Final Project report

CP Engineerly

Mr. Jetnipat Suwanarat 6130085021
Miss Chnan Likitvanichkul 6130096021

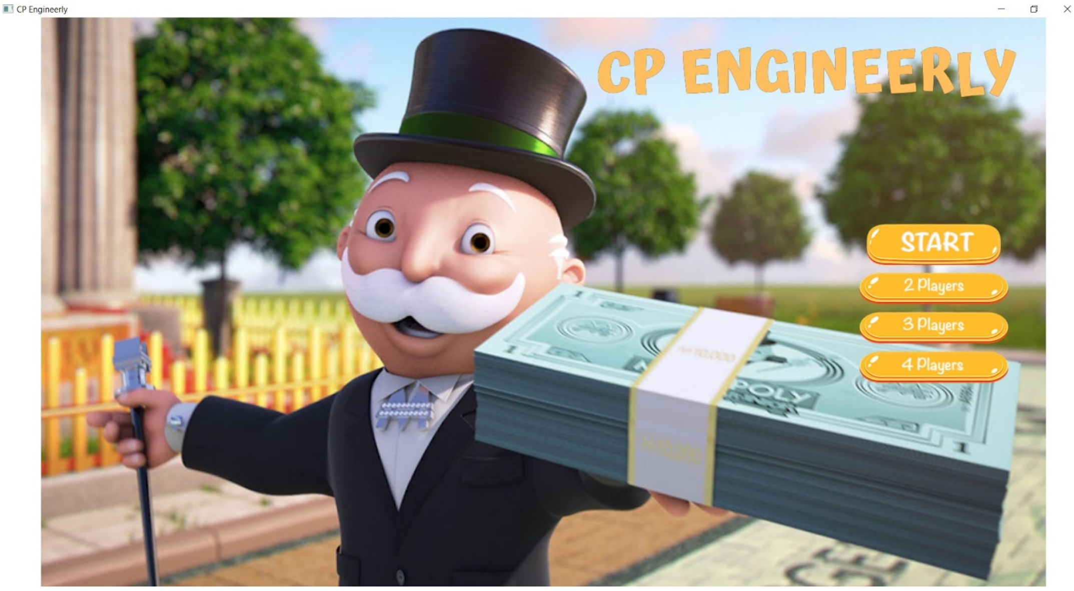
Department of computer enginerring

Game Introduction

Once upon a time there was a boy who is learning in Computer Programming Engineering Department in Chulalongkorn University. Unfortunately, the boy and some his friend was transform to a Pawn and go to virtual world. In the virtual world, it has only one rule for comeback to real world that is win the game. That means it has only one person can comeback to real world. the environment of virtual world is similar as Faculty of Engineering environment, Chulalongkorn University. Now, we will see who is going to comeback to real world!

User Manual

Rule : The winner is person who last survivor the game. If any person has money below zero, he is going to bankrupt.



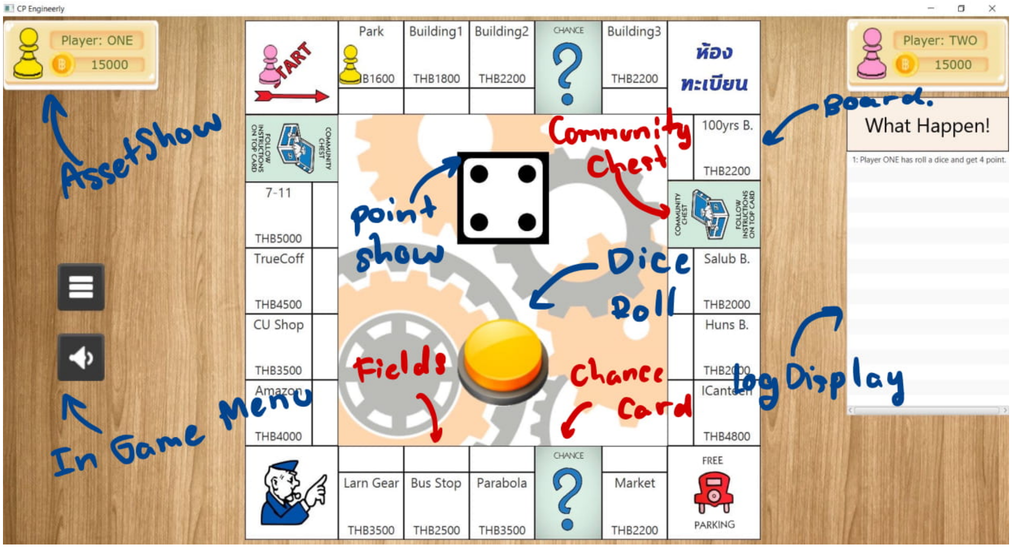
How to play

1.Choose player for playing game

1.1 Minimum player: 2 players

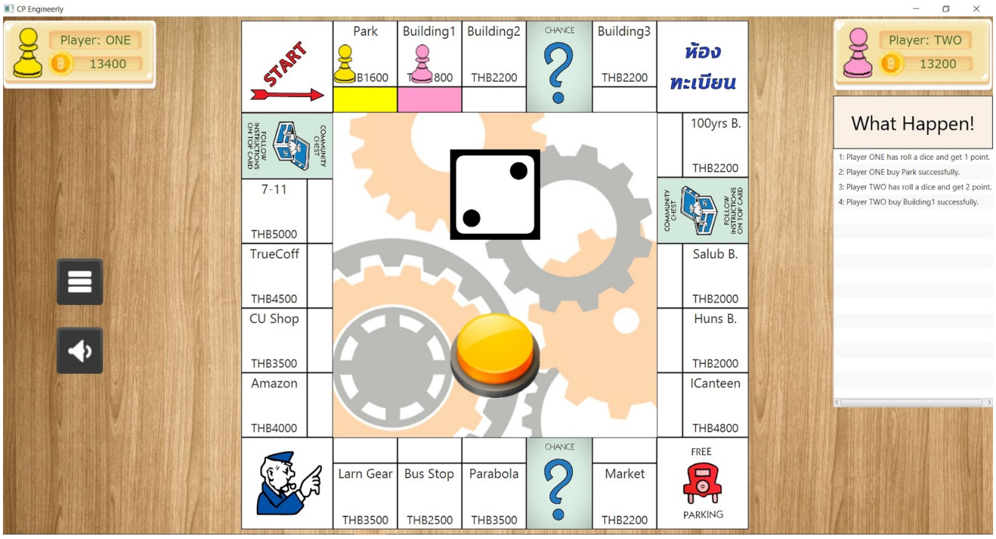
1.2 Maximum player: 4 players

2.In game scene



1: Player ONE has roll a dice and get 4 point.

3.Next step, Roll a dice!



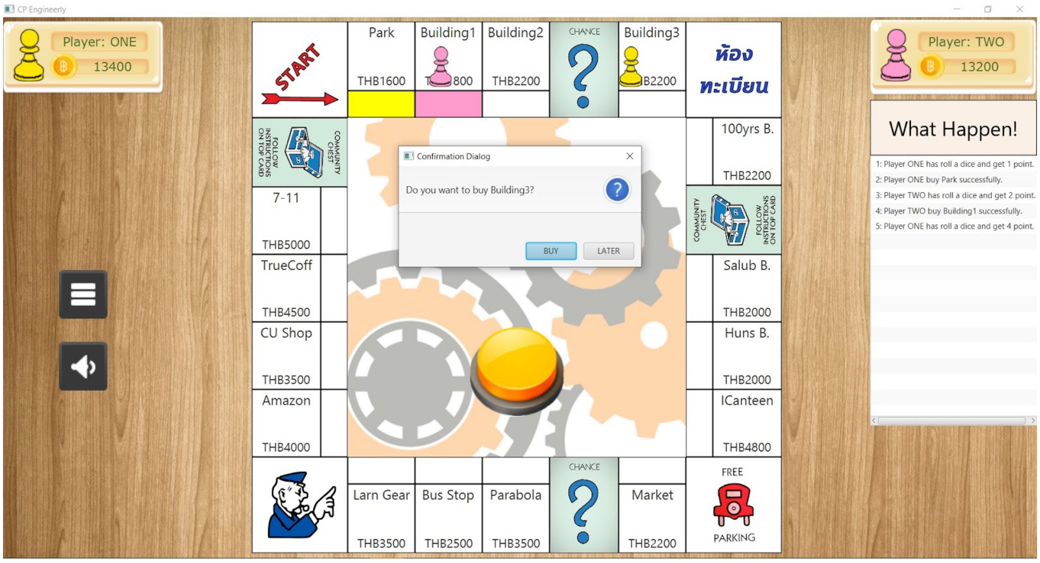
What Happen!

1: Player ONE has roll a dice and get 1 point.
2: Player ONE buy Park successfully.
3: Player TWO has roll a dice and get 2 point.
4: Player TWO buy Building1 successfully.

4.The player will walk to N field due to point that you get from dice.

5.while you get in Land Field, it will happen below.

5.1 If the field not has Owner, you can buy the field. The game will alert you some question like this.



Confirmation Dialog

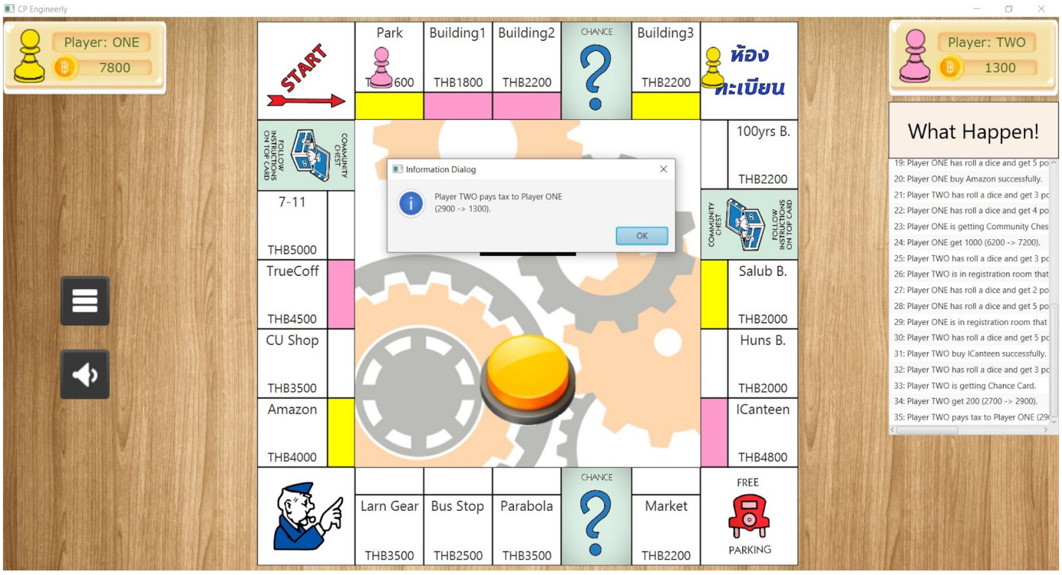
Do you want to buy Building3?

What Happen!

1: Player ONE has roll a dice and get 1 point.
2: Player ONE buy Park successfully.
3: Player TWO has roll a dice and get 2 point.
4: Player TWO buy Building1 successfully.
5: Player ONE has roll a dice and get 4 point.

5.2 If the field has Owner, you can not buy the field. The game will force you to pay tax to Owner of field and show an information of the process immediately.

5.2.1 If you have money greater than or equal price of Land Field, decrease your money equal Land Field price.



Information Dialog

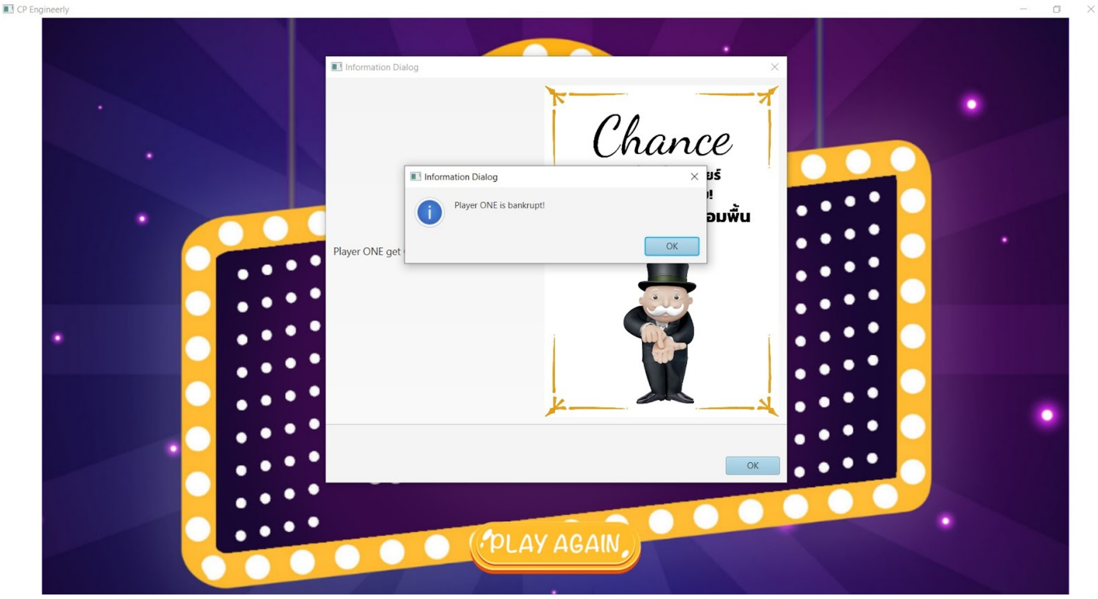
Player TWO pays tax to Player ONE
(2900 -> 1300).

OK

What Happen!

```
19: Player ONE has roll a dice and get 5 po
20: Player ONE buy Amazon successfully.
21: Player TWO has roll a dice and get 3 pc
22: Player ONE has roll a dice and get 4 po
23: Player ONE is getting Community Ches
24: Player ONE get 1000 (6200 -> 7200).
25: Player TWO has roll a dice and get 3 pc
26: Player TWO is in registration room that
27: Player ONE has roll a dice and get 2 po
28: Player ONE has roll a dice and get 5 po
29: Player ONE is in registration room that
30: Player TWO has roll a dice and get 5 pc
31: Player TWO buy ICanteen successfully.
32: Player TWO has roll a dice and get 3 pc
33: Player TWO is getting Chance Card.
34: Player TWO get 200 (2700 -> 2900).
35: Player TWO pays tax to Player ONE (29
```

5.2.2 If you have money less than, you will bankrupt;



Information Dialog

Player ONE get

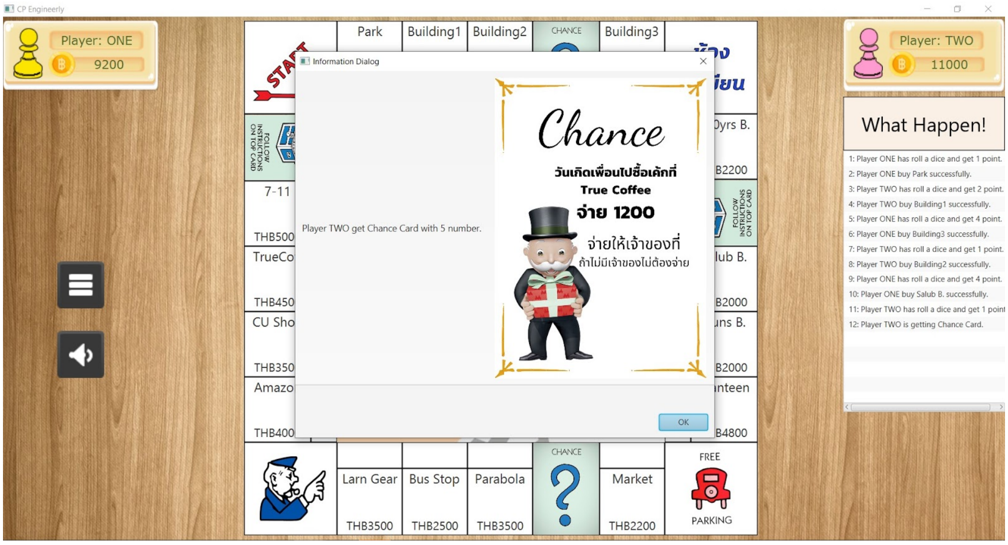
OK

Information Dialog

Player ONE is bankrupt!

OK

6.While you get in Chance Card Field or Community Chest Field, the deck of Chance Card and Community Chest has 10 cards and each of card has different effect.

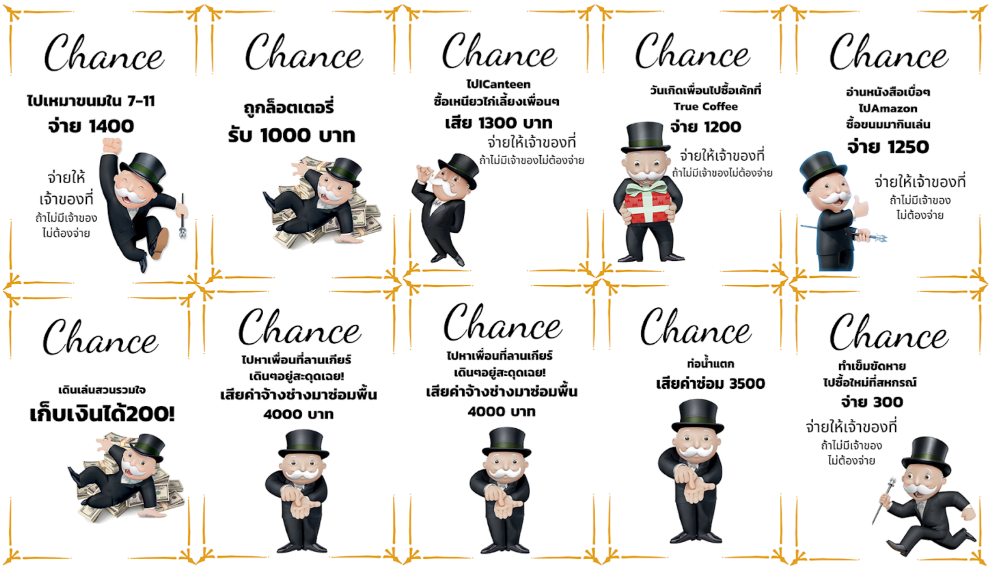


Player TWO get Chance Card with 5 number.

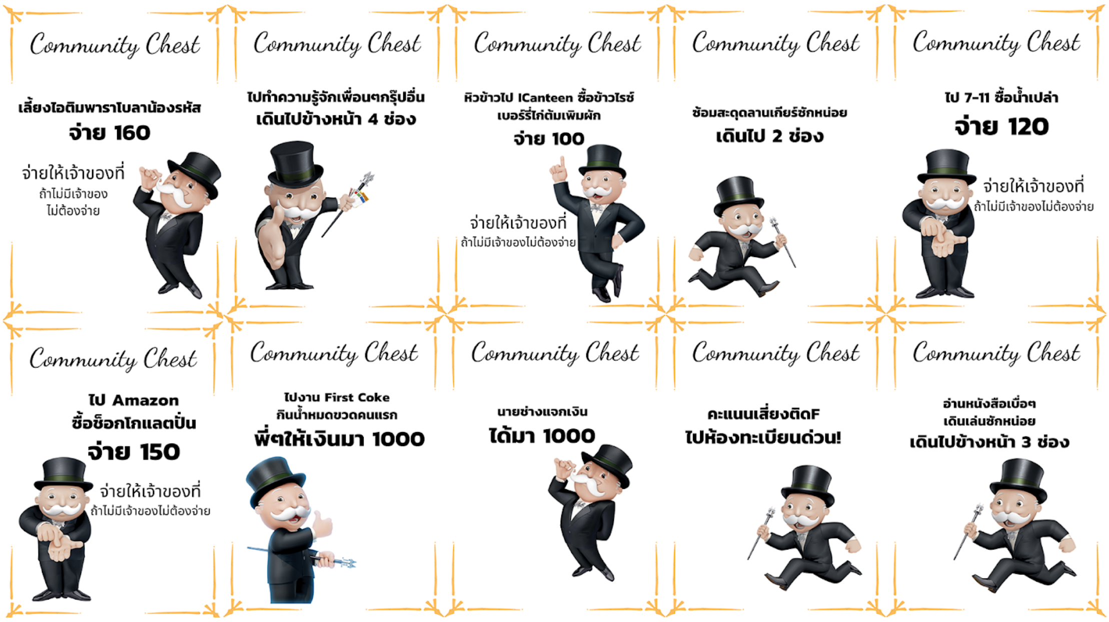
What Happen!

1: Player ONE has roll a dice and get 1 point.
2: Player ONE buy Park successfully.
3: Player TWO has roll a dice and get 2 point.
4: Player TWO buy Building1 successfully.
5: Player ONE has roll a dice and get 4 point.
6: Player ONE buy Building3 successfully.
7: Player TWO has roll a dice and get 1 point.
8: Player TWO buy Building2 successfully.
9: Player ONE has roll a dice and get 4 point.
10: Player ONE buy Salub B. successfully.
11: Player TWO has roll a dice and get 1 point
12: Player TWO is getting Chance Card.

6.1 Chance Card Deck



6.2 Community Chest Deck


Community Chest

เลี้ยงไอติมพาราโบลาน้องรหัส
จ่าย 160

จ่ายให้เจ้าของที่
ถ้าไม่มีเจ้าของ
ไม่ต้องจ่าย

Community Chest

ไปทำความรู้จักเพื่อนๆกรุ๊ปอื่น
เดินไปข้างหน้า 4 ช่อง

Community Chest

หิวข้าวไป ICanteen ซื้อข้าวไรซ์
เบอร์รี่ไก่ต้มเพิมผัก
จ่าย 100

จ่ายให้เจ้าของที่
ถ้าไม่มีเจ้าของไม่ต้องจ่าย

Community Chest

ซ้อมสะดุดลานเกียร์ซักหน่อย
เดินไป 2 ช่อง

Community Chest

ไป 7-11 ซื้อน้ำเปล่า
จ่าย 120

จ่ายให้เจ้าของที่
ถ้าไม่มีเจ้าของไม่ต้องจ่าย

Community Chest

ไป Amazon
ซื้อช็อกโกแลตปั่น
จ่าย 150

จ่ายให้เจ้าของที่
ถ้าไม่มีเจ้าของไม่ต้องจ่าย

Community Chest

ไปงาน First Coke
กินน้ำหมดขวดคนแรก
พี่ๆให้เงินมา 1000

Community Chest

นายช่างแจกเงิน
ได้มา 1000

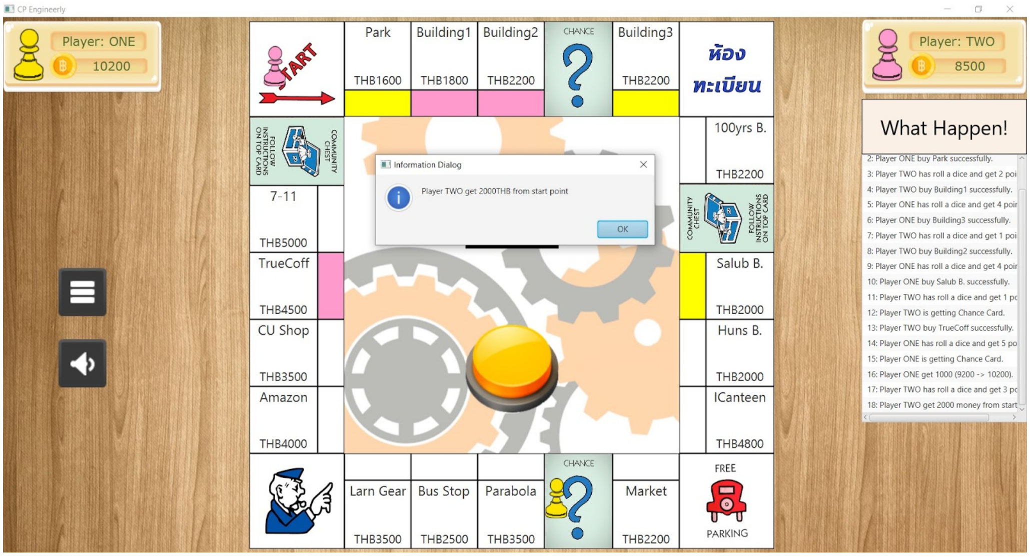
Community Chest

คะแนนเสี่ยงติดF
ไปห้องทะเบียนด่วน!

Community Chest

อ่านหนังสือเบื่อๆ
เดินเล่นซักหน่อย
เดินไปข้างหน้า 3 ช่อง

7. while you get in Corner Field, it has 4 different of Corner Field.

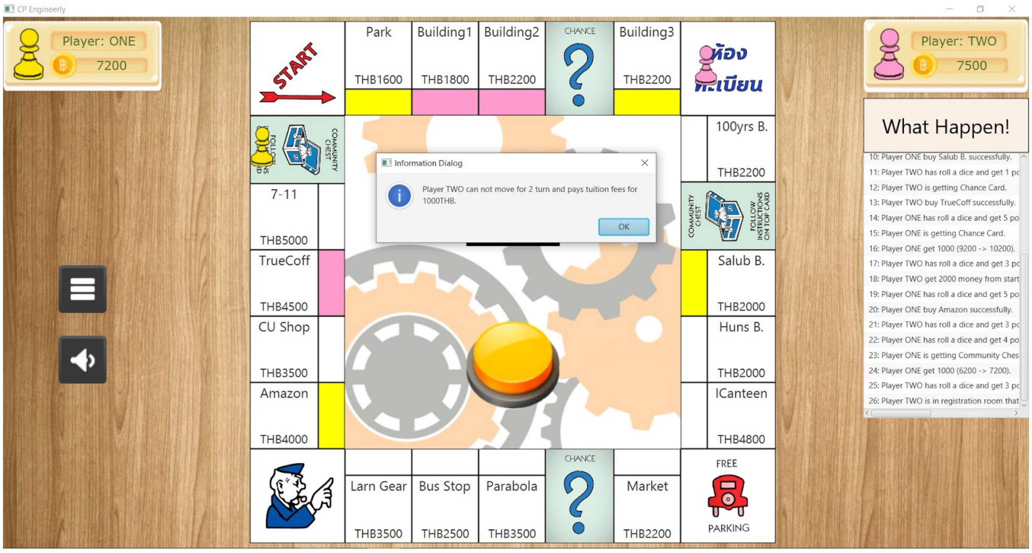


Information Dialog

Player TWO get 2000THB from start point

What Happen!

```
2: Player ONE buy Park successfully.
3: Player TWO has roll a dice and get 2 poi
4: Player TWO buy Building1 successfully.
5: Player ONE has roll a dice and get 4 poir
6: Player ONE buy Building3 successfully.
7: Player TWO has roll a dice and get 1 poi
8: Player TWO buy Building2 successfully.
9: Player ONE has roll a dice and get 4 poir
10: Player ONE buy Salub B. successfully.
11: Player TWO has roll a dice and get 1 pc
12: Player TWO is getting Chance Card.
13: Player TWO buy TrueCoff successfully.
14: Player ONE has roll a dice and get 5 po
15: Player ONE is getting Chance Card.
16: Player ONE get 1000 (9200 -> 10200).
17: Player TWO has roll a dice and get 3 pc
18: Player TWO get 2000 money from start
```



Player TWO can not move for 2 turn and pays tuition fees for 1000THB.

```
10: Player ONE buy Salub B. successfully.
11: Player TWO has roll a dice and get 1 pc
12: Player TWO is getting Chance Card.
13: Player TWO buy TrueCoff successfully.
14: Player ONE has roll a dice and get 5 po
15: Player ONE is getting Chance Card.
16: Player ONE get 1000 (9200 -> 10200).
17: Player TWO has roll a dice and get 3 pc
18: Player TWO get 2000 money from start
19: Player ONE has roll a dice and get 5 po
20: Player ONE buy Amazon successfully.
21: Player TWO has roll a dice and get 3 pc
22: Player ONE has roll a dice and get 4 po
23: Player ONE is getting Community Ches
24: Player ONE get 1000 (6200 -> 7200).
25: Player TWO has roll a dice and get 3 pc
26: Player TWO is in registration room that
```

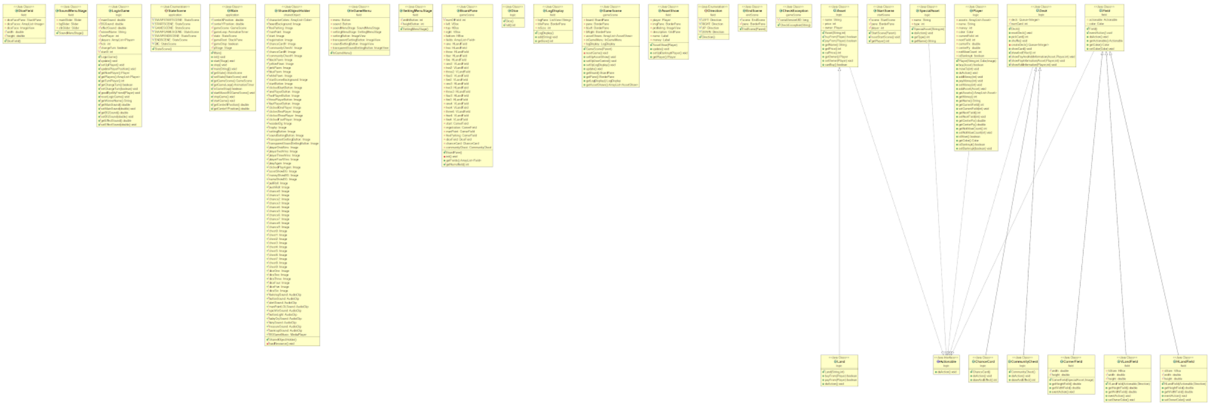
7.1 Start Point: get 2000 THB every time that you pass or get in field.

7.2 Registration Room: pay 1000 THB every time that you get in field and can not move for 2 turn.

7.3 Free Parking: Not any happen.

7.4 Man Point: forcing you go to registration room immediately.

Implementation Detail



* Noted that Access Modifier Notations can be listed below

+ (public)
# (protected)
- (private)

<u>static</u> will be underlined.

abstract will be italic.

final variable will be bold.

1. Package Application.

1. Package Application

1.1 Class Main **extends** Application

1.1.1 Fields

+ double *centerXPosition*;	Center of X axis position of primaryStage.
+ double centerYPosition	Center of Y axis position of primaryStage.
- GameScene gameScene	An instance of GameScene.
- AnimationTimer gameLoop	An instance of GameLoop.
- StateScene state	An instance of StateScene.
- StackPane gameRoot	An instance of StackPane.
- boolean gameStop	A boolean that indicate gameLoop is stop or not.
- Stage pStage	An instance of Stage that is primary stage.

1.1.2 Methods

+ void init()	Override init method of application class.
+ void start(Stage primaryStage)	Override start method of application class.
+ void stop()	Override stop method of application class.
+ void main(String[] args)	Main method of application.
+ StateScene getState()	Return StateScene of application.
+ void setState(StateScene state)	Set StateScene of application.
+ GameScene getGameScene()	Return an instance of gameScene.
+ AnimationTimer getGameLoop()	Return an instance of gameLoop.
+ boolean isGameStop()	Return a boolean of gameStop.
+ void startMusicBGGameScene()	Start the MusicBG in GameScene.
+ void stopGame()	Called when want to stop the game loop.
+ void startGame()	Called when want to start the game loop.
+ double getCenterXPosition()	Return centerXPosition.

+ double getCenterYPosition()	Return centerYPosition.

1.2. Enumeration StateScene

1.2.1 Values

SWAPSTARTSCENE, ***STARTSCENE***, ***GAMESCENE***, ***SWAPGAMESCENE***, ***SWAPENDSCENE***, ***ENDSCENE***, ***DIE***	Values for each state in game loop.

2. Package logic

2.1 Interface Actionable

2.1.1 Method

+ void doAction()	Do some action of each class that implements this interface.

2.2 Abstract Class Asset implements Actionable

2.2.1 Fields

# String name	Sring indicate the name of asset.
# int price	Integer indicate the price of asset.
# Player owner	Player indicate the owner of asset, if not has owner that is null.

2.2.2 Methods

+ Asset(String name, int price)	Constructor of Asset Class.
+ *boolean buyFrom(Player p)*	Method buy this asset. If can buy, owner is Player p then return true. But, not can buy owner is null then return false.
+ *boolean payFrom(Player p)*	Method pay money from Player p.if pay successfully return true. Otherwise, return false.
+ String getName()	Return name of asset.
+ int getPrice()	Return price of asset.
+ int calPrice()	Return calculated price of asset.
+ Player getOwner()	Return owner of asset.
+ void setOwner(Player owner)	Set Owner of asset.
+ boolean canBuy()	if asset not has owner, Return true. Otherwise, false.

2.3 Class ChanceCard extends Deck

2.3.1 Constructs

+ ChanceCard()	Constructor of ChanceCard class.

2.3.2 Methods

+ void doAction()	Override method of Actionable.
+ int drawAndEffect()	Pick a random card and do action from that card.

2.4 Class CommunityChest extends Deck

2.4.1 Constructs

+ CommunityChest()	Constructor of CommunityChest.

2.4.2 Methods

+ void doAction()	Override method of Actionable.
+ int drawAndEffect()	Pick a random card and do action from that card.

2.5 Abstract Class Deck implements Actionable

2.5.1 Fields

- Queue<Integer> deck	An instance of Queue that store Integer.
- int numCard = 10	Number of card in deck.

2.5.2 Constructors

+ Deck()	Constructor of Deck class.

2.5.3 Methods

+ void resetDeck()	Called when want to reset deck.
+ int pickCard()	Called when want to pick a card.
+ void shuffle()	Called when want to shuffle the deck.
- Queue<Integer> createDeck()	Method to create a new deck.
+ void showCard()	Method to show card in deck.
+ int drawAndEffect()	Pick a random card and do action from that card.
+ void showPayAndAddInformation(Asset asset, Player nowPlayer, int price)	Method to show information for action from card if player can pay the money.
+ void showPayInformation(Asset asset,	Method to show payment information if

Player nowPlayer, int price)	player can pay the money. Otherwise, if player will not bankrupt again, show that player is bankrupt.
+ void showAddInformation(Player nowPlayer, int price)	Show information when player get money from card.

2.6 class Dice

2.6.1 Constructor

+ Dice()	Constructor of Dice class.

2.6.2 Method

+ int roll()	Called when dice roll.

2.7 Class Land extends Asset

2.7.1 Constructor

+ Land(String name, int price)	Constructor of Land class.

2.7.2 Methods

+ boolean buyFrom(Player p)	Method buy this asset. If can buy, owner is Player p then return true. But, not can buy owner is null then return false.
+ boolean payFrom(Player p)	Method pay money from Player p. If p has money greater or equal than price of asset, Player p decrease money then return true.But not, Player p go bankrupt then return false..
+ void doAction()	Nothing do.

2.8 Class LogicGame

2.8.1 Fields

- double *mainSound* = 1.0	An instance of double used for store level of main sound.
- double *BGSound* = 1.0	An instance of double used for store level of BGsound.
- double *effectSound* = 1.0	An instance of double used for store level of effect sound.
- String *winnerName*	An instance of String that indicate the winner name.
- int *turnPlayer* = 0	An instance of Integer used for count turn.
- ArrayList<Player> *players*;	An instance of ArrayList that store player

	objects.
- int *tick* = 0	An instance of Integer used for count.
- boolean *changeTurn*;	An instance of boolean used for check that turn can change.
- int *startX* = 15;	Used for calculated the position of player.

2.8.2 Constructor

+ LogicGame()	Constructor of LogicGame.

2.8.3 Methods

+ void update()	Called when update the game every loop.
+ void setUpPlayer()	Called when want to set up player in game.
+ void updatePlayerPosition()	When resize the stage, this method will correct player position.
+ Player getNowPlayer()	Return player in now turn.
+ ArrayList<Player> getPlayers()	Return players that store Player.
+ int getTurnPlayer()	Return turnPlayer.
+ boolean getChangeTurn()	Return changeTurn.
+ void setChangeTurn(boolean changeTurn)	Set changeTurn.
+ void goodByeMyFriend(Player player)	Called when any player go to bankrupt.
+ void resetLogicGame()	Called when reset logic game.
+ String getWinnerName()	Return winnerName.
+ double getMainSound()	Return mainSound.
+ void setMainSound(double mainSound)	Set mainSound.
+ double getBGSound()	Return BGSound.
+ void setBGSound(double bGSound)	Set BGSound.
+ double getEffectSound()	Return effectSound.
+ void setEffectSound(double effectSound)	Set effectSound.

2.9 Class Player extends ImageView implements Actionable

2.9.1 Fields

- ArrayList<Asset> assets	An instance of ArrayList that store Asset.
- String name	String indicate the name of player.
- int money	Integer indicate money of player .
- Color color	An instance of Color.
- int currentField	Integer indicate current field.
- int nextField	Integer indicate next field.
- double centerPx	Integer indicate center X-Axis of player object .
- double centerPy	Integer indicate center Y-Axis of player object .
- int notMoveCount	Integer indicate a number of not move turn.
- boolean isBankrupt	Boolean indicate player is or not bankrupt.

2.9.2 Constructors

+ Player(String name, int money, Color color, Image img)	Constructor of Player class.

2.9.3 Methods

+ boolean buy(Asset asset)	Called when buy the asset.
+ void moveTo(int numOfSquare)	Called when want to set a number of move.
+ void doAction()	Override method from Actionable. Called when update logic game.
+ void addMoney(int money)	Called when want to add money.
+ void payMoney(int money)	Called when want to pay money.
+ void setMoney(int money)	Set money of player.
+ void addAsset(Asset asset)	Add asset to player.
+ ArrayList<Asset> getAssets()	Return assets.
+ int getMoney()	Return money of player.
+ String getName()	Return name of player.
+ int getCurrentField()	Return current field of player.
+ void setCurrentField(int currentField)	Set current field of player.
+ int getNextField()	Return next field of player.

+ void setNextField(int nextField)	Set next field of player.
+ double getCenterPx()	Return centerPx.
+ double getCenterPy()	Return centerPy.
+ int getNotMoveCount()	Return notMoveCount.
+ void setNotMoveCount(int notMoveCount)	Set notMoveCount.
+ boolean isMove()	Return boolean that check player is or not move.
+ Color getColor()	Return color of player.
+ boolean isBankrupt()	Return player is or not bankrupt.
+ void setBankrupt(boolean isBankrupt)	Set player is or not bankrupt

2.10 class SpecialAsset implements Actionable

2.10.1 Fields

- String name	String indicate name of special asset.
- int type	Integer indicate type of special asset.

2.10.2 Constructors

+ SpecialAsset(String name, int typeAsset)	Constructor of SpecialAsset.

2.10.3 Methods

+ void doAction()	Override method from Actionable. Called when update logic game. It do action different due to type of special asset.
+ int getType()	Return type of special asset.
+ String getName()	Return name of special asset.

3. Package Application

3.1 Class SharedObjectHolder

3.1.1 Fields

+ ArrayList<Color> *characterColors*	An instance of ArrayList that store Color.
+ Image	An instance of Image.
+ AudioClip	An instance of AudioClip.
+ MediaPlayer *BGGameMusic*	An instance of MediaPlayer.

3.1.2 Constructor

+ SharedObjectHolder()	Constructor of SharedObjectHolder.

3.1.3 Method

- void loadResource()	Called when load resource that initialize the static object.

4. Package endScene

4.1 **Class** EndScene **extends** Scene

4.1.1 Fields

+ EndScene *scene*	The scene use for end page.
- BorderPane *pane*	The pane use for end page.

4.1.2 Constructor

+ EndScene(Parent p)	Initializes the button to swap to start scene. Display background with name of the winner .

5. package field

5.1 Class AssetShow extends StackPane

5.1.1 Fields

- Player player	An instance of Player.
- BorderPane imgPane	An instance of BorderPane.
- ImageView profileImg	An instance of ImageView.
- GridPane description	An instance of GridPane.
- Label name	An instance of Label that indicate name of field.
- Label money	An instance of Label that indicate name of money.

5.1.2 Constructor

+ AssetShow(Player player)	Constructor of AssetShow.

5.1.3 Methods

+ void update()	Called when update the money of player.
+ void setUpBankruptPlayer()	Called when want to set player go to bankrupt.
+ Player getPlayer()	Return player that assetShow is showing.

5.2 Class CornerField extends Field

5.2.1 Fields

- double width = 140;	Double indicate width property.
- double height = 140;	Doubel indicate height property.

5.2.2 Constructor

+ CornerField(SpecialAsset asset, Image img)	Constructor of CornerField.

5.2.3 Methods

+ double getHeightField()	Return height of field.
+ double getWidthField()	Return width of field.
+ void eventAction()	Called by doAction().

5.3 Class DiceField extends VBox

5.3.1 Fields

- StackPane diceFacePane	An instance of StackPane that indicate Dice Face when rolling.
- ArrayList<Image> diceFaces	An instance of ArrayList that store an Image.
- ImageView diceFace	An instance of ImageView that is face of dice.
- double width = 140	Double indicate width property of DiceField
- double height = 140	Double indicate height property of DiceField.

5.3.2 Constructor

+ DiceField()	Constructor of DiceField.

5.4 Enumeration Direction

5.4.1 Values

LEFT, ***RIGHT***, ***UP***, ***DOWN***	Values of Direction.

5.5 Abstract Class Field extends BorderPane implements Actionable

5.5.1 Fields

# Actionable actionable	An instance of Actionable.
# Color color	An instance of Color.

5.5.2 Methods

+ *void eventAction()*	Called by doActioon().
+ void doAction()	Override method from Actionable. Called when update logic game. It invoke eventAction().
+ Actionable getActionable()	Return actionable.
+ Color getColor()	Return color.
+ void setColor(Color color)	Set color.

5.6 Class HLandField extends Field

5.6.1 Fields

- VBox vStore	An instance of VBox.
- double width = 140	Double indicate width property.
- double height = 100	Doubel indicate height property.

5.6.2 Constructor

+ HLandField(Actionable actionable, Direction dir)	Constructor of HLandField.

5.6.3 Methods

+ double getHeightField()	Return height of field.
+ double getWidthField()	Return width of field.
+ void eventAction()	invoke actionable.doAction().
+ void setOwnerColor()	Set this vStore color with Owner color when buy this field.

5.7 Class InGameMenu extends VBox

5.7.1 Fields

- Button menu	An instance of Button.
- Button sound	An instance of Button.
- SoundMenuStage soundMenuStage	An instance of SoundMenuStage.
- SettingMenuStage settingMenuStage	An instance of SettingMenuStage.
- ImageView settingButton	An instance of ImageView.
- ImageView transparentSettingButton	An instance of ImageView.
- ImageView soundSettingButton	An instance of ImageView.
- ImageView	An instance of ImageView.

transparentSoundSettingButton	

5.7.2 Constructor

+ InGameMenu()	Constructor of InGameMenu.

5.8 Class LogDisplay extends VBox

5.8.1 Fields

- ListView<String> logPane;	An instance of ListView that store String.
- BorderPane titlePane;	An instance of BorderPane.
- int mSize;	Integer indicate size of logPane.

5.8.2 Constructor

+ LogDisplay()	Constructor of LogDisplay.

5.8.3 Methods

+ void add(String msg)	Called when want to add log to LogDisplay.
+ int getSize()	Return mSize.

5.9 Class SettingMenuStage extends Stage

5.9.1 Fields

- int widthButton = 200	Double indicate width property of button.
- int heightButton = 40	Doubel indicate height property of button.

5.9.2 Constructor

+ SettingMenuStage()	Constuctor of SettingMenuStage.

5.10 Class SoundMenuStage extends Stage

5.10.1 Fields

- Slider mainSlider	An instance of Slider.
- Slider bgSlider	An instance of Slider.
- Slider sfxSlider	An instance of Slider.

5.10.2 Constructor

+ SoundMenuStage()	Constructor of SoundMenuStage.

5.11 Class VLandField extends Field

5.11.1 Fields

- HBox hStore	An instance of HBox.
- double width = 100	Double indicate width property.

- double height = 140	Doubel indicate height property.

5.11.2 Constructor

+ VLandField(Actionable actionable, Direction dir)	Constructor of VLandField.

5.11.3 Methods

+ double getHeightField()	Return height of field.
+ double getWidthField()	Return width of field.
+ void eventAction()	invoke by doAction().
+ void setOwnerColor()	Set this hStore color with Owner color when buy this field.

6. Package gameScene

6.1 Class BoardPane extends GridPane

6.1.1 Fields

- int *numOfField* = 24	Integer indicate a number of field in BoardPane.
- VBox	An instance of VBox.
- HBox	An instance of HBox.
- ArrayList<Field> fields	An instance of ArrayList that store field.
- HLandField	An instance of HLandField.
- VLandField	An instance of VLandField.
- CornerField	An instance of CornerField.
- DiceField diceField	An instance of DiceField.
- ChanceCard chanceCard	An instance of ChanceCard.
- CommunityChest communityChest	An instance of CommunityChest.

6.1.2 Constructor

+ BoardPane()	Constructor of BoardPane.

6.1.3 Methods

+ void init()	Method initialize the board pane call after initialize all instance.
+ ArrayList<Field> getFields()	Return fields

+ int getNumoffield()	Return numOfField.

6.2 Class CheckException extends Exception

6.2.1 Fields

- long *serialVersionUID* = -2099410559888851737L	serialVersionUID

6.2.2 Constructor

+ CheckException(String message)	Exception for check correct input pane

6.3 Class GameScene extends Scene

6.3.1 Fields

- BoardPane board	An instance of BoardPane.
- BorderPane pane	An instance of BorderPane.
- BorderPane bLeft	An instance of BorderPane.
- BorderPane bRight	An instance of BorderPane.
- ArrayList<AssetShow> assetShows	An instance of ArrayList that store AssetShow.
- InGameMenu inGameMenu	An instance of InGameMenu
- LogDisplay logDisplay	An instance of LogDisplay.

6.3.2 Constructor

+ GameScene(Parent root) throws CheckException	Constructor of GameScene.

6.3.3 Methods

+ void resetGame() throws CheckException	Called when want to reset game.
+ void setUpAssetShow()	Called for set up Asset show.
+ void setUpUserControl()	Called for set up User control.
+ void setUpLogDisplay()	Called for set up Log display.
+ void update()	Called for update all AssetShow in assetShows.
+ BoardPane getBoard()	Return board.
+ BorderPane getPane()	Return pane.
+ LogDisplay getLogDisplay()	Return logDisplay.
+ ArrayList<AssetShow> getAssetShows()	Return assetShows.

7. Package startScene

7.1 Class StartScene extends Scene

7.1.1 Fields

+ StartScene *scene*	The scene use for start page.
- BorderPane *pane*	The pane use for start page.
- int *player*	Number of players.

7.1.2 Constuctor

+ StartScene(Parent p)	-Initializes scene and pane. -Initializes start button for show Button for choosing the number of players. -Initializes Button for setting number of players.

7.1.3 Methods

+ void resetStartScene()	Called when want to reset scene.
+ int getPlayer()	Return number of players.