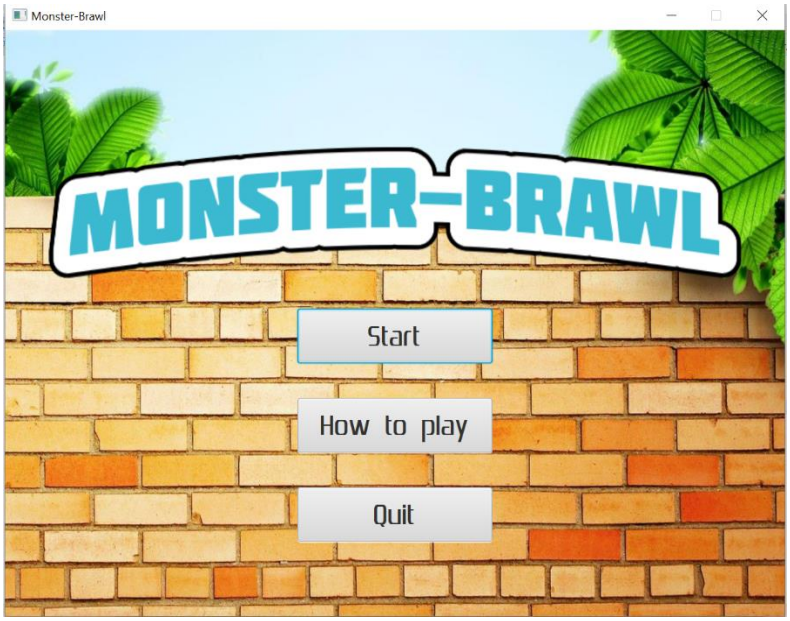
Monster-Brawl Documentation



Created by
Tanut Lertwarachai 6331311821
Pawan Kanjeam 6331322721

2110215 Programming Methodology
Semester 2 Year 2020
Chulalongkorn University

Intro

Introduction

2 Players use three doors to escape from the wild jungle. They must release their powerful monsters to hit the barrel and get damaged from the explode barrel to open their doors. The more powerful damage can unlock the door. After the explosion, the poison under the ground leaks to this lane. If any monsters are released to this lane, they will die immediately

Rule

Two players must compete to win the game. You must use coin to release monster to destroy barrel in each lane. There are five lanes and winner must win three out five lanes. Barrel receives damage from two sides. If different of damages from two sides more than value that we define (100 damages), the side that deal more damages will win in that lane.

Game scene

Main Menu scene

the main menu screen has start button, How-to-play button and Quit button.

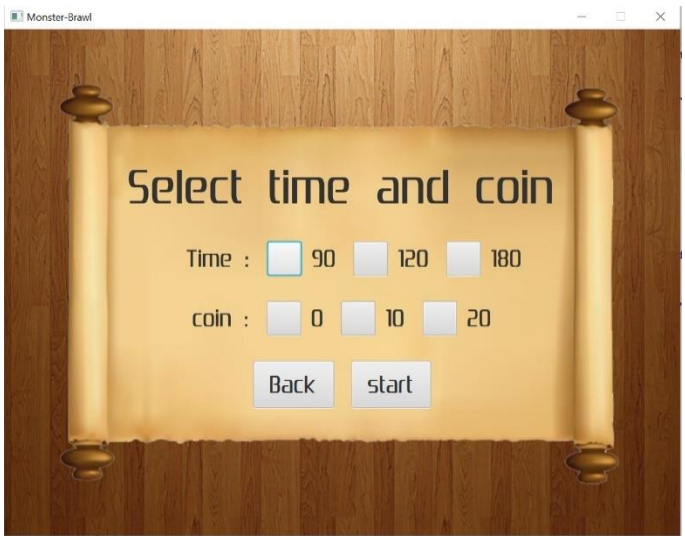
Start button: go to pre-game scene
How-to-play button: go to How-to-play scene
Quit button: quit the program

How-to-play scene

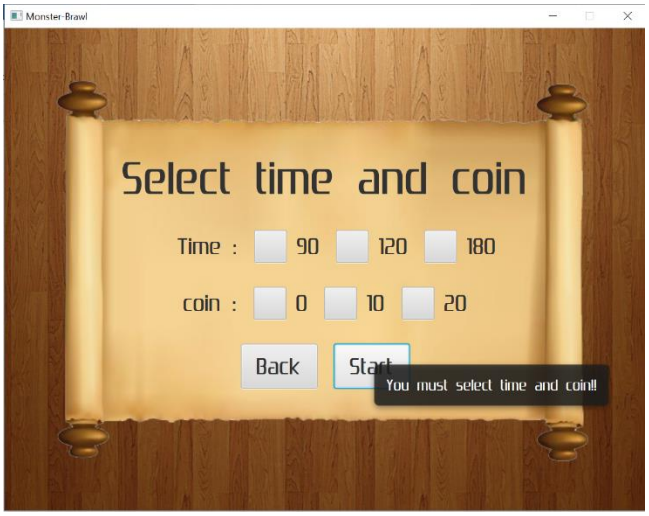
Showing keyboard control of player one and player two
** press Back button to get back to main menu*

Player one	Player two
W: go up	UP: go up
S: go down	DOWN: go down
SPACE BAR: release monster	LEFT: release monster

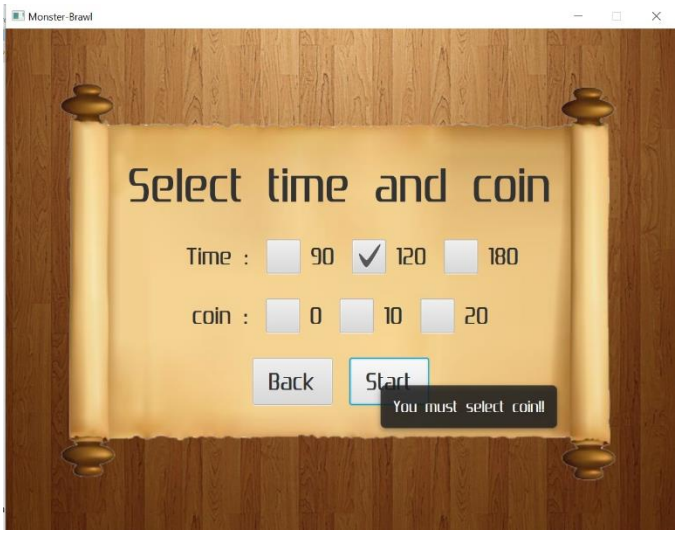
Pre-game scene



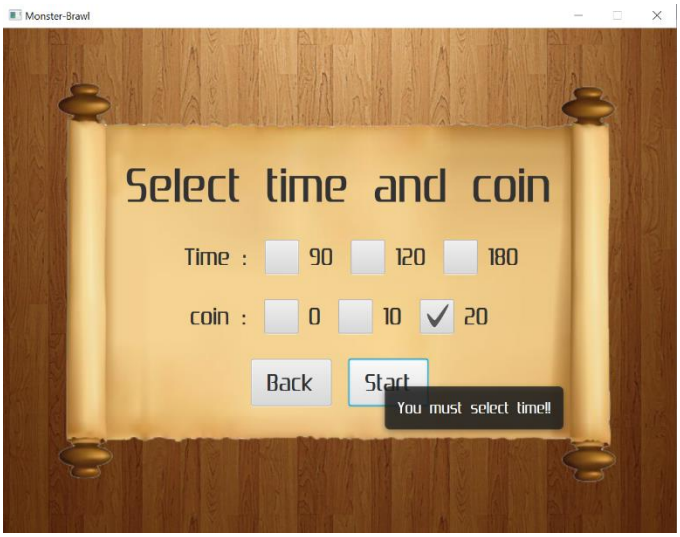
Select start time and start coins in the gameplay
** press Start button to start the game*
**press Back button to go back to menu*



If Both Time box and Coin box are not selected



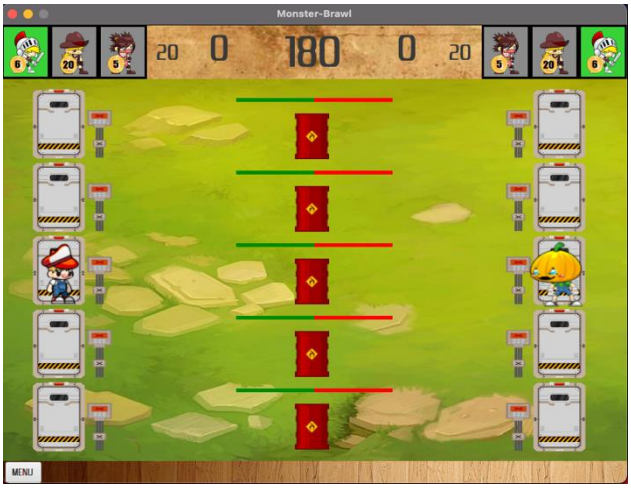
if none of Time boxes is selected



if none of Coin boxed is selected

Gameplay Scene

Start gameplay



Gameplay scene when you selected start game time to 180 and start coin to 20

Releasing monster

- When press releasing monster button, monster will come out from player and run to hit the barrel in that lane.
- Player's coins will decreased by cost of monsters.
- You can't release the monster that you have not enough coins to its cost.
- Player's coins will increase 2 coins/2sec (if player coins equal to 49 , coins will increase by 1)
- All of the doors are closed when game starts, and all switches is off state.

Barrel in each lane will receive damage from the player

- Player one deals more than damages than player two in lane 3. So, green bar will increase, and red bar will decrease.
- Player two deals more than damages than player one in lane 4. So, green bar will decrease, and red bar will increase.

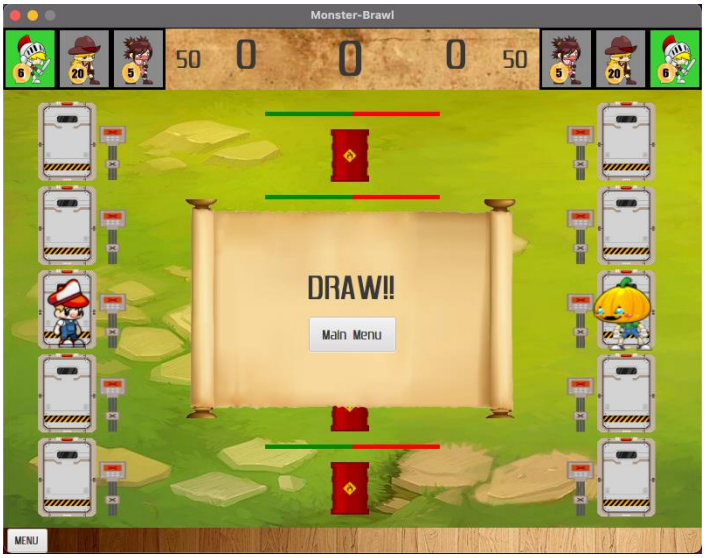
When two player release monsters of each other

Winning

Player one win lane3.

When winning in each lane
- winner's door will open
- switch will be on
- barrel will be destroyed.
- winner point's increases by 1

Cannot release any monsters in the lane that does not have barrel (monster will die instantly if you release)



Game result is draw

- If total damages deal to the barrel from player one and player two are equal, game result is draw.

Player two won the game

Winner can be settled by point and total damage.

- Not time out:
  A player that has 3 points first is a winner.
- Time out:
  1. A player that has higher point is a winner.
  2. If each side got same point, the player who deals higher total damage will win.
  3. If point and total damage of each side is equal, it's a draw.

Monsters

Knight

- has a basic attack , speed and health values

Adventurer

- can kill opponent monster 3 times
- after 3 times of killing , she will run to hit the barrel
- attack is low

Ninja

- highest speed
- she can run to barrel quickly

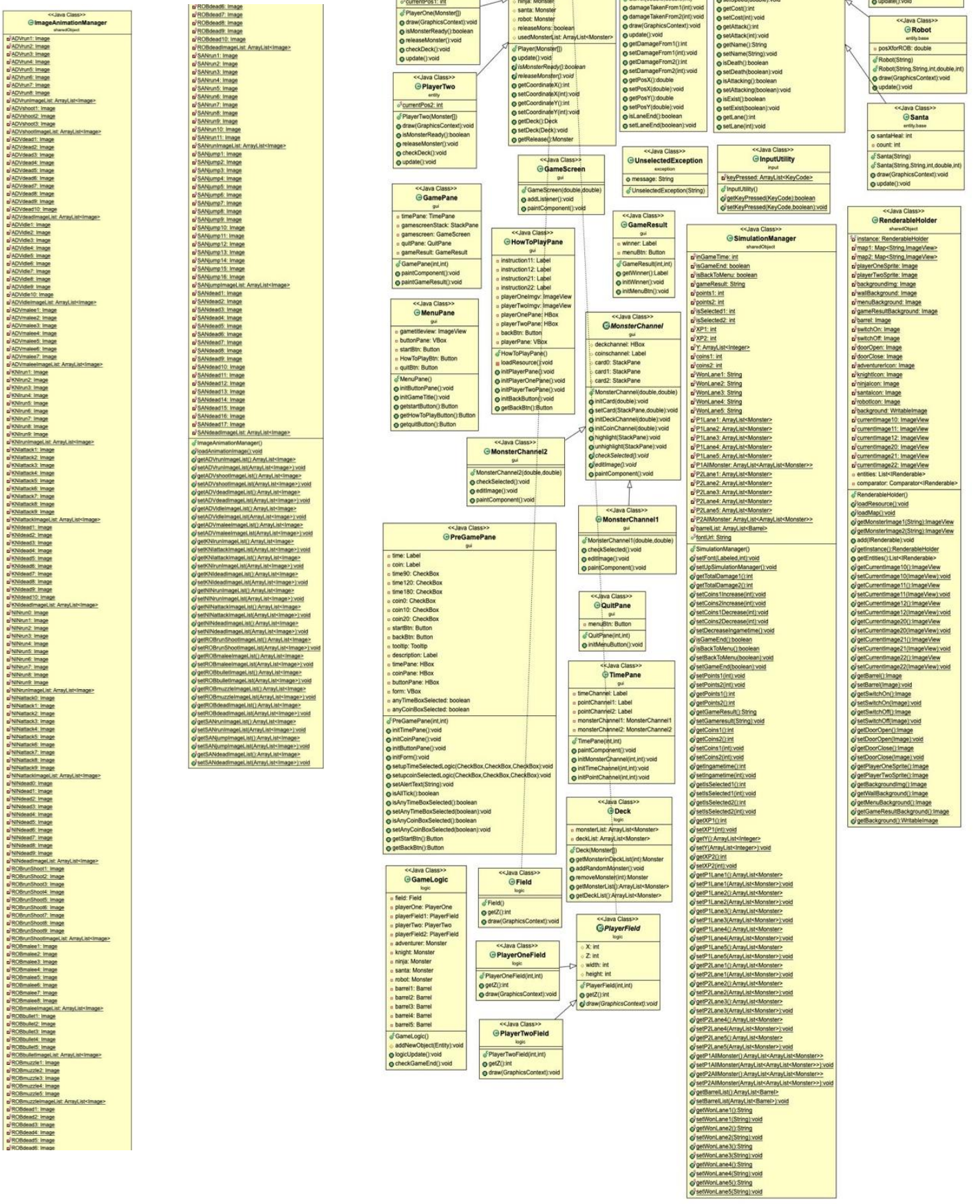
Robot

- can attack the barrel while running

Santa

- has no damaged
- when released , he will give a gift to player by increase player’s coins increasing rate.

Class Diagram



** Noted that Access Modifier Notations are listed below*

+ (public)
# (protected)
- (private)
Underline (static)
Italic (abstract)

1. Package application

1.1 Class Main

Extends application

This class is main class to run the game.

Field

- long lastTimeTriggered	Used to compare with time in animation timer when 1 second pass.
- MenuPane menu	Displays main menu pane
- HowToPlayPane howToPlay	Displays how to play pane
- GamePane gamepane	Entire Gameplay pane
- GameLogic logic	
- Scene menuScene	Displays main menu screen
- Scene pregameScene	Displays pre-game screen
- Scene gameScene	Displays game screen
- Scene HowToPlayScene	Displays how-to-play screen
- MediaPlayer menuMusic	Music which plays in main menu
- MediaPlayer gameplayMusic	Music which plays in gameplay

Method

+ void start(Stage stage)	- initialize menu screen - set action to start button, how-to-play button and quit button in menupane.
+ void setPreGamePane(Stage stage)	- initialize pre-game pane and to pre-game scene - set actions to the buttons in pre-game scene
+ void setHowToPlayPane(Stage stage)	- set how-to-play pane to how-to-play scene - set action to the button in how-to-play scene
+ void setGamePlay(Stage stage)	- initialize the entire gameplay component and displays to the screen
+ void startAnimation(Stage stage)	- run animation of entire game

+ void startTimer(Stage stage)	- start timer of the game - check and update gametime state and coin in the game
+ void initSound(Stage stage)	- initialize menu music and gameplay music
+ void main(String[] args)	Launch the game

2 Package entity

2.1 abstract Class Entity

Implements IRenderable

This class is the abstract class that show in the canvas and can update its state.

Constructor

+ Entity()	Set z to 100.

Method

+ int getZ ()	Return z
+ abstract void update()	Update the position of knight. (posX)

2.2 Class Barrel

Extends Entity

This class represents Barrel type Entity, which is at the middle of each lane.

Field

- int damageFrom1	The amount of damage, hit by PlayerOne.
- int damageFrom2	The amount of damage, hit by PlayerTwo.
- int greenWidth	The length of green tube.
- Int breakValue	If the difference damageFrom1 and damageFrom2 are more than breakValue, this lane is end. (set as 100)

- boolean isLaneEnd	If this lane is end, this will be set as true. (set as false)
# double posX	The position of barrel in X-axis.
# double posY	The position of barrel in Y-axis.
# int Lane	The lane of this barrel.

Constructor

+ Barrel (double posX, double posY, int lane)	initialize barrel: - set posx. - set posY. - set isLaneEnd to false. - set z to 50. - set breakValue to 100. - set damageFrom1 and damageFrom2 to 0. - set greenWidth to 100.

Method

+ void damageTakenFrom1 (int damage)	This method adds the damage receiving from PlayerOne.
+ void damageTakenFrom2 (int damage)	This method adds the damage receiving from PlayerTwo.
+ void draw (GraphicsContext gc)	Draw the barrel if this lane is not end.
+ void update ()	- Update the difference of damage and the length of green tube. - Set the laneEnd of each lane and who is winning the lane. - If lane end, all monsters in the lane will die.
+ getter/setter	

2.3 abstract Class Player

Extends Entity

This class represents Player type Entity, which can release monsters.

Field

# int currentPos	Current position of the player. (from 0 to 4)
# int coordinateX, coordinateY	The coordinate X and Y of player
# boolean[] pos	The array of boolean (position) that true represents the position of player. For example : If player is in the first lane then pos = [true, false ,false, false, false]
# KeyCode up	The player goes up.
# KeyCode down	The player goes down.
# KeyCode releaseMonster	The player releases monster.
# KeyCode choose0	The player chooses the first monster on monster channel.
# KeyCode choose1	The player chooses the second monster on monster channel.
# KeyCode choose2	The player chooses the third monster on monster channel.
# int coins	The coin of the player that uses for buy the monster to release.
# Monster release	The monster that is released.
# Deck deck	The deck of monsters.
# Image currentSprite	The image of the player.
# Monster adventurer, knight, ninja, santa, robot	The type of monsters in the deck for each player.
# boolean releaseMons	The boolean that will be set as true when releaseMonster KeyCode of player is triggered and turn back to false when it's not. (set as false)
# ArrayList<Monster> usedMonsterList	This array list keeps the released monsters.

Constructor

+ Player(Monster… monsters)	Initialize Player and set the following:

	- currentPos = 2 - pos[currentPos] = true Initialize new deck from the inputted array.

Method

+ void update()	Update the position of the player and check if the releaseMonster KeyCode is triggered. (if it's triggered then calls the releaseMonster() method.
+ boolean isMonsterReady()	
+ void releaseMonster()	
+ getter/setter	

2.4 Class PlayerOne

Extends Entity

This class represents PlayerOne type Player, which can release monsters.

Field

+ int currentPos1	Current position of the player1. (from 0 to 4) Use to set the lane and position of monsters. (set as 2)

Constructor

+ PlayerOne (Monster… monsters)	Initialize PlayerOne and set the following: - set coordinateX - set coordinateY - set currentSprite - set up's KeyCode to W - set down's KeyCode to S - set choose0's KeyCode to 1 - set choose1's KeyCode to 2 - set choose2's KeyCode to 3

	- set releaseMonster's KeyCode to SpaceBar - set coin from SimulationManager class - initialize usedMonsterList Initialize new deck from the inputted array.

Method

+ void draw(GraphicsContext gc)	Draw the PlayerOne sprite in the right position.
+ boolean isMonsterReady()	Check if the player's coins are enough for the selected monster.
+ void releaseMonster()	- Set releaseMons as true. - Decrease the coins. - Remove monster that is released from deck, add a random monster - Add the monster that was released to the monsterList.
+ void checkDeck()	Set the current image of monster channel.
+ void update()	- Update the position of the player, chosen monster, coins and current position of PlayerOne - If releaseMons is true then releases the monster that PlayerOne chooses. - Update all monsters that PlayerOne releases.

2.5 Class PlayerTwo

Extends Entity

This class represents PlayerTwo type Player, which can release monsters.

Field

+ int currentPos2	Current position of the player2. (from 0 to 4) Use to set the lane and position of monsters. (set as 2)

Constructor

+ PlayerTwo (Monster… monsters)	Initialize PlayerTwo and set the following: - set coordinateX - set coordinateY - set currentSprite - set up's KeyCode to UP - set down's KeyCode to DOWN - set choose0's KeyCode to Equals - set choose1's KeyCode to MINUS - set choose2's KeyCode to 0 - set releaseMonster's KeyCode to LEFT - set coin from SimulationManager class - initialize usedMonsterList Initialize new deck from the inputted array.

Method

+ void draw(GraphicsContext gc)	Draw the PlayerTwo sprite in the right position.
+ boolean isMonsterReady()	Check if the player's coins are enough for the selected monster.
+ void releaseMonster()	- Set releaseMons as true. - Decrease the coins. - Remove monster that is released from deck, add a random monster - Add the monster that was released to the monsterList.
+ void checkDeck()	Set the current image of monster channel.
+ void update()	- Update the position of the player, chosen monster, coins and current position of PlayerTwo. - If releaseMons is true then releases the monster that PlayerTwo chooses. - Update all monsters that PlayerTwo releases.

3 Package entity.base

This package contains all the monsters in this game.

3.1 abstract Class Monster

Extends Entity

This class represents Monster type Entity, which can be release by player.

Field

- long lastTimeTriggered	Used to compare with time in animation timer when 1 second pass.
# String sender	The sender of this monster.
# String name	The name of this monster.
# int attack	The attack value of this monster.
# double speed	The speed of this monster.
# int cost	The cost of this monster.
# boolean isDeath	This will be set as true if this monster is dead, as false if it's still alive.
# boolean isAttacking	This will be set as true if this monster is attacking, otherwise this will be false.
# boolean isExist	This will be set as true if this monster is existed, as false if it's not.
# double posX	The position of the monster in X-axis.
# double posY	The position of the monster in Y-axis.
# boolean toSetY	This field is for set the posY. (set as true)
# int currentTime	The current time of each monster from when it's released. (set as 0)
# int lane	The lane that the monster is released.
# int attackCount1	The value of damage this monster deals to barrel for PlayerOne. (set as 0)
# int attackCount2	The value of damage this monster deals to barrel for PlayerTwo. (set as 0)

Constructor

+ Monster(String sender)	initialize monster and set its sender.
+ Monster(String sender, String name, int attack, double speed, int cost)	initialize monster: - set sender, name, attack, speed, cost.

	- set isDeath as false. - set isAttackking as false. - set isExist as true. - set z to 100. - set posX to 0. Initialize new AnimationTimer() for this monster and start.

Method

+ void addMonsterToLaneList(int lane, String sender, Monster monster)	This method add monster to the specific lane and sender list.
+ getter/setter	

3.2 Class Adventurer

Extends Monster

This class represents Adventurer type Monster, which can be release by player.

Constructor

+ Adventurer (String sender)	initialize the default adventurer and set its sender. Default adventurer : - name = “Adventurer” - attack = 2 - speed = 0.5 - cost = 20
+ Adventurer (String sender, String name, int attack, double speed, int cost)	initialize the adventurer with the specific stats.

Method

+ void draw (GraphicsContext gc)	Draw the Adventurer from posX and posY for each player in 5 actions. - Run - Killing - Idle

	- Attack - Death
+ void update()	Update the position of adventurer. (posX)
+ getter/setter	

3.3 Class Ninja

Extends Monster

This class represents Ninja type Monster, which can be release by player.

Constructor

+ Ninja (String sender)	initialize the default ninja and set its sender. Default ninja : - name = “Ninja” - attack = 3 - speed = 1.5 - cost = 5
+ Ninja (String sender, String name, int attack, double speed, int cost)	initialize the ninja with the specific stats.

Method

+ void draw (GraphicsContext gc)	Draw the Ninja from posX and posY for each player in 3 actions. - Run - Attack - Death
+ void update()	Update the position of ninja. (posX)
+ getter/setter	

3.4 Class Knight

Extends Monster

This class represents Knight type Monster, which can be release by player.

Constructor

+ Knight (String sender)	initialize the default knight and set its sender.

	Default knight : - name = “Knight” - attack = 5 - speed = 0.6 - cost = 6
+ Knight (String sender, String name, int attack, double speed, int cost)	initialize the knight with the specific stats.

Method

+ void draw (GraphicsContext gc)	Draw the Knight from posX and posY for each player in 3 actions. - Run - Attack - Death
+ void update()	Update the position of knight. (posX)
+ getter/setter	

3.5 Class Robot

Extends Monster

This class represents Robot type Monster, which can be release by player.

Constructor

+ Robot (String sender)	initialize the default robot and set its sender. Default robot : - name = “Robot” - attack = 4 - speed = 0.7 - cost = 9
+ Robot (String sender, String name, int attack, double speed, int cost)	initialize the robot with the specific stats.

Method

+ void draw (GraphicsContext gc)	Draw the Robot from posX and posY for each player in 3 actions.

	- Run + Attack (draw muzzle and bullet) - Attack - Death
+ void update()	Update the position of robot. (posX)
+ getter/setter	

3.6 Class Santa

Extends Monster

This class represents Santa type Monster, which can be release by player.

Constructor

+ Santa (String sender)	initialize the default santa and set its sender. Default santa : - name = “Santa” - attack = 0 - speed = 0.65 - cost = 6
+ Santa (String sender, String name, int attack, double speed, int cost)	initialize the santa with the specific stats.

Method

+ void draw (GraphicsContext gc)	Draw the Santa from posX and posY for each player in 3 actions. - Run - Jump (get money) - Death
+ void update()	Update the position of santa. (posX)
+ getter/setter	

4 Package exception

4.1 Class UnselectedException

Extends Exception

This class contains exception that use to throw message to pregame pane.

Field

+ message	Alert Text from constructor

Constructor

+ UnselectedException(String message)	Set message to field

5 Package gui

This package contains pane that represents user interface.

5.1 Class MenuPane

Extends StackPane

This class represents pane that shows menuscreen at the start of the game.

Field

- Button startBtn	Button that contains text “start”. When is clicked, the screen changes to pregame session
- Button HowToPlayBtn	Button that contains text “How to play”. When is clicked, the screen changes to how to play screen.
- Button quitBtn	Button that contains text “quit”. When is clicked, game is closed.
- VBox butttonpane	Contains startBtn, HowToPlayBtn and quitBtn accordingly in a vertical alignment.
- ImageView gametitleview	Used to represent gametitle

Constructor

+ MenuPane()	initialize button pane and gametitle

Method

+ void intiButtonPane()	initialize button pane , startBtn , HowToPlayBtn and quitBtn.
+ void initGameTitle()	initialize gametitleview
+ getter	

5.2 Class PreGamePane

Extends StackPane

This class represents pane that shows pre-game session after clicking start button.

Field

- Label time	label with text “time”
- Label coin	label with text “coin”
- CheckBox time90 , time120 , time180	Time check box to select start-game time - time90 -> 90 sec - time120 -> 120 sec - time180 -> 180 sec
- CheckBox coin0 , coin10 , coin20	Coin check box to select start-game coins - coin0 -> 0 coin - coin10 -> 10 coins - coin20 -> 20 coins
- Button startBtn	Start button
- Button backBtn	Back button
- Tooltip tooltip	Show alert when tick in each box is out of condition
- Label description	Label that shows description
- HBox timePane	Contains time label and time checkboxes accordingly in horizontal alignment.
- HBox coinPane	Contains coin label and coin checkboxes accordingly in horizontal alignment.
- HBox buttonPane	Contains start button and back button horizontally.
- VBox form	Contain time pane, coin pane and button pane.
- boolean anyTimeBoxSelected	True : when any time boxes is selected False : when none of time boxes is selected
- boolean anyCoinBoxSelected	True : when any coin boxes is selected

	False : when none of coin boxes is selected

Constructor

+ PreGamePane()	Initialize time pane , coin pane , button pane and form

Method

+ void initTimePane()	Init time pane , time label and 3 time checkboxes
+ void initCoinPane()	Init coin pane , coin label and 3 coin checkboxes.
+ void initButtonPane()	Init button pane ,start button, tooltip and back button.
+ void initForm()	Initialize form
+ void setupTimeSelectedLogic	- Set time boxes that can choose only one box in each pane - when each box is selected then set start game time of each box
+ void setupCoinSelectedLogic	Set coin boxes that each pane can choose only one box
+ void setAlertText(String message)	- set message to tooltip - set tooltip to start button
+ Boolean isAllTick() throw UnselectedException()	True : if one of time box is selected and one of coin box is selected. False : if either time box or coin box is not selected , then throw an exception
+ getter/setter of some field	Return that field

5.3 Class HowToPlayPane

Extends StackPane

This class represents pane that shows after clicking How to play button.

Field

- Label instruction11, instruction12	Display the player1's keyboard control of the game

- Label instruction21, instruction22	Display the player2's keyboard control of the game
- ImageView playerOneImgv , playerTwoImgv	players icon images
- HBox playerOnePane , playerTwoPane	Contains icon image and keyboard controller label
- Button backBtn	Back button
- Vbox playerPane	Contains playerOnePane , playerTwoPane and back button

Constructor

+ HowToPlay()	Call these methods - loadResource() - initPlayerOnePane() - initPlayerTwoPane() - initBackButton() - initPlayerPane()

Method

- void loadResource()	Load player1 image and player2 image (reverse image for player2)
+ void initPlayerOnePane()	Initialize playereOnePane and player1's instruction labels
+ void initPlayerTwoPane()	Initialize playerTwoPane and player2's instruction labels
+ void initBackButton()	Initialize back button
+ void initPlayerPane()	Initialize player pane
+ Button getBackBtn()	Return backBtn

5.4 Class GamePane

Extends VBox

This class also contains TimePane ,GameScreen and QuitPane. It all components that displays on

Field

- TimePane timePane	shows deck , coin , point and time.

- StackPane gamescreenStack	Contains gamescreen and result (when game is over)
- GameScreen gamescreen	Displays the game animation.
- QuitPane quitPane	Shows menu button to quit game to main menu
- GameResult gameResult	Shows result when game is over

Constructor

+ GamePane(int width , int height)	Init all fields

Method

+ void paintComponent()	- Paint all component on entire screen from time pane and game screen. - if game is over , display result screen.
+ void paintResult()	displays result screen

5.5 Class TimePane

Extends HBox

This class also displays game current time, points and ready-monster in deck

Field

- Label timeChannel	Displays game current time
- Label pointchannel1	Displays player1 points
- Label pointChannel2	Displays player2 points
- Label monsterChannel1	Displays ready-monster in player1 deck
- Label monsterChannel2	Displays ready-monster in player2 deck

Constructor

+ TimePane(int width , int height)	Init all fields

Method

+ void initMonsterChannel(int width , int height)	Init monsterChannel1 and monsterChannel2
+ void initTimeChannel(int width , int height)	Init timeChannel

+ void initPointChannel(int width , int height)	Init pointChannel1 and pointChannel2
+ void paintComponent()	Set Text in time channel to match current time

5.6 Abstract Class MonsterChannel

Extends HBox

This class represents ready-monsters in deck.

Field

# HBox deckChannel	Shows ready-monsters in deck
# Label coinsChannel	Shows current player's coin text
# StackPane card0, card1 , card2	Contains monster's icon image. Monster's image in the game refer to monster that can be released at that time

Constructor

+ MonsterChannel(double width , double height)	Initialize all fields

Method

+ void initCard(double width)	Initialize card0 , card1 and card2
+ void setCard(StackPane card , int width)	Set card's properties
+ void initDeckChannel(double width)	Initialize deckChannel
+ void initCoinChannel(double width)	Initialize coinChannel
+ void highlight()	Set green color to background
+ void unhighlight()	Set white color to background
+ *void checkselected()*	
+ *void editImage()*	
+ void paintComponent()	Update and Set all component that displays in time pane (not including setting current coin)

5.7 Class MonsterChannel1

Extends MonsterChannel

This class represents ready-monsters in player1's deck

Constructor

+ MonsterChannel1(double width , double height)	- init all fields - set player1's current icon image in the deck - add cards to deckChannel in player1 pattern

Method

+ void checkSelected()	Highlight the card which player1 choose at that moment
+ void editImage()	Set player1's cards image in deckChannel to related to current card in the deck.
+ void paintComponent()	Update and Set all component that displays in time pane (including setting player1's current coin)

5.8 Class MonsterChannel2

Extends MonsterChannel

This class represents ready-monsters in player2's deck

Constructor

+ MonsterChannel1(double width , double height)	- Initialize all fields - set player2's current icon image in the deck - add cards to deckChannel in player2 pattern

Method

+ void checkSelected()	Highlight the card which player2 choose at that moment
+ void editImage()	Set player1's cards image in deckChannel to related to current card in the deck.
+ void paintComponent()	Update and Set all component that displays in time pane (including setting player2's current coin)

5.9 Class GameScreen

Extends Canvas

This class represents all component in canvas

Constructors

+ GameScreen(int width , int height)	- Set listener to this class

	- set height and width

Method

+ void addlistener()	- set key-pressed logic when pressed keyboard - set key-released logic when release keyboard
+ void paintComponent()	draw all components that must show in the canvas at that time

5.10 Class QuitPane

Extends HBox

This class contains menu button that used to get back to main menu

Field

- Button menuBtn	- Menu button - getting back to main menu when clicked

Constructor

+ QuitPane()	Initialize field

Method

+ void initMenuBtn	Initialize menu button

5.11 Class GameResult

Extends VBox

This class displays result of the game when game is over.

Field

- Label winner	Shows the result text (if no one win result can be “DRAW”)
- Button menuBtn	- menu button - getting out to main menu when is clicked

Constructor

+ GameResult(int width , int height)	Init all fields

Method

+ void initWinner()	Init winner label and set game result to label's text
+ void initMenuBtn()	- Init menu button

6 Package Input

This package contains InputUtility class that helps to manage input from keyboard.

6.1 Class InputUtility

This class helps other class to notice that what keyboard buttons are pressed at the moment

Field

- ArrayList<KeyCode> keyPressed	Contains keyboard buttons that is pressed at that time

Constructor

+ Boolean getKeyPressed(KeyCode keycode)	- If keyboard button is pressed at that time , return true - else , return false
+ void setKeyPressed(KeyCode keycode , boolean pressed)	- Add keycode to keyPressed List if that keycode is pressed at that time - if not , remove keycode from list.

7 Package logic

7.1 Class Deck

This class manage player's ready-monsters logic.

Field

- ArrayList<Monster> monsterList	Contains monster that is unavailable to release
- ArrayList<Monster> deckList	Contains monsters that ready to release

Constructor

+ Deck([]Monster monsters)	- add 3 first monsters to deck

	- add 2 latest monsters to monsterList

Method

+ Monster getMonsterinDeckList(int i)	Return monster No. i in deck
+ void addRandomMonster()	Random monster in monsterList and add to deck
+ Monster removeMonster(int i)	Remove monster No.i in deck

7.2 Class Field

This class represents game background

Implements IRenderable

Method

+ int getZ()	Return z value
+ draw(GraphicsContext gc)	Draw game background

7.3 *Abstract Class PlayerField*

This class represents door and switch the gameplay screen.

Implements IRenderable

Field

# int X	coordinateX
# int Z	coordinateZ
# int width	The width of a field.
# int height	The height of a field.

Constuctor

+ PlayerField(double width , double height)	- set width - set length - set Z = -5000

Method

+ *draw(GraphicsContext gc)*	
+ int getZ()	Return Z

7.4 Class PlayerField1

This class represents door and switch of player1

Extends PlayerField

Constructor

+ Player1Field(double width , double height)	- call super class - set X to the proper size. - set X and Y in SimulationManager.

Method

+ draw(GraphicsContext gc)	- check win-lose state of each lanes of player1 and draw door and switch.

7.5 Class PlayerField2

This class represents door and switch of player2

Extends PlayerField

Constructor

+ Player2Field(double width , double height)	- call super class - set X to the proper size. - set X and Y in SimulationManager.

Method

+ draw(GraphicsContext gc)	- check win-lose state of each lanes of player2 and draw door and switch.

7.6 Class GameLogic

This class initializes and updates PlayerOne, PlayerTwo and all barrels. This class also can check if the game is end.

Field

- Field field	The field of the game.
- PlayerOne playerOne	The player on the left side.
- PlayerField playerField1	Field for PlayerOne.
- PlayerTwo playerTwo	The player on the right side.
- PlayerField playerField2	Field for PlayerTwo.
- Monster adventurer, knight, ninja, santa, robot	Monster in this game.

- Barrel barrel1, barrel2, barrel3, barrel4, barrel5	The barrels at the middle of every lane.

Constructor

+ GameLogic()	- Set up the SimulationManager - Call loadmap() - Initialize all monsters. - Initialize Barrels at all lanes. - Initialize players and player's fields. - Initialize field.

Method

# void addNewObject(Entity entity)	Add entity to RenderableHolder.
+ void logicUpdate()	Update the following: - playerOne - playerTwo - all barrels - check if the game is end. - If game is end, all monsters on the lane die.
+ void checkGameEnd()	Divided to 2 cases: - (time up) player with higher point win. If the score is equal, the player who deal more damage win. - The first player who get three point is a winner.

8 Package sharedObject

8.1 interface Class IRenderable

Method

+ int getZ()	Get z value
+ void draw(GraphicsContext gc)	draw

8.2 Class ImageAnimationManager

This class store the many frames image to make an animation.

Field

The fields of this class is the image of each frame in the animation.

Method

+ void loadAnimationImage()	- Initialize all images - add each group of images to the array list
+ getter/setter	

8.3 Class RenderableHolder

This class manage entities that must show in canvas and stores images that are not animation image

Field

- final RenderableHolder instance	Object of RenderableHolder class
- Map<String , ImageView> map1	Map that contains monster's name as key and monster's image view as values (Player1 only)
- Map<String , ImageView> map2	Map that contains monster's name as key and monster's image view as values (Player2 only)
- Image playerOneSprite	Player1's player image
- Image playerTwoSprite	Player2's player image
- Image backgroundImg	Game background iamge
- Image wallBackground	Time pane background image
- Image menuBackground	Menu background image
- Image gameResultBackground	Result background image
- Image barrel	Barrel image
- Image switchOn	Switch on image
- Image switchOff	Switch off image
- Image doorOpen	Door open image
- Image doorClose	Door close image
- Image adventureIcon	Adventurer image in deck slot
- Image knightIcon	knight image in deck slot
- Image ninjaIcon	ninja image in deck slot

- Image santaIcon	santa image in deck slot
- Image robotIcon	robot image in deck slot
- WriteableImage background	Cropped background image
- ImageView currentImage10	Contains current image of monster index No. 0 in player1's deck
- ImageView currentImage11	Contains current image of monster index No. 1 in player1's deck
- ImageView currentImage12	Contains current image of monster index No. 2 in player1's deck
- ImageView currentImage20	Contains current image of monster index No. 0 in player2's deck
- ImageView currentImage21	Contains current image of monster index No. 1 in player2's deck
- ImageView currentImage22	Contains current image of monster index No. 2 in player2's deck
- List<IRenderable> entities	Collect all conponent that must show in canvas at that current time
- Comparator<IRenderable> comparator	Compare Z value between two entities (which one has smaller value will be taken Iist first)

Constructor

+ RenderableHolder()	Init entities and comparator

Method

+ void loadResource()	Init all images
+ loadMap()	Init map1 and map2
+ ImageView getMonsterImage1(String name)	Return monster's image for player1
+ ImageView getMonsterImage2(String name)	Return monster's image for player2
+ void add(IRenderable entity)	Add entity to list that collect all entity at that time.
+ Renderable getInstance()	Return object of RenderableHolder class
+ IRenderable getEntities()	Return entities (list that collect all entity at that moment)
+ gettter/setter of some fields	

8.4 Class SimulationManager

This class stores the miscellaneous fields and methods.

Field

- int inGameTime	The time in this game.
- boolean isGameEnd	If the game is end, this will be set as true.
- boolean isBackToMenu	If the player triggers Menu button, this will be set as true.
- String game Result	The string that tells which player is a winner.
- int points1, points2	The point of the players.
- int isSelected1, isSelected2	The number that tells which monster the player is choosing at the moment.
- int XP1, XP2	The coordinate of each player.
- ArrayList<Integer> Y	The array list that keeps all coordinateY.
- int coins1, coins2	The money for each player to release monsters.
- String WonLane1 -> WonLane5	The string that tell who is the winner of the specific lane.
- ArrayList<Monster> P1Lane1	The array list that keeps all PlayerOne's monsters in the first lane.
- ArrayList<Monster> P1Lane2	The array list that keeps all PlayerOne's monsters in the second lane.
- ArrayList<Monster> P1Lane3	The array list that keeps all PlayerOne's monsters in the third lane.
- ArrayList<Monster> P1Lane4	The array list that keeps all PlayerOne's monsters in the forth lane.
- ArrayList<Monster> P1Lane5	The array list that keeps all PlayerOne's monsters in the fifth lane.
- ArrayList<ArrayList<Monster>> P1AllMonster	The array list that keeps all lane list of PlayerOne.
- ArrayList<Monster> P2Lane1	The array list that keeps all PlayerTwo's monsters in the first lane.
- ArrayList<Monster> P2Lane2	The array list that keeps all PlayerTwo's monsters in the second lane.
- ArrayList<Monster> P2Lane3	The array list that keeps all PlayerTwo's monsters in the third lane.

- ArrayList<Monster> P2Lane4	The array list that keeps all PlayerTwo's monsters in the forth lane.
- ArrayList<Monster> P2Lane5	The array list that keeps all PlayerTwo's monsters in the fifth lane.
- ArrayList<ArrayList<Monster>> P2AllMonster	The array list that keeps all lane list of PlayerTwo.
- ArrayList<Barrel> barrelList	The array list that keeps all barrels.
+ String fontUrl	The url of font.

Method

+ void setFont(Labeled label, int size)	Set the font.
+ void setUpSimulationManager()	Initialize the following: - The selected monster to 0. - The point to 0. - GameEnd and BackToMenu to false. - All array lists. (add sub array to the main array) - set all WonLanes as "none"
+ int getTotalDamage1()	Sum up damage from PlayerOne.
+ int getTotalDamage2()	Sum up damage from PlayerTwo.
+ void setCoin1Increase(int x)	Add playerOne's coin.
+ void setCoin2Increase(int x)	Add playerTwo's coin.
+ void setCoin1Decrease(int x)	Remove playerOne's coin.
+ void setCoin2Decrease(int x)	Remove playerTwo's coin.
+ void setDecreaseInGameTime()	Decrease time in game.
+ getter/setter	