COMP6710 STRUCTURED PROGRAMMING

Assignment 2 Presentation

Group tue 12 g

Group Members

- HongGic, Oh, u7234659 - Develop GUI
- Juren Xu, u7149851 - Develop Methods
- Xindi Liu, u6225282 - Develop Methods

What did we do for this assignment?

- Class

  Object oriented based programming to build our game

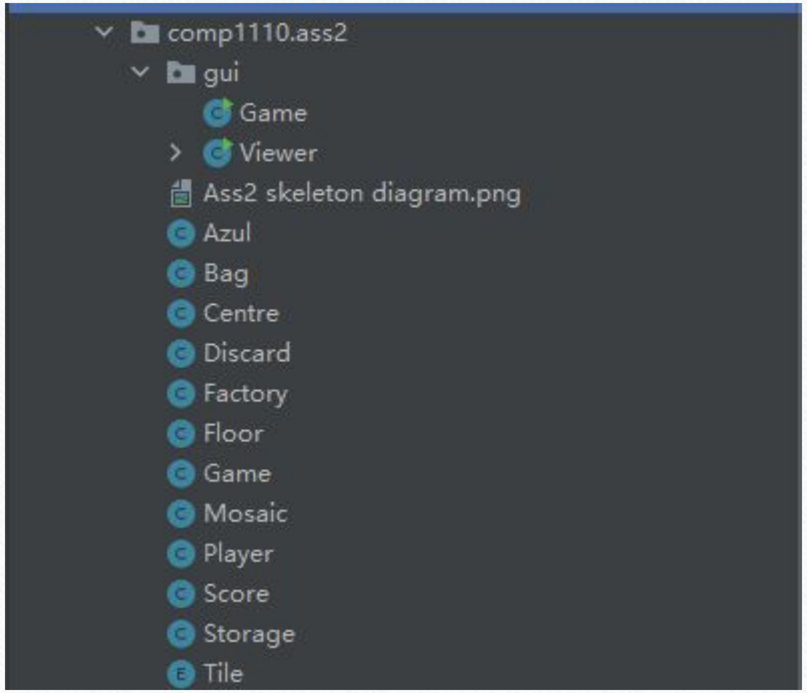
  Finish task 2~14

- GUI

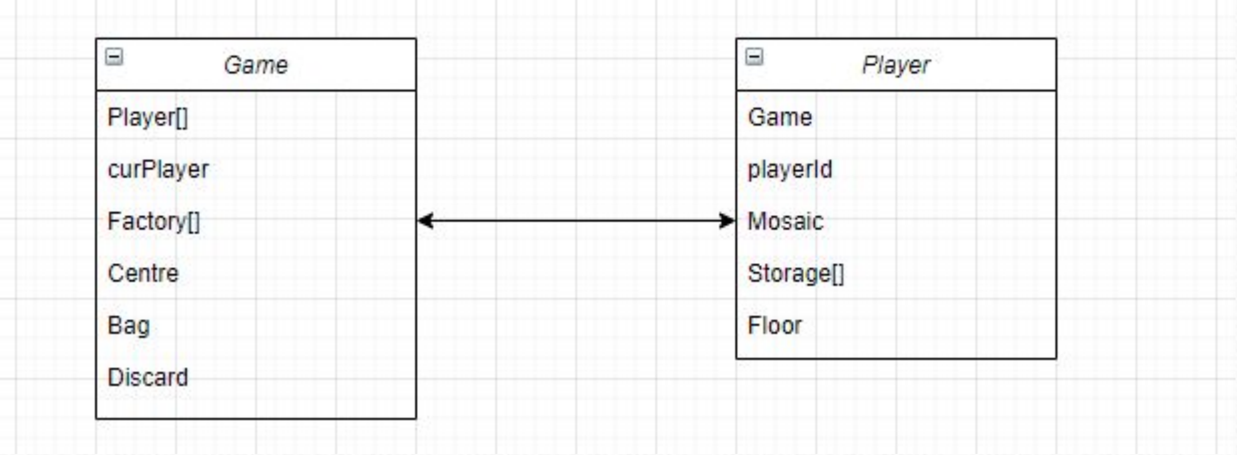
  Use various functions(polygon,stackpane,group,textfield,image, button..)

  Enhance the GUI by our ideas

Design Approach



```
comp1110.ass2
  gui
    Game
    Viewer
  Ass2 skeleton diagram.png
  Azul
  Bag
  Centre
  Discard
  Factory
  Floor
  Game
  Mosaic
  Player
  Score
  Storage
  Tile
```



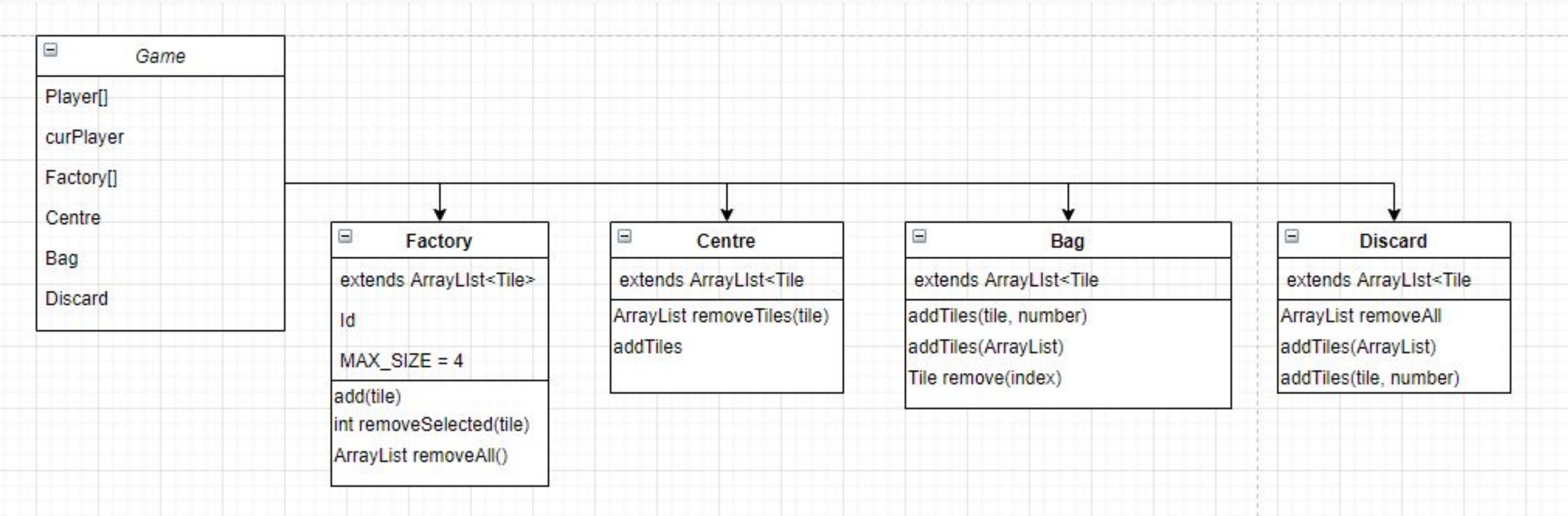
Game

- Player[]
- curPlayer
- Factory[]
- Centre
- Bag
- Discard

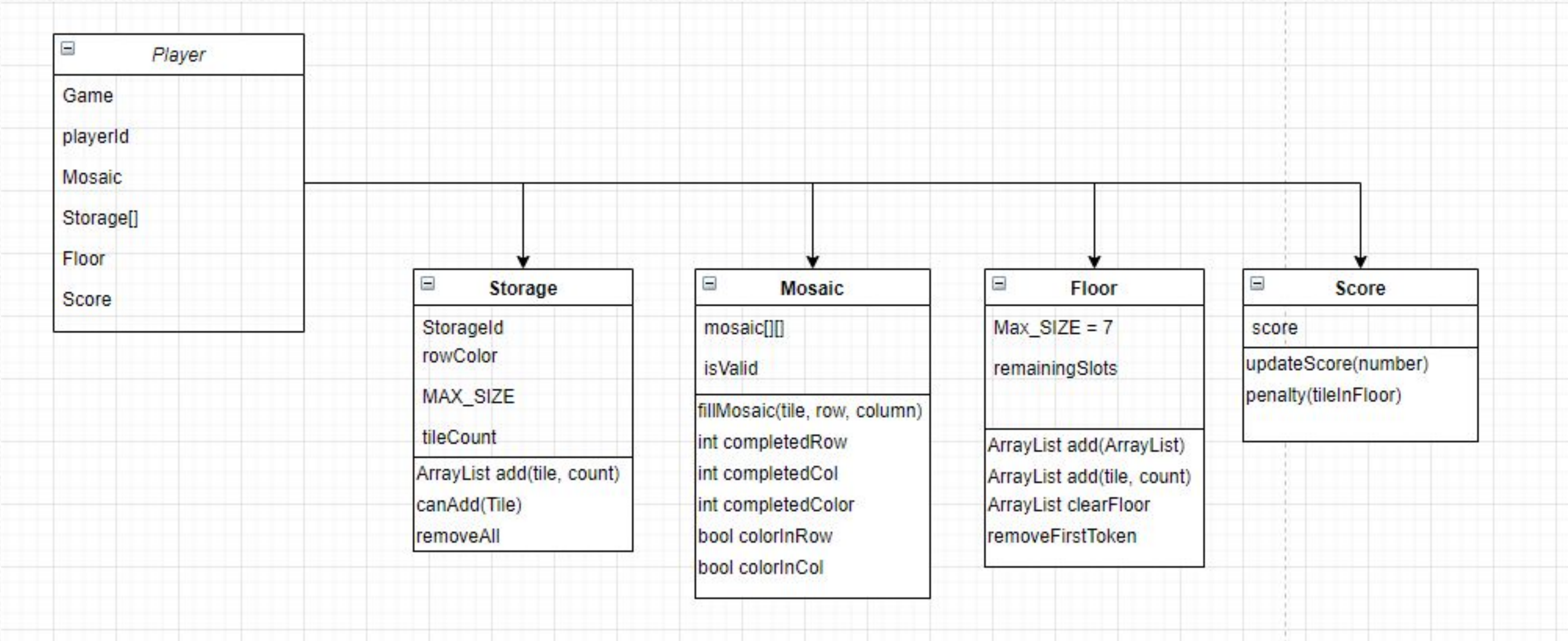
Player

- Game
- playerId
- Mosaic
- Storage[]
- Floor

Game Board Design



Player Design



Player

Game

playerId

Mosaic

Storage[]

Floor

Score

Storage

StorageId

rowColor

MAX_SIZE

tileCount

ArrayList add(tile, count)

canAdd(Tile)

removeAll

Mosaic

mosaic[][]

isValid

fillMosaic(tile, row, column)

int completedRow

int completedCol

int completedColor

bool colorInRow

bool colorInCol

Floor

Max_SIZE = 7

remainingSlots

ArrayList add(ArrayList)

ArrayList add(tile, count)

ArrayList clearFloor

removeFirstToken

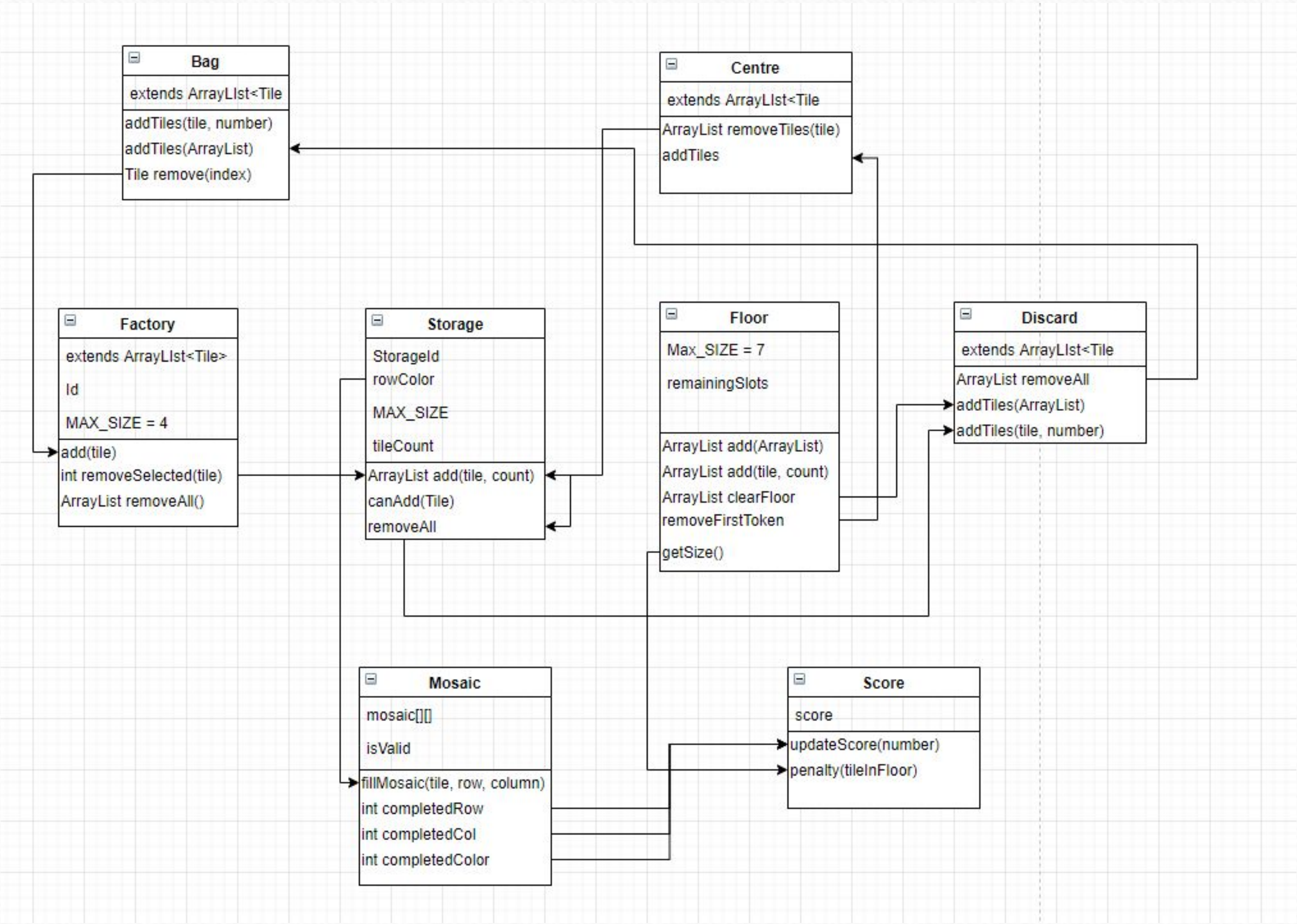
Score

score

updateScore(number)

penalty(tileInFloor)

Connection of classes



Bag

extends ArrayLIst<Tile

addTiles(tile, number)

addTiles(ArrayList)

Tile remove(index)

Centre

extends ArrayLIst<Tile

ArrayList removeTiles(tile)

addTiles

Factory

extends ArrayLIst<Tile>

Id

MAX_SIZE = 4

add(tile)

int removeSelected(tile)

ArrayList removeAll()

Storage

StorageId

rowColor

MAX_SIZE

tileCount

ArrayList add(tile, count)

canAdd(Tile)

removeAll

Floor

Max_SIZE = 7

remainingSlots

ArrayList add(ArrayList)

ArrayList add(tile, count)

ArrayList clearFloor

removeFirstToken

getSize()

Discard

extends ArrayLIst<Tile

ArrayList removeAll

addTiles(ArrayList)

addTiles(tile, number)

Mosaic

mosaic[][]

isValid

fillMosaic(tile, row, column)

int completedRow

int completedCol

int completedColor

Score

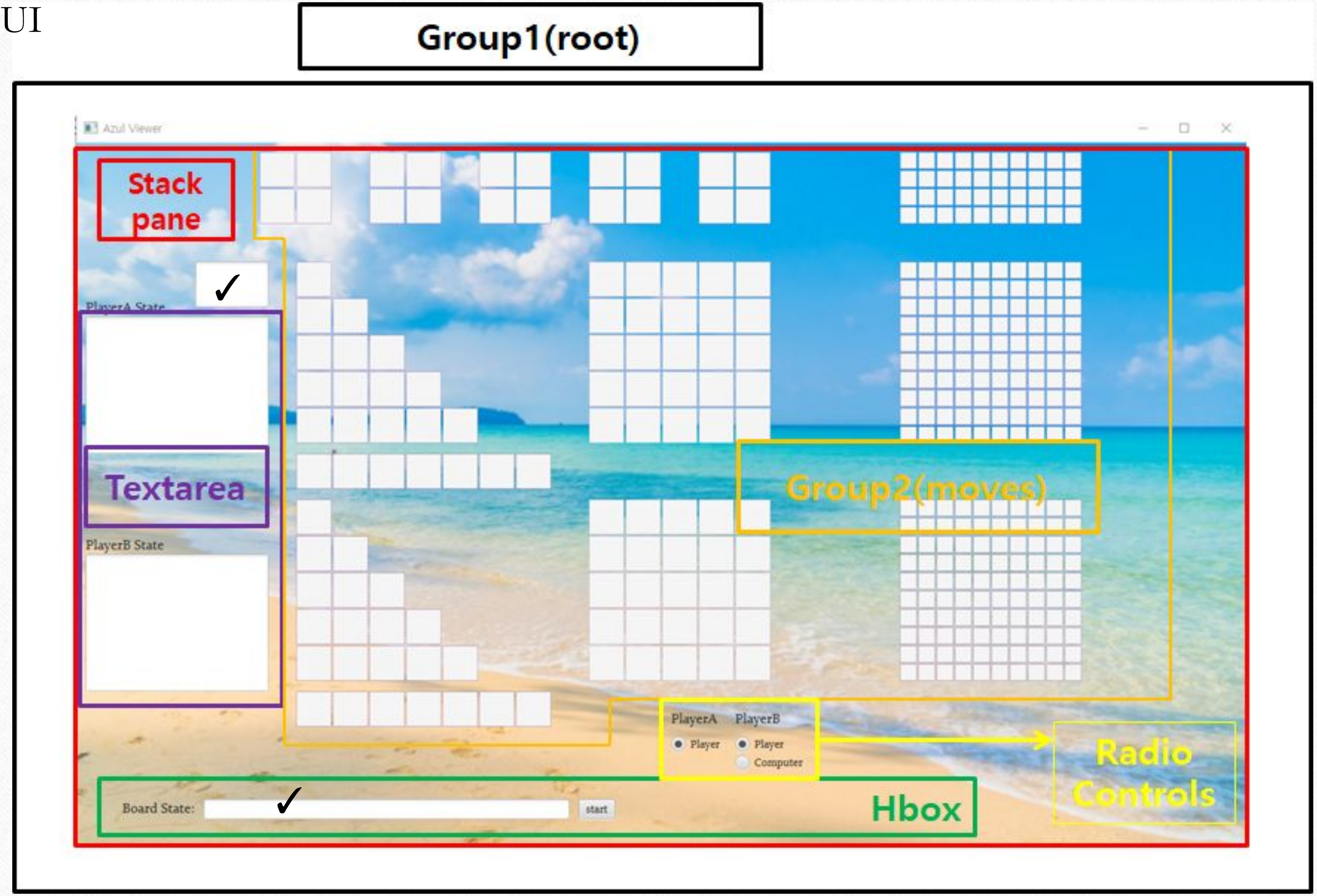
score

updateScore(number)

penalty(tileInFloor)

Layout of GUI

✓ : textfield



PlayerA PlayerB

● Player ● Player

○ Computer

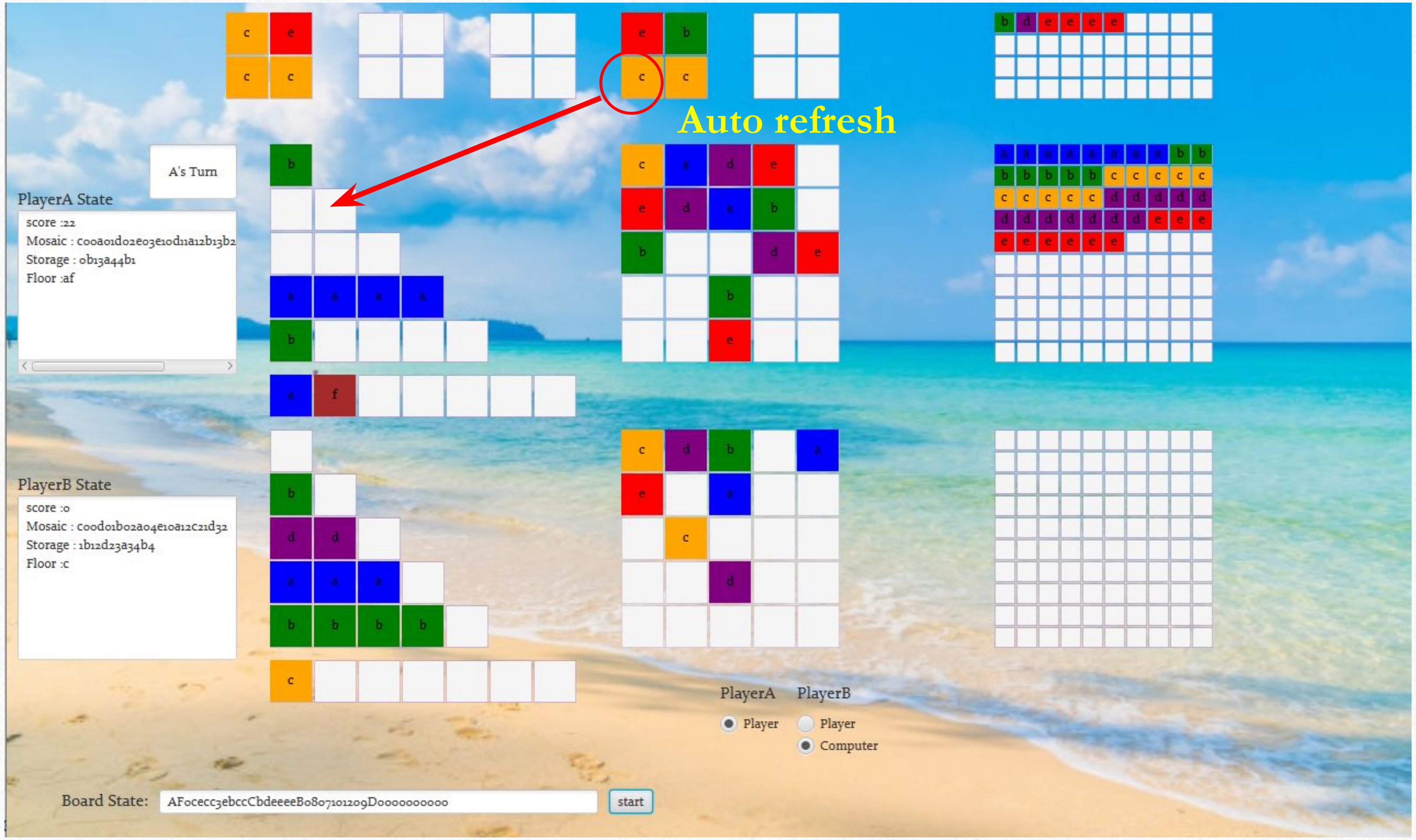
Problem 1

How can I draw the tiles? If I draw tiles separately, code length would be long

Section	Map Index	X axis Start	Y axis Start	X axis End	Y axis End	Tile size	column	row
Centre	0	900	10	1100	100	20	4	10
Bag	1	900	130	1100	340	20	10	10
Discard	2	900	390	1100	600	20	10	10
Factory1	3	200	10	280	100	40	2	2
Factory2	4	320	10	400	100	40	2	2
Factory3	5	440	10	520	100	40	2	2
Factory4	6	560	10	640	100	40	2	2
Factory5	7	680	10	760	100	40	2	2
A mosaic	8	560	130	760	340	40	5	5
B mosaic	9	560	390	760	600	40	5	5
A storage	10	240	130	440	340	40	5	0
B storage	11	240	390	440	600	40	5	0
A floor	12	240	340	520	380	40	1	7
B floor	13	240	600	520	640	40	1	7

Problem 2

Try to delete refresh button. When we are playing the game, click the refresh button is really tiresome



Auto refresh

A's Turn

PlayerA State

score :22
Mosaic : c00a01d02e03e10d11a12b13b2
Storage : 0b13a44b1
Floor :af

PlayerB State

score :0
Mosaic : c00d01b02a04e10a12c21d32
Storage : 1b12d23a34b4
Floor :c

PlayerA: ● Player

PlayerB: ○ Player ● Computer

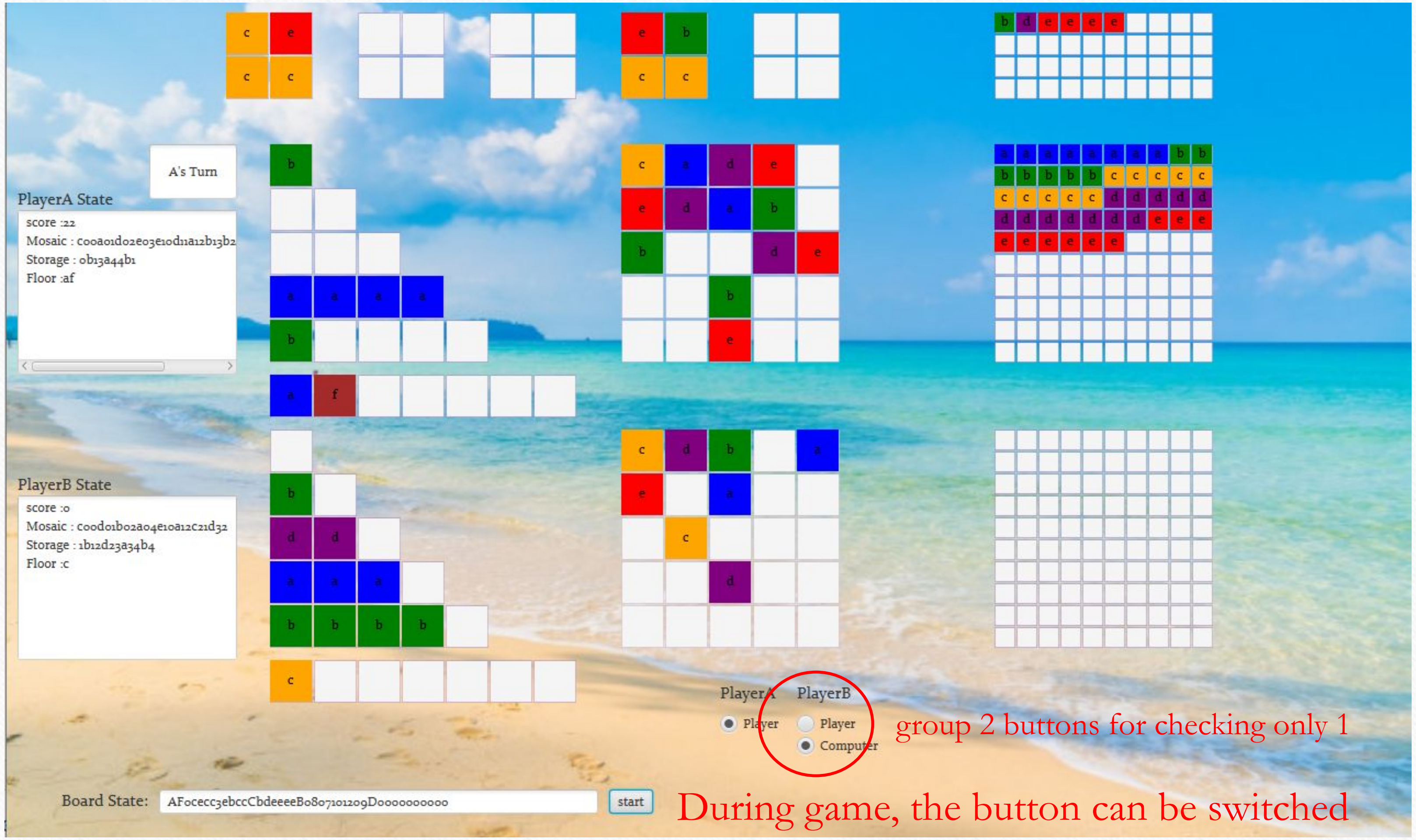
Board State: AF0cecc3ebccCbdeeeeB0807101209D0000000000

What we do?

- Remove refresh button
- Update status automatically

Problem 3

Switch Player B between computer and User.



A's Turn

PlayerA State

score :22
Mosaic : c00a01d02e03e10d11a12b13b2
Storage : 0b13a44b1
Floor :af

PlayerB State

score :0
Mosaic : c00d01b02a04e10a12c21d32
Storage : 1b12d23a34b4
Floor :c

PlayerA	PlayerB
● Player	○ Player
	● Computer

Board State: AF0cecc3ebccCbdeeeeB0807101209D0000000000 start

group 2 buttons for checking only 1

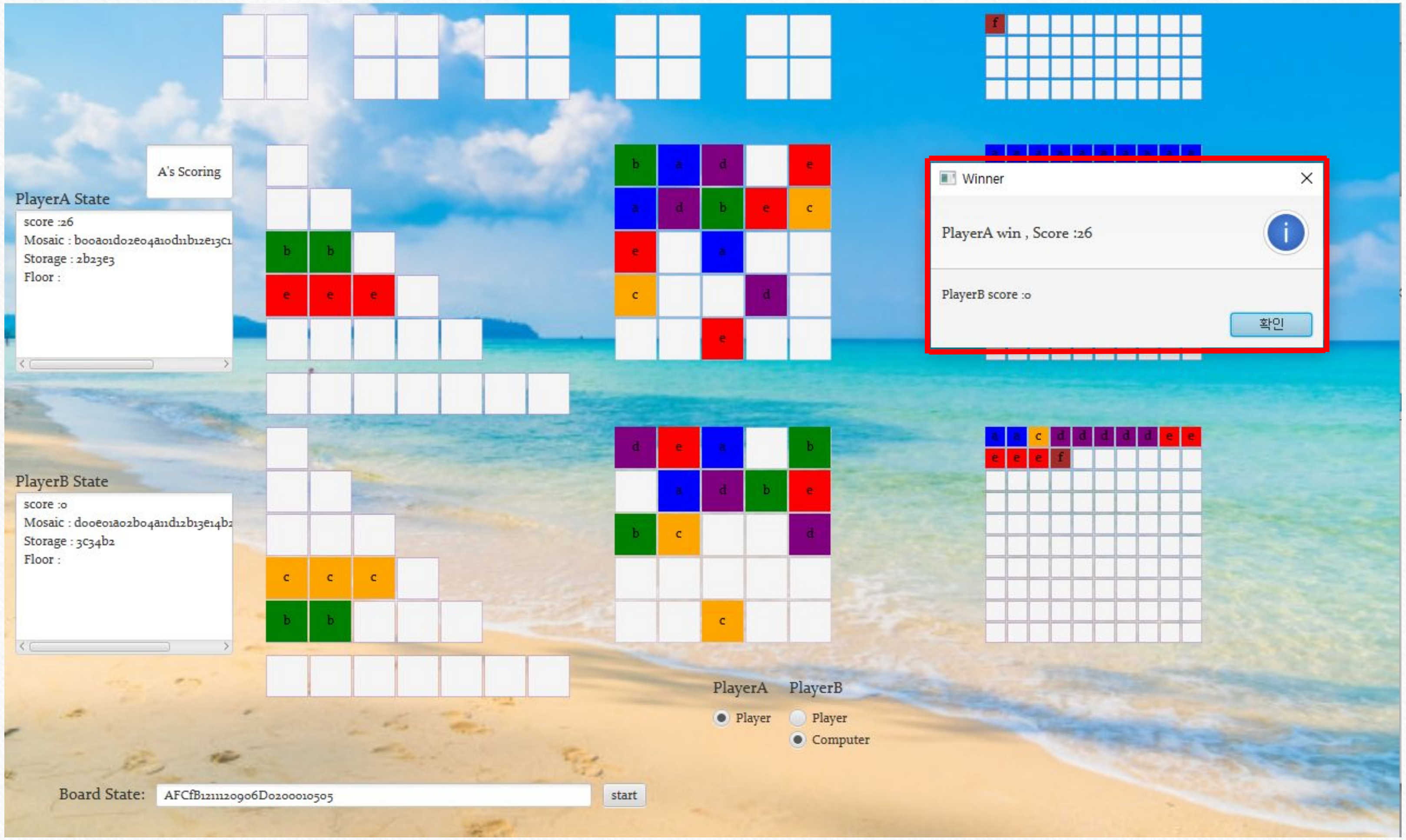
During game, the button can be switched

What we do?

- Add radio button to switch Player/Computer by radio button
- only allow choose one in the button

Problem 4

When the game is finished, I want to show the result with popup.



PlayerA State

score :26
Mosaic : b00a01d02e04a10d11b12e13c1
Storage : 2b23e3
Floor :

PlayerB State

score :0
Mosaic : d00e01a02b04a11d12b13e14b2
Storage : 3c34b2
Floor :

Winner

PlayerA win , Score :26

PlayerB score :0

PlayerA ● Player

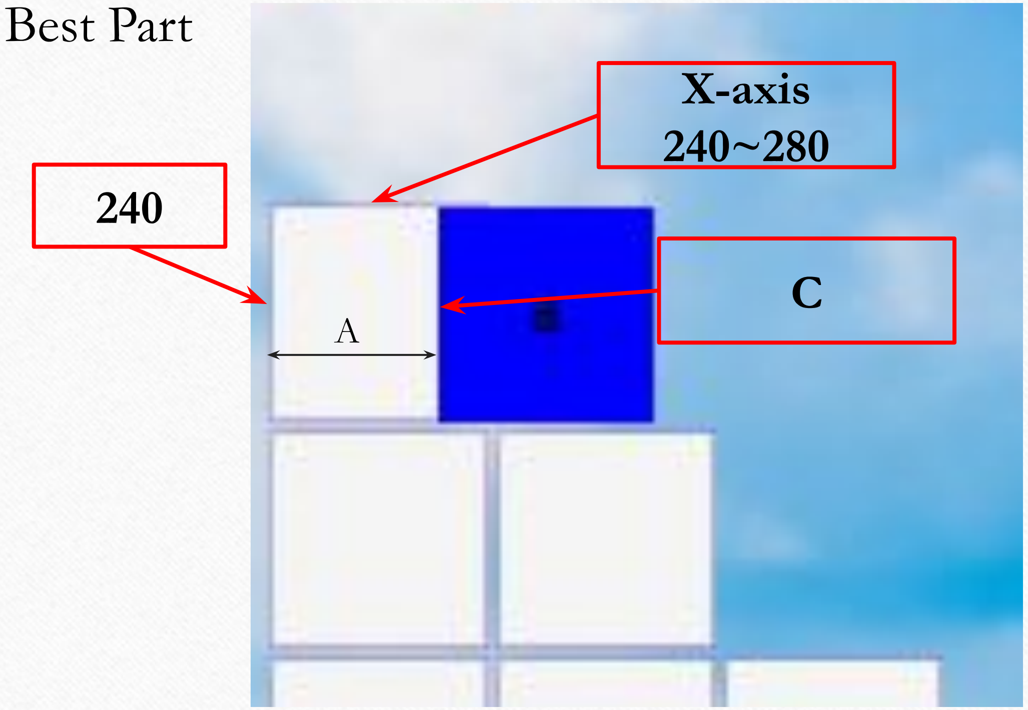
PlayerB ○ Player ● Computer

Board State: AFCfB1211120906D0200010505

What we do?

- Add alert function
- Final score and winner in pop-up

Best Part



C = Current tile's X-layout
X-axis = 240~280
A(Gap) = (C – 240)%40
B(Standard value) = side /3 = 40/3 = 13.33

If(A>B)
X layout = C + Side – A
else(A<B)
X layout = C -A

Y-axis runs with same logic

Case 1
C=270, A =30, B = 13.33(A>B)
X layout = C + Side – A = 270+40-30 = 280 -> **OUT**

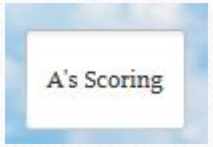
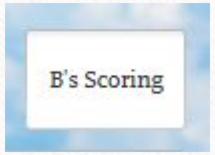
Case 2
C= 250, A= 10, B = 13.33 (A<B)
C = 250-10 = 240 -> **IN**

Case 3
C = 220, A = -20, B = 13.33 (A<B)
C = 220-(-20) = 240 -> **IN**

Best Part

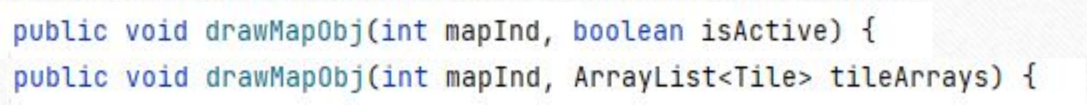
Showing player turn with 4 type

Change Status Automatically

 Factory,Centre → Storage(A),Floor(A)	 Storage(A)→Mosaic(A)
 Factory,Centre → Storage(B),Floor(B)	 Storage(B)→Mosaic(B)

Further improvement

- Clear unnecessary code
- Combine similar methods
- Make playable with 4 Players



```java
public void drawMapObj(int mapInd, boolean isActive) {
public void drawMapObj(int mapInd, ArrayList<Tile> tileArrays) {
```

Credit : Opensource
Q & A icon | Free icon rainbow | Over 4500 royalty free icons (free-icon-rainbow.com)