Hub99 Documentation

Created by

Ratchanon Wattanataweekul 6231354721
Alongkot Intaragumhang 6231373621

2110215 Programming Methodology
Semester 2 Year 2020
Chulalongkorn University

Hockey Game

Introduction

The Hockey Game is actual ice hockey game which is created by 2 novice programmers. The objective of this game is to get more scores compared to other players by hitting a ball into rival's goal.

Rules

In the beginning, an only objective is to hit a puck to other goals, but there is an item to spawn by 25% chance. When a ball hitting with the item, a ball would change from a NormalBall to a special one, BombBall. In this case, BombBall always has 10 seconds before it explodes. A player who can push a BombBall to other side of the field gets a score.

Both players can choose the special skills either FreezingSkill or BlockingSkill.

Each game has 60 seconds to play before it ends.

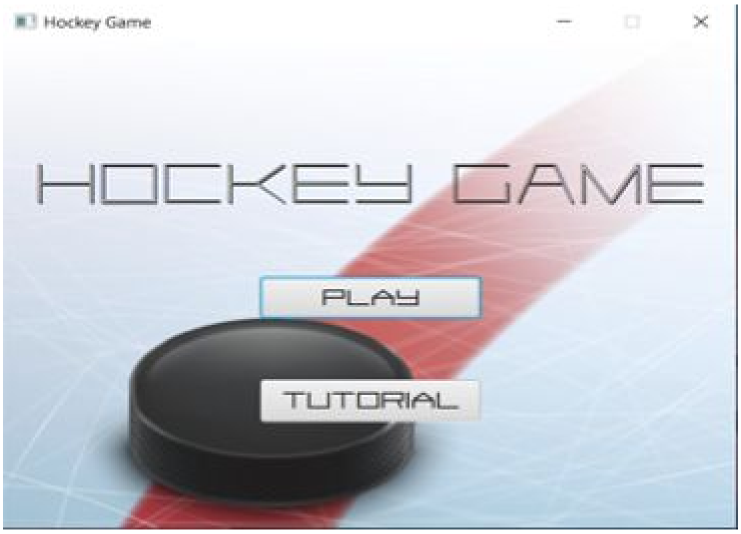
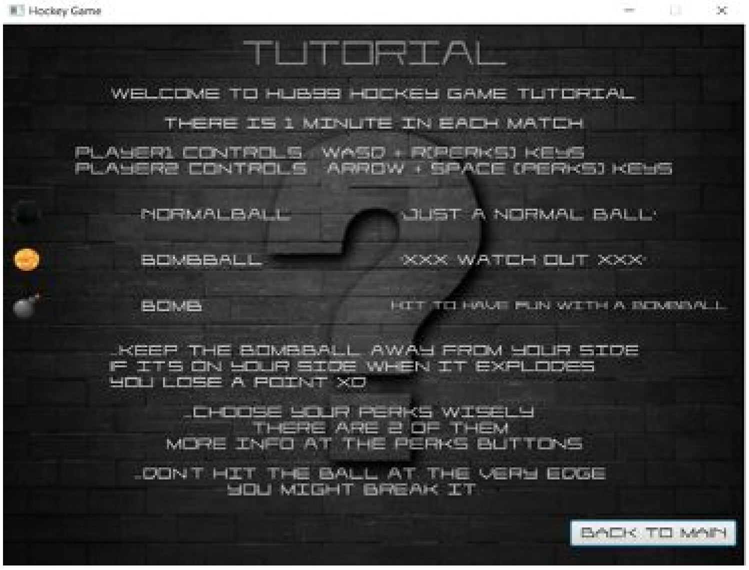


Fig.1 Welcome Scene

This scene provides 2 button for play game and to read the tutorial for a game.



TUTORIAL

WELCOME TO HUB99 HOCKEY GAME TUTORIAL

THERE IS 1 MINUTE IN EACH MATCH

PLAYER1 CONTROLS WASD + R(PERKS) KEYS
PLAYER2 CONTROLS ARROW + SPACE (PERKS) KEYS

NORMALBALL JUST A NORMAL BALL·

BOMBBALL ·XXX WATCH OUT XXX·

BOMB HIT TO HAVE FUN WITH A BOMBBALL

.KEEP THE BOMBBALL AWAY FROM YOUR SIDE
IF ITS ON YOUR SIDE WHEN IT EXPLODES
YOU LOSE A POINT XD

.CHOOSE YOUR PERKS WISELY
THERE ARE 2 OF THEM
MORE INFO AT THE PERKS BUTTONS

.DON'T HIT THE BALL AT THE VERY EDGE
YOU MIGHT BREAK IT

Fig.2 Tutorial Scene

This scene clearly explains all rules and how to play this game.

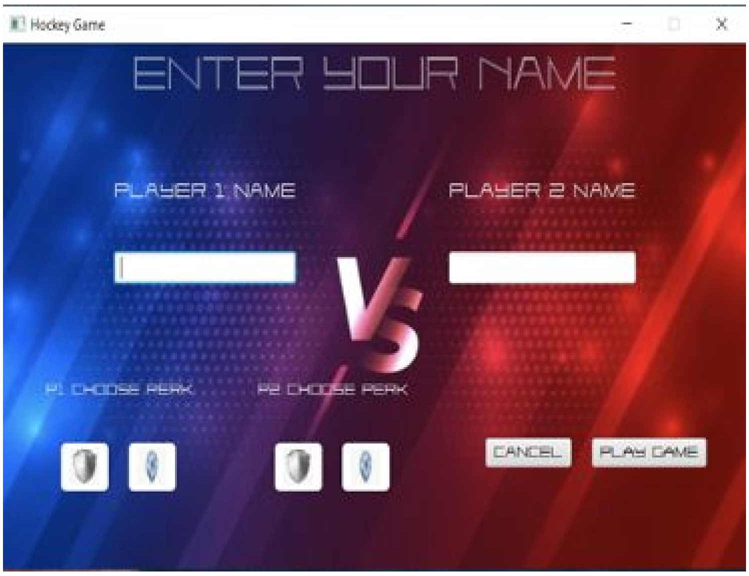


Fig.3 Player Scene

This scene provides 2 boxes to insert players’ names and buttons for both players to choose their skills in order to get advantages of the game. Both players can either go back to menu or play the game.

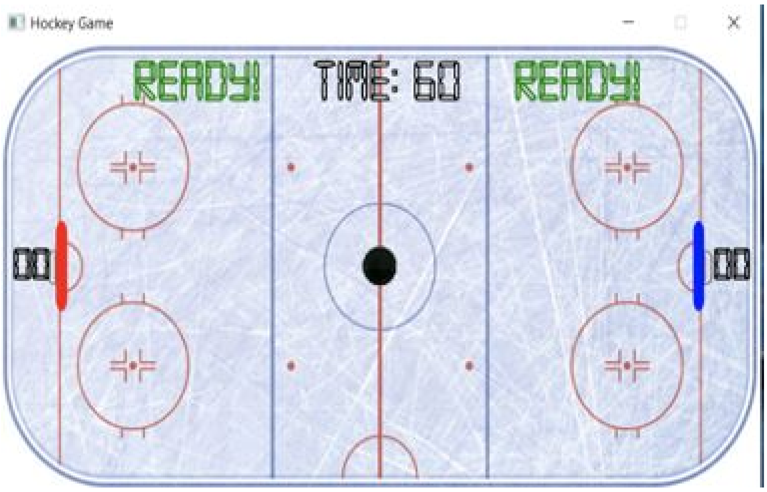


Fig.4 Game Scene

This scene is for players to play their game, the only objective is to get more score than another player.

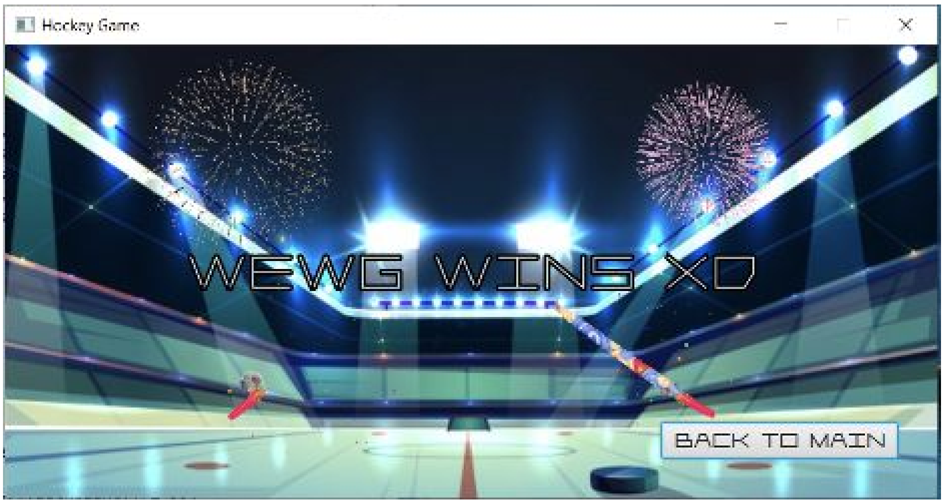


Fig.5 Winning Scene

This last scene is for announcing the winner of the game. You can go back to menu.

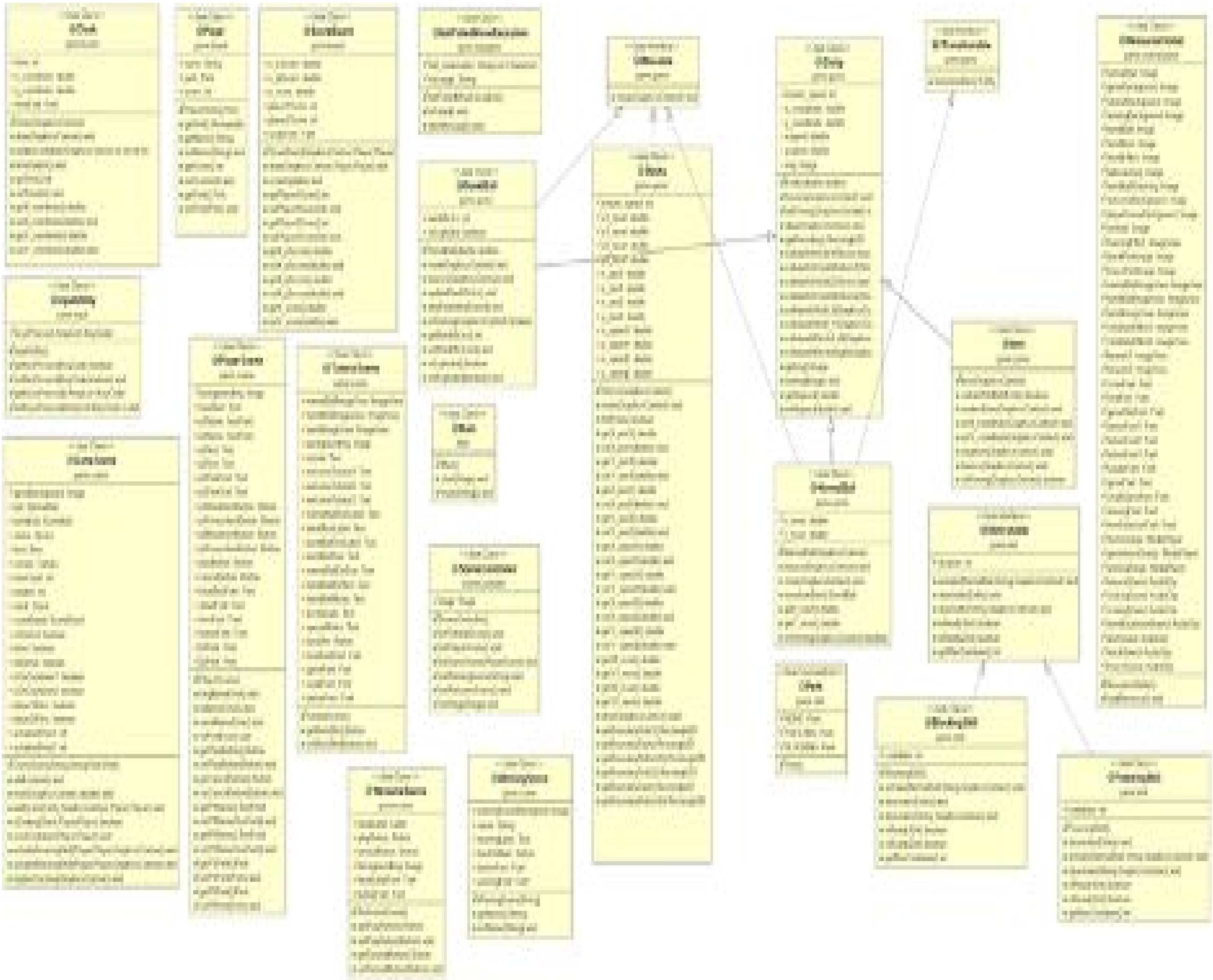


Fig.6 Overall Class Diagram

1. Package game.board

1.1 Class Clock

1.1.1 Fields

- int time;	- Game’s duration = 60 seconds
- double x_coordinate;	- X-position of a clock in the scene
- double y_coordinate;	- Y-position of a clock in the scene
- Font timeFont;	- Set the font with timeFont for a clock from ResourceHolder

1.1.2 Constructor

+ Clock(GraphicContext gc)	- Set XY-position of a clock with GraphicContext

1.1.3 Methods

+ void draw(GraphicsContext gc)	- Set linewidth = 1.5 - Set font with timeFont - Draw a clock in the scene with GraphicContext in XY-position

+ void updateCooldown(GraphicsContext gc, int maxCooldown1, int cooldown1, int maxCooldown2, int cooldown2)	- Calculate the durations with both pointers - Draw skills' cooldowns of both players in the proper position - Set Color.GREEN when cooldown is ready, else set Color.CHOCOLATE
+ void timeUpdate()	- Subtract time variable by 1
Generate Getters/Setters	

1.2 Class Player

1.2.1 Fields

- String name;	Player's name
- Perk perk;	Player's skill = Perk.NONE
- int score	Player's score = 0

1.2.2 Constructor

+ Player(String name, Perk perk)	- Set name and perk

1.2.3 Methods

+ IActivatable getSkill()	- Return skill which matches with player’s perk
Generate Getters/Setters	

1.3 Class ScoreBoard

1.3.1 Fields

- double x_p1score;	- X-position of player1’s sore
- double x_p2score;	- X-position of player2’s sore
- int player1Score;	- Player1’s score
- int player2Score;	- Player2’s score
- Font scoreFont;	- Set font with scoreFont from ResourceHolder for scoreboard

1.3.2 Constructor

+ ScoreBoard(GraphicsContext gc, Player player1, Player player2)	- Set both players’ score - Set XY-position for scoreboorad

1.3.3 Methods

+ void draw(GraphicsContext gc, Player player1, Player player2)	- Set linewidth = 1.5 - Set font - Draw both players' score in the XY-position
Generate Getters/Setters	

2. Package game.exception

2.1 Class ballFailedMoveException

2.1.1 Fields

+ static ArrayList<Character> ball_movements;	- Collect ball's movements
+ static String message;	- Message when a ball fails to move - "failed movement occurs"

2.1.2 Constructor

+ static void isFailed()	- Checking movements of ball in ball_movements. - If the same movement occurs twice in a time, throw this exception.

2.1.3 Methods

+ void printMessage()	- Print message

3. Package game.game

3.1 Class BombBall extends Entity implements IMovable

3.1.1 Fields

- int bombTicks;	- BombBall’s duration before explodes

3.1.2 Constructor

+ BombBall(double x, double y)	- Set XY-positon - Set BombBall image from ResourceHolder - Set bombTicks = 10 - Set isExplode = false

3.1.3 Methods

+ void move(GraphicsContext gc)	- Draw movements of BombBall
+ void bounce(GraphicsContext gc)	- Set speed of BombBall to the opposite when BombBall hit the walls - Collect BombBall's movements in ballFailedMoveException
+ void updateBombTicks()	- Subtract bombTicks variable by 1
+ void playExplodingSound()	- If BombBall is not exploded yet and bombTicks = 0, play explosion sound from ResourceHolder and set isExplode = true

+ boolean isWinning(GraphicsContext gc)	- Return if bombTicks <= 0 and bombTicks > -3
Generate Getters/Setters	

3.2 abstract Class Entity

3.2.1 Fields

# final int instant_speed;	- Instant speed of Entity = 6
# double x_cooridinate;	- X-position of Entity
# double y_cooridinate;	- Y-position of Entity
# double x_speed;	- X-axis speed of Entity
# double y_speed;	- Y-axis speed of Entity
# Image img;	- Image of Entity

3.2.2 Constructor

+ Entity(double x, double y)	- Set XY-positon

3.2.3 Methods

+ abstract void bounce(GraphicsContext gc)	- Bounce Entity when hitting the walls
+ abstract boolean isWinning(GraphicsContext gc)	- Return if Entity get in the winning condition
+ void draw(GraphicsContext gc)	- Draw Entity with XY-position
+ Rectangle2D getBoundary()	- Return Rectangle2D covering entire Entity
+ boolean collideWithSide1(Sticks other)	- Return if Entity hits with stick1's side position
+ String collideWithTopOrButtom1(Sticks other)	- Return whether top or bottom which stick1 hits Entity
+ boolean collideWithSide2(Sticks other)	- Return if Entity hits with stick2's side position

+ String collideWithTopOrBottom2(Sticks other)	- Return whether top or bottom which stick1 hits Entity
+ String collidewithWall_X(GraphicsContext gc)	- Return whether top or bottom which Entity hits the walls
+ boolean collidewithWall_Y(GraphicsContext gc)	- Return if Entity hits Y-walls
+ boolean collidewithBricksLeft(GraphicsContext gc)	- Return if Entity hits left special wall
+ boolean collidewithBricksRight(GraphicsContext gc)	- Return if Entity hits right special wall
Generate Getters/Setters	

3.3 interface IMovable

3.3.1 Method

+ void move(GraphicContext gc)	- move object

3.4 Class Item extends Entity

3.4.1 Constructor

+ Item(GraphicsContext gc)	- Set X-positon = 0 - Set Y-postion = -50 - Set Item’s image from ResourceHolder

3.4.2 Methods

+ boolean collideWithBall(Entity other)	- Return if item get hit by another Entity
+ void randomDraw(GraphicsContext gc)	- Draw item in the given XY-position
+ void setX_coordinate(GraphicsContext gc)	- Set X-position by random from proper range
+ void setY_coordinate(GraphicsContext gc)	- Set Y-position by random from proper range

+ void clearItem(GraphicsContext gc)	- Clear exist item - Set XY-position out of the scene
+ void bounce(GraphicsContext gc)	- Always do nothing here
+ boolean isWinning(GraphicsContext gc)	- Always return false

3.5 Interface ITransitionable

3.5.1 Method

+ Entity transition(Item item)	- Change form of Entity

3.6 Class NormalBall extends Entity implements IMovable, ITranstionable

3.6.1 Fields

- final double x_reset;	- X-position in middle of the scene

- final double y_reset;	- Y-position in middle of the scene

3.6.2 Constructor

+ NormalBall(GraphicsContext gc)	- Set XY-position in middle of the scene - Set both x_reset and y_reset with proper value - Set NormalBall image from ResourceHolder - Draw a NormalBall in the scene

3.6.3 Methods

+ void bounce(GraphicsContext gc)	- Set speed of NormalBall to the opposite when NormalBall hit the walls - Collect NormalBall's movements in ballFailedMoveException
+ void move(GraphicsContext gc)	- Draw movements of NormalBall

+ BombBall transition(Item item)	- Return BombBall with NormalBall's speeds
+ boolean isWinning(GraphicsContext gc)	- Return if NormalBall hits Y-walls
Generate Getters	

3.7 Class Sticks extends IMovable

3.7.1 Fields

- final int instant_speed;	- Sticks' speed = 3
- final double x1_reset;	- X-position in the left side of the scene
- final double x1_reset;	- Y-position in the left side of the scene
- final double x1_reset;	- X-position in the right side of the scene
- final double x1_reset;	- Y-position in the right side of the scene
- double x_pos1;	- Stick1's X-position
- double y_pos1;	- Stick1's Y-position

- double x_pos2;	- Stick2’s X-position
- double y_pos2;	- Stick2’s Y-position
- double x_speed1;	- Stick1’s X-speed
- double y_speed1;	- Stick1’s Y-speed
- double x_speed2;	- Stick2’s X-speed
- double y_speed2;	- Stick2’s Y-speed

3.7.2 Constructor

+ Sticks(GraphicsContext gc)	- Set XY-position of both sticks - Draw both sticks

3.7.3 Methods

+ void move(GraphicsContext gc)	- Set speed of both sticks when pressing wrong keys - Set speed of both sticks in the directions of the keys - Draw both sticks

+ boolean hit(Entity ball)	- If sticks get hit by Entity, set speed of Entity to the opposite and return true, else return false
+ void draw(GraphicsContext gc)	- Draw both sticks
+ Rectangle2D getBoundarySide1()	- Return Rectangle2D covering both sides of stick1
+ Rectangle2D getBoundaryTop1()	- Return Rectangle2D covering top side of stick1
+ Rectangle2D getBoundaryBottom1()	- Return Rectangle2D covering top side of stick1
+ Rectangle2D getBoundarySide2()	- Return Rectangle2D covering both sides of stick2
+ Rectangle2D getBoundaryTop2()	- Return Rectangle2D covering top side of stick2
+ Rectangle2D getBoundaryBottom2()	- Return Rectangle2D covering top side of stick2
Generate Getters/Setters	

4. Package game.input

4.1 Class InputUtility

4.1.1 Fields

- static ArrayList<KeyCode> keysPressed	- Collect keys pressed

4.1.2 Methods

+ static boolean getKeyPressed(KeyCode keycode)	- Return if keycode is in keysPressed
+ static void setKeyPressed(KeyCode keycode,boolean pressed)	- Add that keycode to keysPressed if that keycode is pressed, else remove that keycode
Generate Getter/Setter	

5. Package game.scene

5.1 Class GameScene extends Stackpane

5.1.1 Fields

- final Image gameBackground	Initialize the game background image.
- NormalBall ball	Normal ball
- BombBall bombBall	Bomb ball
- Sticks sticks	Sticks
- Item item	Bomb item
- Canvas canvas	Canvas
- int timeCount	Time waiting for player to be ready. Initialized to 0
- int random	Chances of an item to spawn.
- Clock clock	Clock
- ScoreBoard scoreBoard	Scoreboard
- boolean isStarted	Start the game boolean. Initialized to false.
- boolean failed	Ball movemet failure boolean. Initialized to false.
- boolean isNormal	Normal state of the game which is when the ball is a Normal Ball. Initialized to true.

- boolean isOnCooldown1	Cooldown state of Player1’s perks. Initialized to false.
- boolean isOnCooldown2	Cooldown state of Player1’s perks. Initialized to false.
- boolean player1Wins	Player1’s victory state. Initialized to false.
- boolean player2Wins	Player2’s victory state. Initialized to false.
- int activatedTime1 = 0	Time that player1’s skill was last used. Initialized to 0.
- int activatedTime2 = 0	Time that player2’s skill was last used. Initialized to 0.

5.1.2 Constructor

+ GameScene(String name1, String name2, Perk perk1, Perk perk2)	- Initialize the remaining fields, including text font and fill settings. - Add all the nodes to the pane. - Create a game loop using amimation timer.

5.1.3 Methods

+ void addListener()	Input listener.
+ void reset(GraphicsContext gc, double toLoser)	Ball and item reset after scoring happens.
+ void addScore(Entity ball, GraphicsContext gc, Player player1, Player player2)	Score controller.
+ boolean isEnding(Clock clock, Player player1, Player player2)	States the end of the game which is when the time is up.
+ void checkCooldown(Player player1, Player player2)	Check each player skill’s cooldown.
+ void activateFreezingSkill(Player player1, Player player2, GraphicsContext gc)	Freezing skill controller.
+ void activateBlockingSkill(Player player1, Player player2, GraphicsContext gc)	Blocking skill controller.
+ void updateCracking(GraphicsContext gc)	Update cracking ball effect.

5.2 Class PlayerScene extends Vbox

5.2.1 Fields

- final Image backgroundImg	Background image. Initialized from the resource holder.
- Text headText	Header of the scene.
- TextField p1Name	Player1’s name input.
- TextField p2Name	Player2’s name input.
- Text p1Text	Player1’s label text.
- Text p2Text	Player2’s label text.
- Text p1PerkText	Player1’s perk label text.
- Text p2PerkText	Player2’s perk label text.
- Button p1BlockItemButton	Player1’s “Block” perk option button.
- Button - p1FreezeItemButton	Player1’s “Freeze” perk option button.
- Button p2BlockItemButton	Player1’s “Block” perk option button.
- Button p2FreezeItemButton	Player1’s “Freeze” perk option button.
- Button playButton	Play the game button.
- Button cancelButton	Go back to main button.
- Font headTextFont	Header font. Initialized from the resource holder.

- Font detailFont	Detail font. Initialized from the resource holder.
- Font itemFont	Item selection font. Initialized from the resource holder.
- Font buttonFont	Button font. Initialized from the resource holder.

5.2.2 Contructor

+ PlayerScene()	- Set the layout of the pane - Initialize the remaining fields, including text font and fill settings. -Set the proper handlers for each button. - Add all the nodes to the pane.

5.2.3 Methods

+ void longNameError()	Shows the alert if the name(s) is/are too long (more that 8 letters).
+ void noNameError()	Shows the alert if the input(s) have/has no name(s).

+ void sameNameError()	Shows the alert if the inputs have the same name.
+ void noPerkError()	Shows the alert if player(s) has/have not select the perk(s).
Generate Getter/Setter	

5.3 Class TutorialScene extends Vbox

5.3.1 Fields

- ImageView normalBallImageView	Normal ball image view. Initialize from ResourceHolder.
- ImageView bombBallImageView	Bomb ball image view. Initialize from ResourceHolder.
- ImageView bombImageView	Bomb image view. Initialize from ResourceHolder.
- Image backgroundImg	Background image. Initialize from ResourceHolder.
- Text tutorial	Head text.
- Text welcomeTutorial1	Part1 of the welcome tutorial text.
- Text welcomeTutorial2	Part2 of the welcome tutorial text.

- Text welcomeTutorial3	Part3 of the welcome tutorial text.
- Text normalBallTextLabel	Normal ball label text.
- Text bombTextLabel	Bomb label text.
- Text bombBallTextLabel	Bomb ball label text.
- Text bombDefText	Bomb definition text.
- Text normalBallDefText	Normal ball definition text.
- Text bombBallDefText	Bombball ball definition text.
- Text bombBallRules	Bombball ball rules text.
- Text itemDetails	Item detail text.
- Text specialRules	Special rules text.
- Button backBtn	Back to main button.
- Font headTextFont	Head text font. Initialize from ResourceHolder.
- Font gameFont	Regular - sized detail font.
- Font smallFont	Small detail font.
- Font buttonFont	Button font.

5.3.2 Constructor

+ TutorialScene()	- Set the layout of the pane - Initialize the remaining fields, including text font and fill settings.

	-Set the proper handlers for each button. - Add all the nodes to the pane.

5.3.3 Methods

Generate Getter/Setter	

5.4 Class WelcomeScene extends Vbox

5.4.1 Fields

- Label headLabel	Head label of the pane.
- Button playButton	Heading to Player Scene button.
- Button tutorialButton	Heading to Tutorial Scene button.
- final Image backgroundImg	Background image.
- Font headLabelFont	Head label font. Initialize from ResourceHolder.
- Font buttonFont	Button font. Initialize from ResourceHolder.

5.4.2 Constructor

+ WelcomeScene()	- Set the layout of the pane - Initialize the remaining fields, including text font and fill settings. -Set the proper handlers for each button. - Add all the nodes to the pane.

5.4.3 Methods

Generate Getter/Setter	

5.5 Class WinningScene extends StackPane

5.5.1 Fields

- Image winningSceneBackgroud	Backgroundimage.
- String name	Name of the winner.
- Text winningLabel	Winner annoucement text.

- Button backToMain	Back to main scene button.
- Font buttonFont	Button font.
- Font winningFont	Winner announcement text’s font.

5.5.2 Constructor

+ WinningScene(String name)	- Set the layout of the pane - Initialize the remaining fields, including text font and fill settings. -Set the proper handlers for each button. - Add all the nodes to the pane.

5.5.3 Methods

Generate Getter/Setter	

6. Package game.sharedObject

6.1 Class ResourceHolder

6.1.1 Fields

+ static Image normalBall	Normal ball image
+ static Image gameBackground	Game scene background image.
+ static Image tutorialBackground;	Tutorial scene background image.
+ static Image winningBackground	Winning scene background image.
+ static Image bombBall	Bombball image.
+ static Image bombItem	Bomb item image.
+ static Image bombEffect	Bomb explosion effect image.
+ static Image ballCracking	Normal ball cracking effect.
+ static Image bombballCracking	Bomb ball cracking effect.
+ static Image welcomeBackground	Welcome scene background image.
+ public static Image playerSceneBackground	Player scene background image.
+ public static Image iceWall	Blocking skill image.
+ public static ImageView	Freezing effect.

freezingEffect	
+ static Image blockPerkImage	Block perk button image.
+ static Image freezePerkImage	Freeze perk button image.
+ static ImageView normalBallImageView	Normal ball image view.
+ static ImageView bombBallImageView	Bomb ball image view.
+ static ImageView bombImageView	Bomb image view.
+ static ImageView celebrateEffect1	Celebration1 image view.

+ static ImageView celebrateEffect2	Celebration2 image view.
+ static ImageView firework1	Firework1 image view.
+ static ImageView firework2	Firework2l image view.
+ static Font scoreFont	Scoreboarrd font.
+ static Font timeFont	Timer font.
+ static Font gameTitleFont	Welcome scene head text’s font.
+ static Font buttonFont1	Button font. Type1
+ static Font buttonFont2	Button font. Type2
+ static Font buttonFont3	Button font. Type3

+ static Font headerFont	Head text font.
+ static Font gameFont	Game detail font.
+ static Font smallGameFont	Small game detail font.
+ static Font winningFont	Winner announcement font.
+ static Font itemSelectionFont	Item selection font.
+ static MediaPlayer themeSongs	Background theme song.
+ static MediaPlayer gamethemeSongs	Game background theme song.
+ static MediaPlayer winningSongs	Winning scene cheering song.
+ static AudioClip bounceSound	Bouncing sound effect.
+ static AudioClip clickingSound	Clicking sound effect.
+ static AudioClip scoringSound	Scoring sound effect.

+ static AudioClip + bombExplosionSound	Explosion sound effect.
+ static AudioClip alertSound	Alert sound effect.
+ static AudioClip blockSound	Blocking perk selection sound effect.

+ static AudioClip freezeSound	Freezing perk selection sound effect.

6.1.2 Constructor

static {loadResource()}	Load resources using class loader.

6.1.3 Methods

+ static void loadResource()	Initialize all variables.

7. Package game.skill

7.1 Class BlockingSkill implements IActivatable

7.1.1 Fields

+ final int cooldown	Cooldown duration of the skill.

	Initialize to 20.

7.1.2 Methods

+ void activate(NormalBall ball, String player, GraphicsContext gc)	Activate the blocking effect. When each player casts the skill, the wall is created to help player block shots.
+ deactivate(Entity ball)	Do nothing
+ deactivate(String player, GraphicsContext gc)	When the wall time is over, the wall disappears.
+ boolean isReady1(int activatedTime1)	Check if player1 casts the skill.
+ boolean isReady2(int activatedTime2)	Check if player2 casts the skill.
Generate Getter/Setter	

7.2 Class FreezingSkill implements IActivatable

7.2.1 Fields

+ final int cooldown	Cooldown duration of the skill. Initialize to 15.

7.2.2 Methods

+ void activate(NormalBall ball, String player, GraphicsContext gc)	Activate the freezing effect. When each player casts the skill, the ball moves slower than usual.
+ deactivate(Entity ball)	When the freeze time is over, set the ball to normal speed
+ deactivate(String player, GraphicsContext gc)	Do nothing
+ boolean isReady1(int activatedTime1)	Check if player1 casts the skill.
+ boolean isReady2(int activatedTime2)	Check if player2 casts the skill.
Generate Getter/Setter	

7.3 Interface IActivatable

7.3.1 Fields

+ static final int duration	Duration of the skill, initilize to 2 seconds.

7.3.2 Methods

+ void activate(NormalBall ball, String player, GraphicsContext gc)	Activate the skill.
+ void deactivate(Entity ball)	Deactivate the skill.
+ void deactivate(String player, GraphicsContext gc)	Deactivate the skill.
+ boolean isReady1(int activatedTime1)	Check if player1 casts the skill.
+ boolean isReady2(int activatedTime2)	Check if player2 casts the skill.
+ int getMaxCooldown()	Return the cooldown of the skill.

7.4 Enum Perk

NONE, FREEZING, BLOCKING

8. Package SceneController

8.1 Class SceneController

8.1.1 Fields

- static Stage stage	The stage that all the scenes will be set.

8.1.2 Methods

+ static void setTutorialScene()	Set the tutorial scene to be displayed on the stage.
+ static void setPlayerScene()	Set the player scene to be displayed on the stage.
+ static void setGameScene(PlayerScene playerScene)	Set the game scene to be displayed on the stage.
+ static void setWinningScene(String name)	Set the winning scene to be displayed on the stage.
+ static void setWelcomeScene()	Set the welcome scene to be displayed on the stage.
+ static void setStage(Stage stage)	Set the stage that will display the scenes .

9. Package Main

9.1 Class Main extends Application

9.1.1 Methods

+ void start(Stage stg)	Initialize and set the stage’s component.
+ static void main(String[] args)	Launch the application