

รายงานเกม Quoridor

จัดทำโดย

6231310021 นายชยุต ตรีนรินทร์

6232006521 นายฐกนต์ สมพงษ์

วิชา 2110215 PROGRAMMING

METHODOLOGY

ภาคเรียนที่ 2 ปีการศึกษา 2019

จุฬาลงกรณ์มหาวิทยาลัย

บทนำ

นิสิต 2 คนได้เล่นเกมกระดาน Quoridor แต่ไม่พอใจในกฎของเกม จึงคิดที่จะทำเกม Quoridor ของตัวเองขึ้นมาแทน

วิธีเอาชนะเกม:

เกมใหม่นี้มีกฎเหมือนเดิม นั่นก็คือ ไปอีกฝั่งของกระดานให้ได้ แต่ก็ได้กฎใหม่ขึ้นมา ซึ่งก็คือ ลดพลังชีวิตของศัตรูให้เหลือ 0

กฎการเล่น:

1. เริ่มเกม

ผู้เล่นแต่ละฝั่งจะต้องเลือกสีของหมากตัวเอง (ต้องเลือกคนละสี หากเป็นสีเดียวกัน จะไม่มีปุ่ม next ขึ้นให้เล่นเกม) เมื่อเริ่มเกมมา ผู้เล่นจะมีไอเท็มเริ่มต้นดังต่อไปนี้:

- ระเบิด 3 ลูก
- Barricade 10 อัน
- Barricade Removal 5 ครั้ง
- 3 Life Point

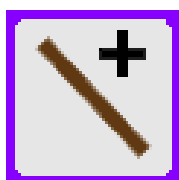
2. ในแต่ละเทิร์น ผู้เล่นจะต้องทำ 1 อย่างดังนี้:

- เดิน : เลือกเดินได้ 4 ทิศทาง : ขึ้น ลง ซ้าย ขวา โดยที่ผู้เล่นจะไม่สามารถเดินผ่าน Barricade หรือ เดินทับหมากของผู้เล่นอื่นได้
 - วาง Barricade : เลือกวาง Barricade ส่วนไหนก็ได้ตรงช่องกระดาน โดยให้เลือกวางแนวตั้งกับแนวนอน โดยที่กฎการวางนั้นมีอยู่ข้อเดียว : ฝั่งตรงข้ามและเราจะต้องเดินไปจุดเกิดและฝั่งตรงข้ามกับจุดเกิดของตนเองได้หลังจากที่วางไปแล้ว ไม่งั้นจะไม่สามารถวางได้
 - Remove Barricade : เลือกเอา Barricade ที่อยู่บนกระดานออก อันไหนก็ได้
 - วางระเบิด : ลงตรงไหนก็ได้ของกระดาน เมื่อฝั่งใดก็ได้เดินไปทับระเบิดจะกลับไปจุดเกิด และลด Lifepoint ลง 1
- เกมไม่ให้คลิกที่หมากของผู้เล่น

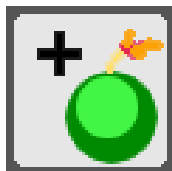
3. Special Tile:

ทุกๆ 2 เทิร์น จะสุ่มเกิด Special Tile บนกระดาน
ตรงไหนก็ได้ที่เป็นพื้นธรรมดา(ไม่มีผู้เล่น,ระเบิด หรือ Special Tile อื่นๆ) โดยมีทั้งหมด 8
แบบ ดังนี้:

1. ได้ Barricade เพิ่ม 1 อัน
2. ได้ระเบิด 1 ลูก
3. เพิ่ม Life point ผู้เล่นที่เหยียบ 1 ถ้าผู้เล่นเหยียบขณะมี Life point 3
จะไม่เกิดอะไรขึ้น
4. Move Player : สุ่มผู้เล่นที่เหยียบเดินไปทิศไหนก็ได้ 1 ทิศ
โดยถ้าสุ่มเดินแล้วมี Barricade หรือผู้เล่นอื่นอยู่ก็ไม่ได้เดิน
5. Move Other Player :
สุ่มผู้เล่นฝั่งตรงข้ามเดินไปทิศไหนก็ได้โดยกฎเหมือนกับ Move Player
แต่จะ**บังคับ**ฝั่งตรงข้ามใช้เทิร์นและกลับมาเทิร์นตัวเองทันที
6. Remove All Barricade : ทำให้ Barricade ทั้งหมดบนกระดานหายไป
7. Remove All Special Tile : ทำให้ Special Tile
ทั้งหมดบนกระดานหายไป
8. Random Special Tile สุ่มเอฟเฟกต์ของ Special Tile 7
อันข้างบนเมื่อเหยียบลงไป



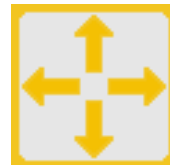
Get 1 Barricade



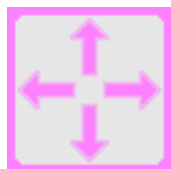
Get 1 Bomb



Heal 1



Move Player



Move other Player



Remove All player



Remove special tile

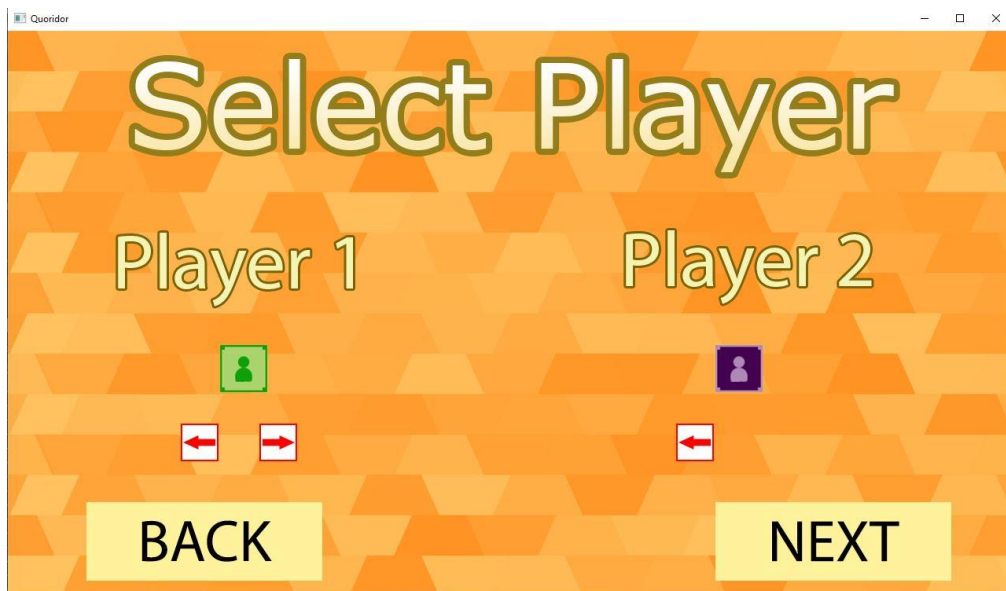


Random

Main Menu จะมีหน้าต่างแบบนี้

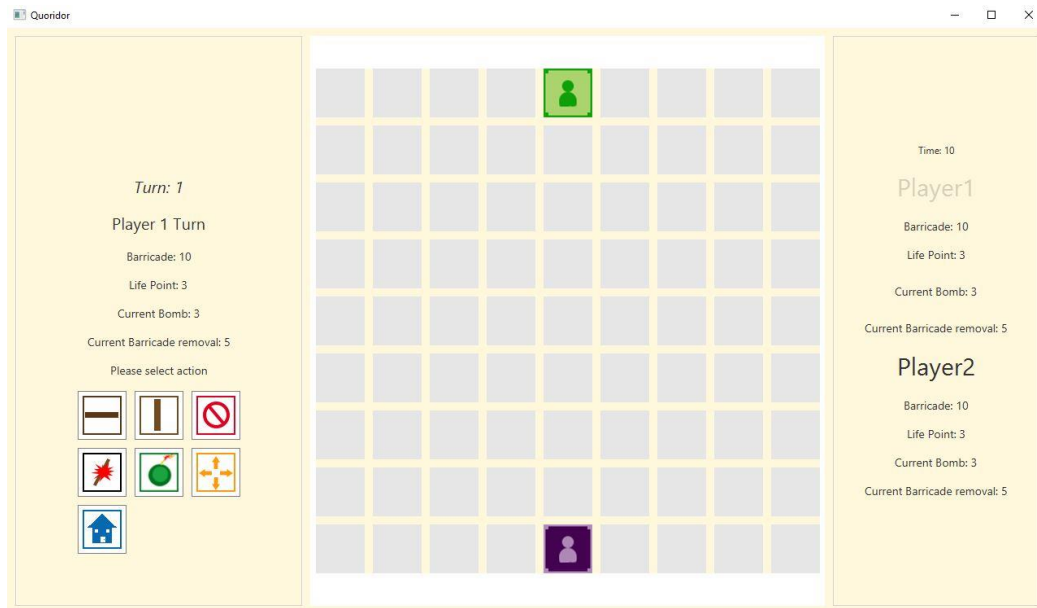


เมื่อกด Play จะไปหน้าเลือกหมาก ต้องเป็นหมากคนละสีถึงจะไปหน้าต่อไปได้



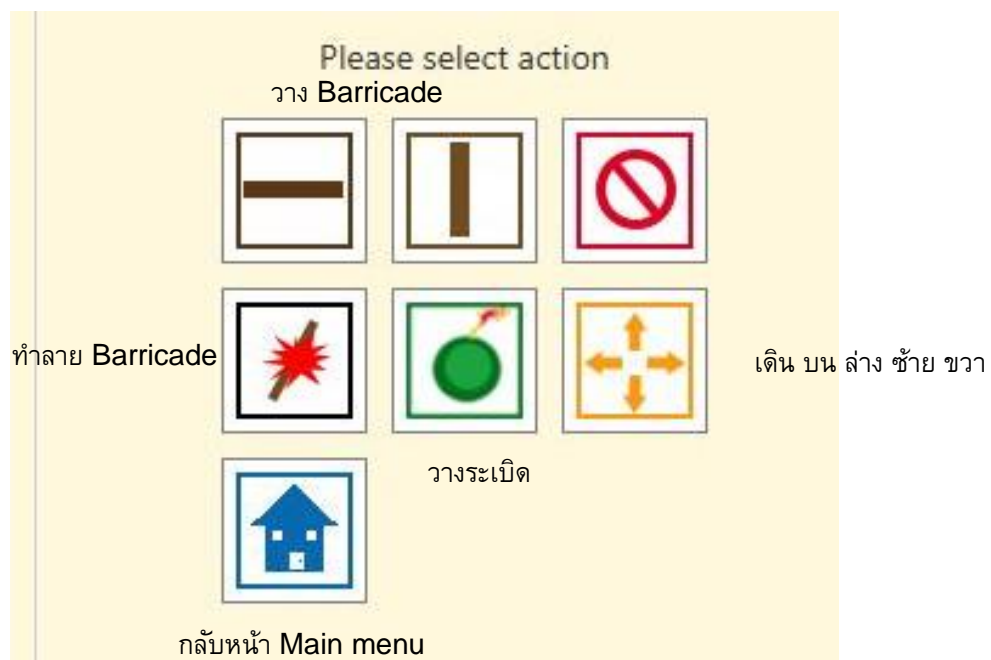
หน้าเกมจะมีหน้าตาแบบนี้

Player 1 จะอยู่ตรงนี้

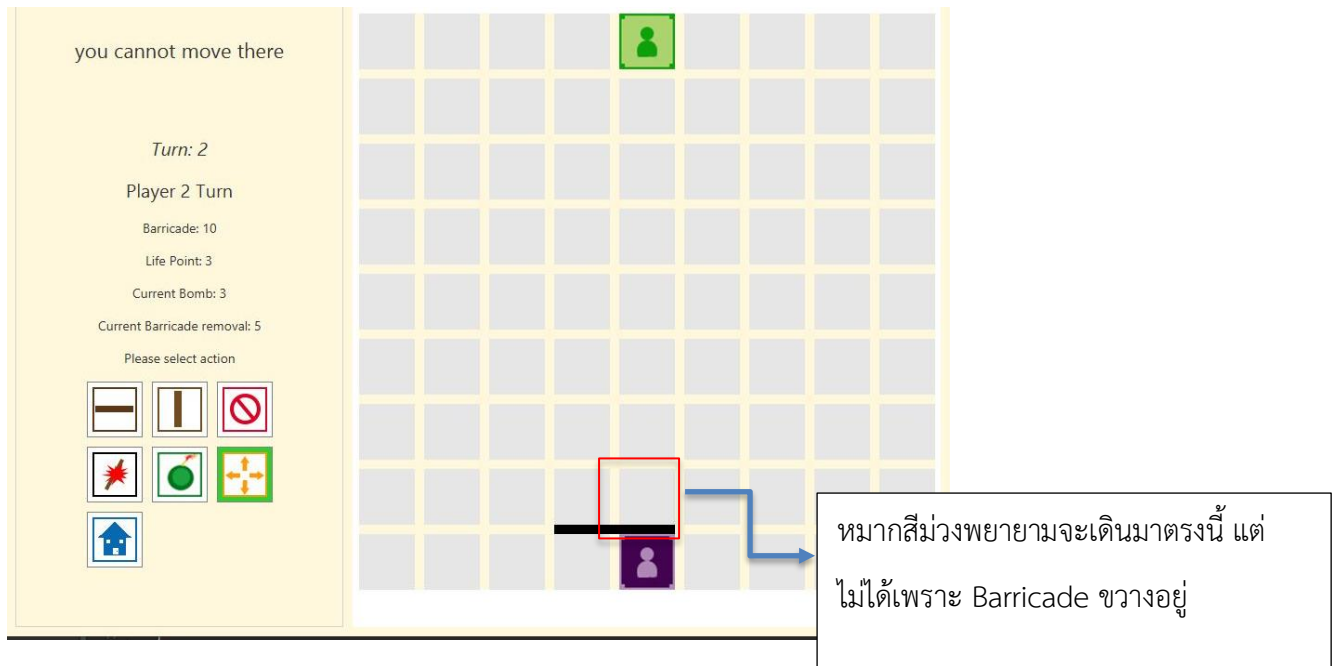


Player 2 อยู่ตรงนี้

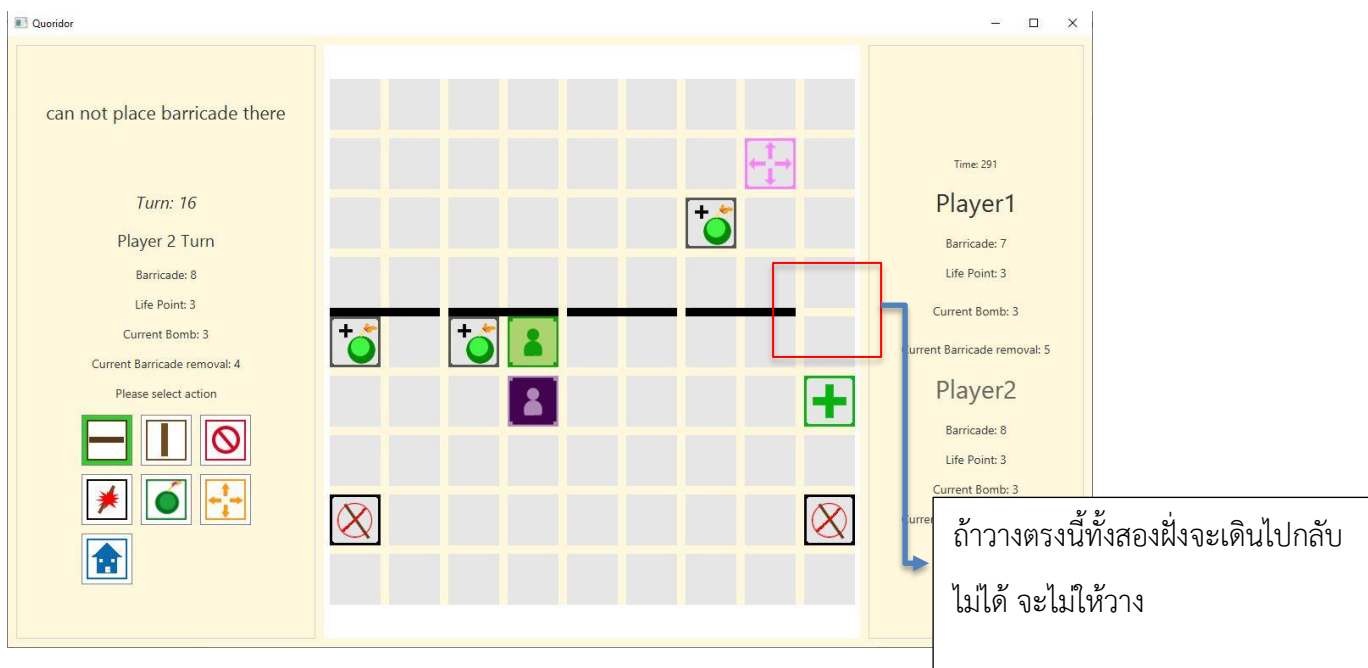
สามารถเลือกทำได้ 4 อย่าง



ไม่สามารถเดินได้ถ้ามี Barricade ขวางอยู่



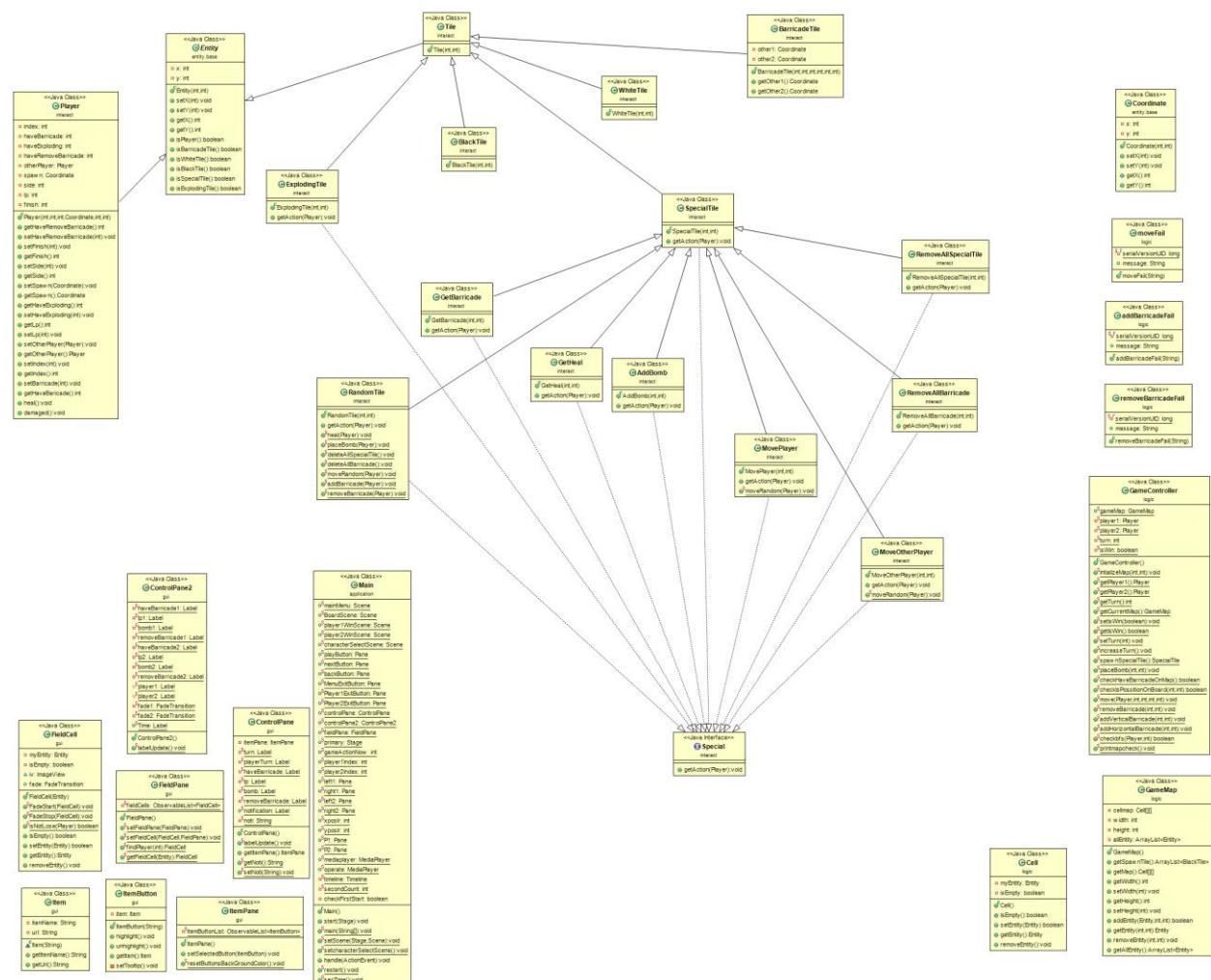
หากเลือกวาง Barricade แล้วหมากทั้งสองฝั่งไม่สามารถเดินไปจุดเกิดและฝั่งชนะได้ จะไม่ให้วาง



เมื่อ Player ชนะจะขึ้นหน้าแบบนี้



UML DIAGRAM



Package Overview

- Private
- + Public
- # Protected

Underlined Static

1. Package: entity.base

สำหรับเป็น object ใน interact เพื่อให้ใช้ได้ง่ายๆ

1.1 Class: coordinate

Fields

- int x
- int y

Method

+ Coordinate(int x, int y)	Initialize object
+ void setX()	ปรับค่า x
+ void setY()	ปรับค่า y
+ int getX()	คืนค่า x
+ int getY()	คืนค่า y

1.2 Abstract Class: Entity

Fields

- Int x
- Int y

Method

+ Entity(int x, int y)	Initialize object
+ void setX()	set ค่า x
+ void setY()	set ค่า y
+ int getX()	คืนค่า x
+ int getY()	คืนค่า y
+ boolean isPlayer()	ตรวจสอบว่าเป็น class Player หรือไม่
+ boolean isBarricadeTile()	ตรวจสอบว่าเป็น class Barricade หรือไม่

+ boolean isWhiteTile()	ตรวจสอบว่าเป็น class WhiteTile หรือไม่
+ boolean isSpecialTile()	ตรวจสอบว่าเป็น class SpecialTile หรือไม่
+ boolean isExplodingTile()	ตรวจสอบว่าเป็น class ExplodingTile หรือไม่

2. Package: interact

2.1 Interface: Special

Method

+ getAction(Player e)	ทำ Action พิเศษกับ Player
-----------------------	---------------------------

2.2 Class: Tile extends Entity

Method

+ Tile(int x,int y)	Initialize object
---------------------	-------------------

2.3 Class: ExplodingTile extends Tile implements Special

+ ExplodingTile(int x,int y)	Initialize object
+ void getAction(Player player)	-decrease 1 lp of player -respawn player at spawn position

2.4 Class: WhiteTile extends Tile

Method

+ WhiteTile(int x,int y)	Initialize object
--------------------------	-------------------

2.5 Class:BlackTile extends Tile

Method

+ BlackTile(int x,int y)	Initialize object
--------------------------	-------------------

2.6 Class: Player extends Entity

Fields

- Int index -> character type(there are 7 types, 1 to 7)
- Int haveBarricade
- Int haveExploding
- Int haveRemoveBarricade
- Player otherPlayer
- Coordinate spawn
- Int side -> player side(can only be 1 or 2)
- Int lp -> player lifepoint (maximum 3, dead when 0)
- Int finish -> finish position of x

Method

+ Player(int x,int y)	Initialize object
+int getHaveRemoveBarricade()	รับค่า haveRemoveBarricade
+ void setHaveRemoveBarricade(int haveRemoveBarricade)	Set ค่า haveRemoveBarricade
+ void setFinish()	Set finish position
+ int getFinish	Get finish position
+ void setSide	Set player side (can only be 1 or 2)

+ int getSide	Get player side
+ void setSpawn	Set spawn position for player
+ Coordinate getSpawn()	Get spawn position
+ int getHaveExploding	Get haveExploding
+ void setHaveExploding(int haveExploding)	Set haveExploding
+ int getLp()	Get lp
+ void setOtherPlayer()	Set otherPlayer
+ Player getOtherPayer()	Get otherPlayer
+ void setIndex()	Set Index
+ int getIndex()	Get Index
+ void setBarricade(int haveBarricade)	Set haveBarricade
+ int getHaveBarricade()	Get haveBarricade

+ void heal()	Increase lp by 1 if lp if already 3 don't increase
+ void damaged()	Decrease lp by 1

2.7 Class: BarricadeTile extends Tile

Fields

- Coordinate other1
- Coordinate other2

Method

+ BarricadeTile(int x,int y,int X1,int Y1,int X2,int Y2)	Initialize object
+ Coordinate getOther1()	Get other1
+ Coordinate getOther2()	Get other2

2.8 Class: SpecialTile extends Tile implements Special

Method

+ SpecialTile(int x,int y)	Initialize object
+ void getAction(Player e)	This function will not be used

2.9 Class: AddBomb extends SpecialTile implement Special

+ AddBomb(int x,int y)	Initialize object
+ void getAction(Player e)	- increase 1 haveExploding of player e -Set both ControlPane and ControlPane2 labelUpdate

2.10 Class: GetBarricade extends SpecialTile implement Special

Method

+ GetBarricade(int x,int y)	Initialize object
-----------------------------	-------------------

+ void getAction(Player e)	- increase 1 haveBarricade of player e -Set both ControlPane and ControlPane2 labelUpdate
----------------------------	--

2.11 Class: GetHeal extends SpecialTile implement Special

Method

+ GetHeal(int x,int y)	Initialize object
+ void getAction(Player e)	- increase 1 lp of player e -Set both ControlPane and ControlPane2 labelUpdate

2.12 Class: MoveOtherPlayer extends SpecialTile implement Special

Method

+ MoveOtherPlayer(int x,int y)	Initialize object
--------------------------------	-------------------

+ void getAction(Player e)	- move other player randomly by using function moveRandom(Player e) -Set both ControlPane and ControlPane2 labelUpdate
+ void moveRandom(Player e)	Move player e in random position

2.13 Class: MovePlayer extends SpecialTile implement Special

Method

+ MoveOtherPlayer(int x,int y)	Initialize object
+ void getAction(Player e)	- move player randomly by using function moveRandom(Player e) -Set both ControlPane and ControlPane2 labelUpdate
+ void moveRandom(Player e)	Move player e in random position

2.14 Class: RemoveAllBarricade extends SpecialTile implement Special

Method

+ RemoveallBarricade(int x,int y)	Initialize object
+ void getAction(Player e)	- remove all barricade on the board -Set both ControlPane and ControlPane2 labelUpdate

2.15 Class: RemoveAllSpecialTile extends SpecialTile implement Special

Method

+ RemoveallSpecial(int x,int y)	Initialize object
+ void getAction(Player e)	- remove all SpecialTile on the board -Set both ControlPane and ControlPane2 labelUpdate

2.16 Class: RandomTile extends SpecialTile implement Special

Method

+ RandomTile (int x,int y)	Initialize object
+ void getAction(Player e)	<p>- randomly choose action(there are 7 actions move player, move other player, increase player barricade , increase remove player barricade, delete all specialTile on the board, delete all barricades on the board, heal player, increase player haveExploding.</p> <p>-Set both ControlPane and ControlPane2 labelUpdate</p>

3. Package: logic

3.1 Class: addBarricadeFail extends exception

Fields

- final long serialVersionUID = 1L
- + String message

Method

public addBarricadeFail(String message)	Set string message
---	--------------------

3.2 Class: moveFail extends exception

Fields

- final long serialVersionUID = 1L
- + String message

Method

public moveFail(String message)	Set string message
---------------------------------	--------------------

3.3 Class: removeBarricadeFail extends exception

Fields

- final long serialVersionUID = 1L

+ String message

Method

public RemoveBarricadeFail(String message)	Set string message
--	--------------------

3.4 Class: Cell

Fields

- Entity myEntity
- boolean isEmpty

Method

+ Cell()	Initialize object
+ Boolean isEmpty()	Get isEmpty
+ setEntity(Entity e)	Set myEntity
+ Entity getEntity()	Get myEntity

+ void removeEntity()	-Set myEntity to null -set isEmpty to true
-----------------------	---

3.5 Class: GameMap

Fields

- Cell[][] cellmap
- Int width
- Int height
- ArrayList<Entity> allEntity

Method

+ GameMap()	Initialize object -set height and row -create all cell in cellmap
+ ArrayList<BlackTile> getSpawnTile()	Get all blackTile that can be changed to specialtile
+ Cell[][] getMap()	Get map
+ int getWidth()	Get width

+ void setWidth(int width)	Set width
+ void getHeight()	Get height
+ void setHeight(int height)	Set height
+ boolean addEntity(Entity e,int x,int y)	Add entity to the map
+ Entity getEntity(int x,int y)	Get entity at position (x,y) on the map
+ removeEntity(int x,int y)	Remove entity at position (x,y) on the map
+ ArrayList<Entity> getAllEntity()	Get all entity on the map

3.6 Class: GameController

Fields

- + GameMap gameMap
- Player player1
- Player player2
- Int turn
- Boolean isWin

Method

+ InitializeMap(int player1index,int player2index)	Initialize object
+ Player getPlayer1()	Get player 1
+ Player getPlayer2()	Get player 2
+ int getTurn()	Get current turn
+ GameMap getCurrentMap()	Get current map
+ void setIsWin(Boolean stat)	Set isWin
+ boolean getIsWin()	Get iswin
+ void setTurn(int t)	Set turn
+ void increaseTurn()	Increase turn
+ SpecialTile spawnSpecialTile()	Spawn a specialtile randomly on the map(this function may or may not spawn the special. It depend on density of none blackTile

	on the map. Also a Specialtile won't spawn near other Specialtiles)
+ void placeBomb(int x,int y)	Place bomb on the map
+ boolean checkHaveBarricadeOnMap()	Return true if there is atleast one barricade tile on the map otherwise return false
+ checkIsPositionOnBoard(int x,int y)	Check if position (x,y) on the board (17*17)
+ void move(Player player,int posx,int posy,int x,int y)	Move player
+ void removeBarricade(int x,int y)	Remove barricade on the map
+ void addVerticalBarricade(int x,int y)	Place barricade vertically
+ void addHorizontalBarricade(int x,int y)	Place barricade horizontally

+ boolean checkbfs(Player pplayer,int finish)	Return true if player can get to the finish position and can get back to the spawn position(using breadth-first search algorithm), otherwise return false
+ void printmapcheck()	Print map on console for checking

4. Package: gui

4.1 Class: ControlPane extends VBox

Fields

- ItemPane itemPane;
- Label turn
- Label playerTurn
- Label haveBarricade
- Label lp
- Label bomb
- Label removeBarricade
- + Label notification
- String noti

Method

+ ControlPane()	Initialize object
+ void labelupdate()	Update all labeles
+ ItemPane getItemPane()	Get itemPane
+ String getNoti()	Get noti
+ setNoti(String noti)	Set noti

4.2 Class: ControlPane2 extends VBox

Fields

- Label haveBarricade1
- Label lp1
- Label bomb1
- Label removeBarricade1
- Label haveBarricade2
- Label lp2
- Label bomb2
- Label removeBarricade2

- # Label player1

- # Label player2

- FadeTransition fade1
- FadeTransition fade2

+ Label Time

Method

+ ControlPane2()	Initialize object
+ labelUpdate()	Update all labeles

4.3 Class: FieldCell extends Pane

Fields

- Entity myEntity
- Boolean isEmpty
- + public FadeTransition fade

Method

+ FieldCell(Entity entity)	<p>Initialize object</p> <p>Set image to this Pane</p> <p>addHandler(Mouse Click)</p> <p>- gameActionnow = 1,2,3,4,5 for each number means type of action at the current time</p> <p>addHandler(Mouse enter)</p> <p>- fadeTransition</p> <p>addHandler(Mouse exit)</p> <p>-fadeTransition</p>
+ void FadeStart(FieldCell now)	For set fieldcell
+ void FadeStop(FieldCell now)	For set fieldcell
+ boolean isNotLose(Player p)	Check if not loose
+ boolean isEmpty()	Check is this field empty
+ Boolean setEntity(Entity e)	Set Entity to this field
+ Entity getEntity()	Get myEntity

+ void removeEntity()	Set myEntity to null And set isEmpty to true
-----------------------	---

4.4 Class: FieldPane

Fields

- ObservableList<FieldCell> fieldCells

+ <u>FieldPane()</u>	<u>Initialize object</u>
+void <u>setFieldPane(FieldPane e)</u>	<u>Set fieldPane</u>
+ void <u>set FieldCell(FieldCell e,FieldPane E)</u>	<u>Set fieldCell</u>
+ <u>FieldCell findPlayer(int side)</u>	<u>Return field cell at player is in</u>
+ <u>FieldCell getFieldCell(Entity E)</u>	<u>Return get field cell entity E currently live in</u>

4.5 Class: Item

Fields

- String itemName
- String url

Method

Item(String itemName)	Initialize Object
String getItemName()	Get itemName
String getUrl()	Get Url

4.6 Class: ItemButton

Fields

- Item item

+ ItemButton(String itemName)	Initialize object
+ void highlight()	Highlight button to LimeGreen color
+ void unhighlight()	Unhighlight button
+ Item getItem()	Get Item

+ setTooltip()	Set tool tip
----------------	--------------

4.7 Class: ItemPane

Fields

- ObservableList<ItemButton> itemButtonList

Method

+ ItemPane()	Get ItemPane
+ void setSelectedButton(ItemButton selecteditemButton)	- resetButtonBackgroundColor () -Set selected button
+ void resetButtonBackgroundColor()	Reset background color

5. Package application

5.1 Class Main extends Application

Fields

- + Scene mainMenu
- + Scene BoardScene
- + Scene player1WinScene
- + Scene player2WinScene
- + Scene characterSelectScene
- + Pane playButton
- + Pane nextButton
- + Pane backButton
- + Pane MenuExitButton
- + Pane Player1ExitButton
- + Pane Player2ExitButton
- + ControlPane controlPane
- + ControlPane2 controlPane2
- + Fieldpane fieldpane
- + Stage primary
- + int gameActionNow = 0
- + int player1Index = 4
- + int player2Index = 4
- + Pane left1
- + Pane left2
- + Pane right1
- + Pane right2
- + int xposlr = 120
- + int yposlr = 500
- + Pane P1
- + Pane P2

- + Mediaplayer mediaplayer
- + Mediaplayer operate
- Timeline timeline
- int secondCount
- boolean checkFirstStart = False

Method

+ void start(Stage primaryStage)	<ul style="list-style-type: none"> - set mainMenuScene - set characterSelectScene - set player1WinScene - set player2WinScene - set BoardScene - set Background music
+ void main(String[] args)	- main method
+ void setScene(Stage primaryStage, Scene scene)	<ul style="list-style-type: none"> - set Title to “Quoridor” - Set scene to primaryStage - show primaryStage
+ void setcharacterSelectScene()	<ul style="list-style-type: none"> - disable left2/right2 pane if player1Index/player2Index = 1/7 - load the picture of player token from selected index - set Player1/Player2 Index using gamecontroller
+ void handle(ActionEvent event)	- set the scene into characterSelectScene if user clicked playButton
+ void restart()	- restart the game board

+ void secTime()

- set time frame
- update time frame