curret hp • Set curL to approach desL with speed • draw BLACK rectangle with x, y, width, height = pX, pY, BAR_L, BAR_H • draw GREEN rectangle with x, y, height = pX, pY, BAR_H with width corresponding to curL

	if there is boss in game draw boss's hp • draw BLACK rectangle with x, y, width, height = bX, bY, BOSS_BAR_L, BOSS_BAR_H • draw ORANGE rectangle with x, y, height = pX, pY, BAR_H with width corresponding to bdesL draw Score • draw WHITE text with size 45 + GameLogic.socre/200 at pX, pY-10
+ GUI getInstance()	Singleton of GUI object
+ String toString()	return "GUI [playerHp=" + playerHp + ", BAR_L=" + BAR_L + ", BAR_H=" + BAR_H + ", X=" + pX + ", Y=" + pY + ", SPEED=" + SPEED + ", curL=" + curL + ", desL=" + desL + "]"; (for debuggin)