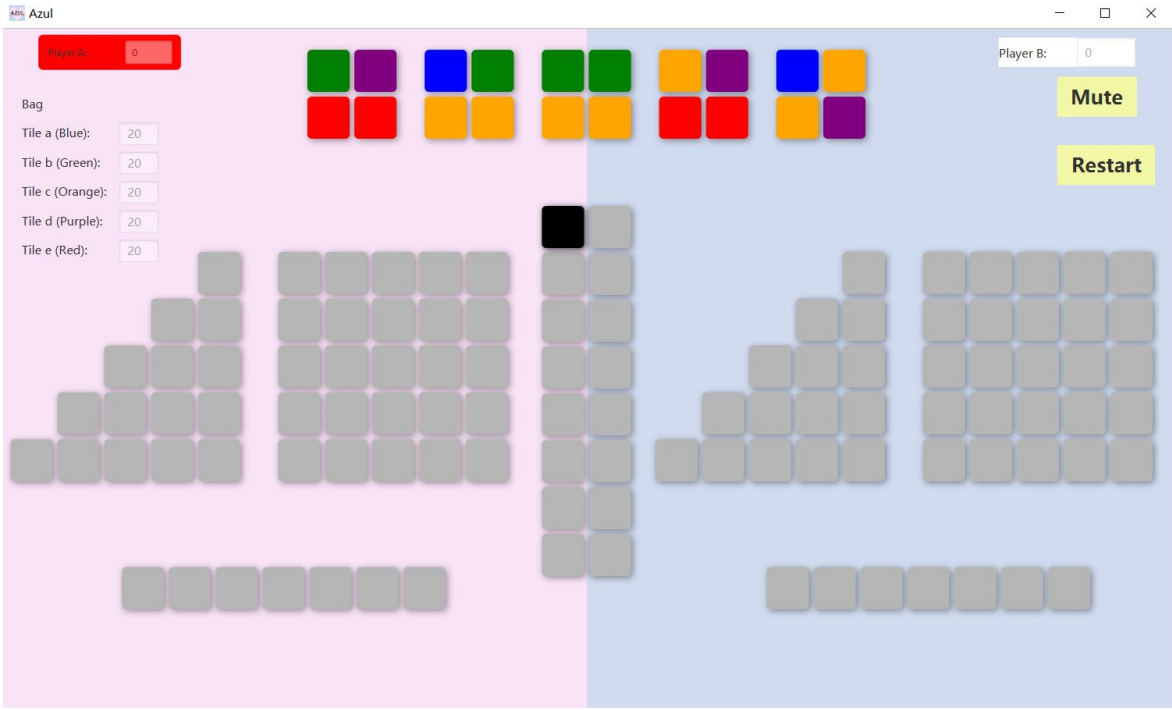
Assignment Two: Azul

Ruizheng Shen (u6814863)
Xiao Xu (u7205634)
Yixin Ge (u6616522)

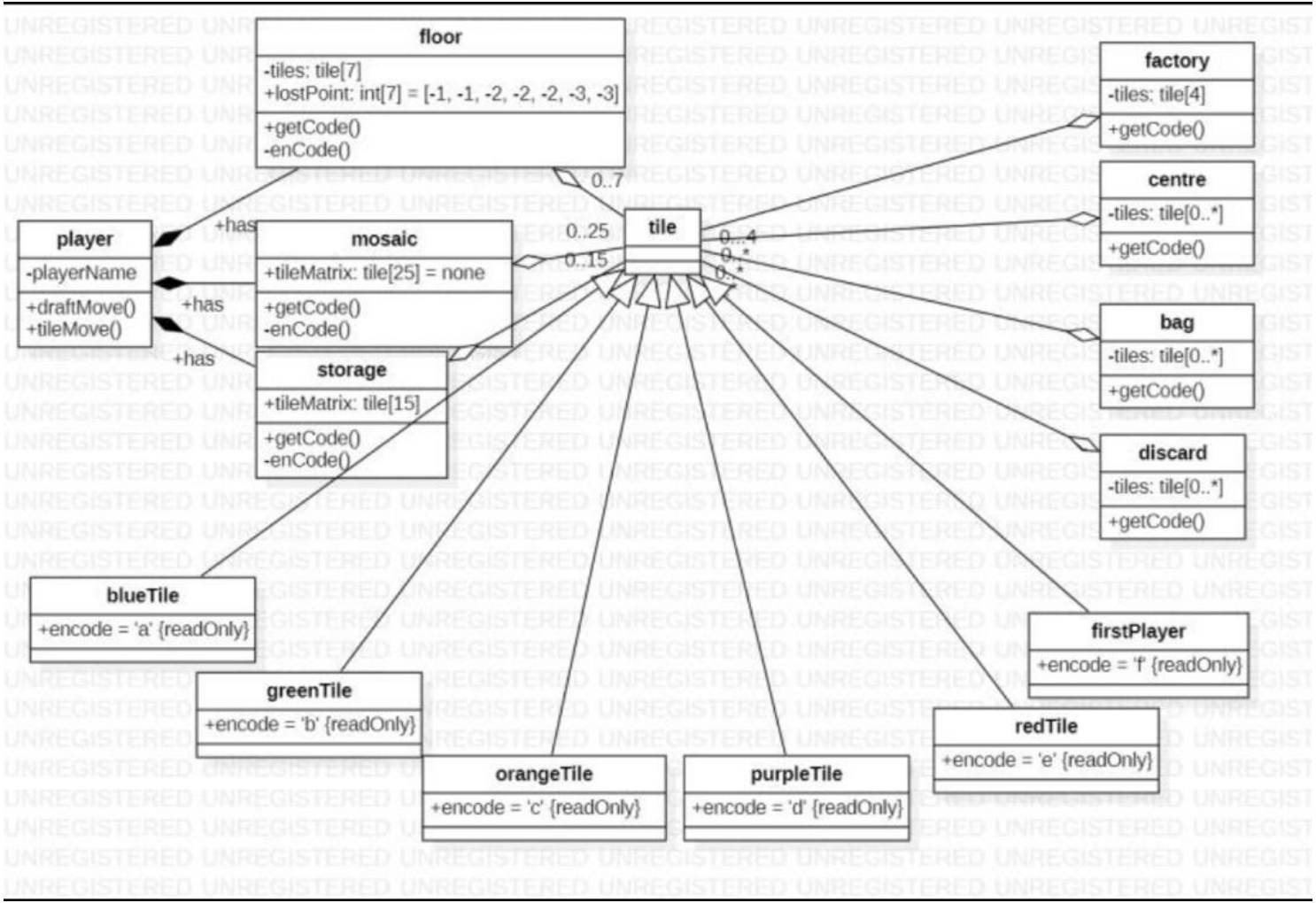
Summary

The Azul game we developed was designed to be a one-on-one strategic board game.

The game was broken into three parts, the GUI parts to show the current state of the game; the Control parts to control the process of the game; and the Member parts to represent the logic and behaviours of each actor(Storage, Mosaic, Factory, etc).



Skeleton diagram



Design Skeleton

Member - Storage, Mosaic, Floor, Center, Factories, Bag, Discard

Represents the the logic of each actor.

Azul - isSharedStateValid, isPlayerStateValid, refillFactory, getBonusScore, isMoveValid, applyMove

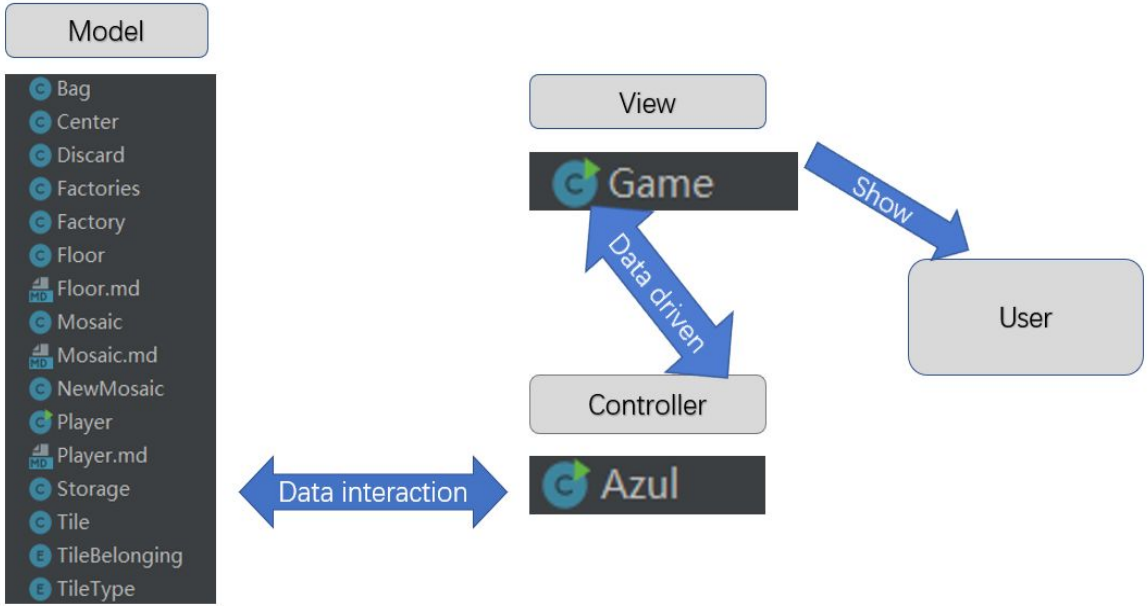
Control the game process.(by processing the String of game state)

GUI - Game

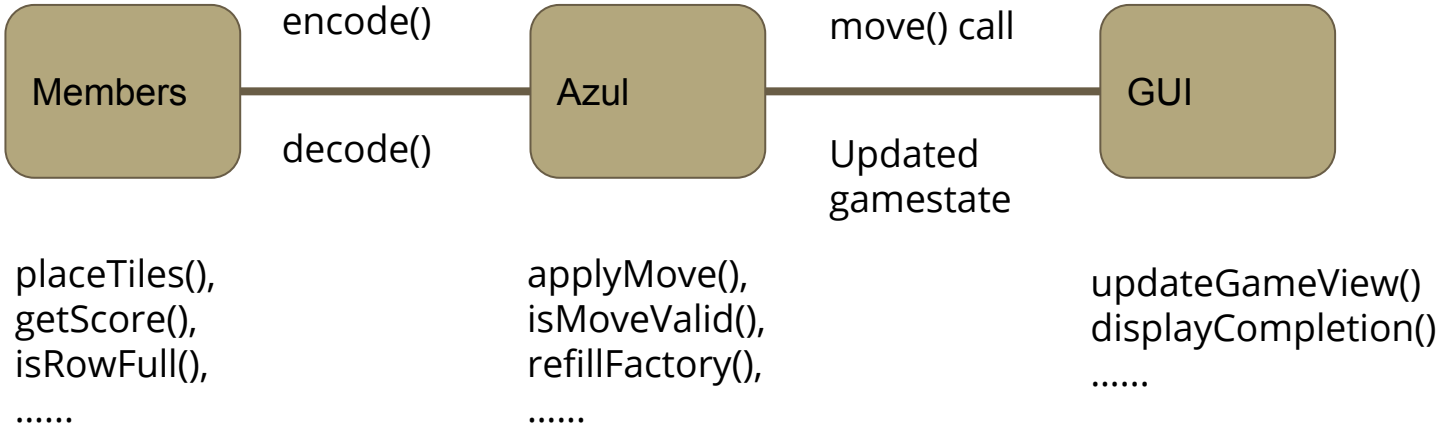
Visualize the game state.(by visualize the abstract game state String)

Diagram

There is the diagram about the relationship between user,members, Game and Azul.



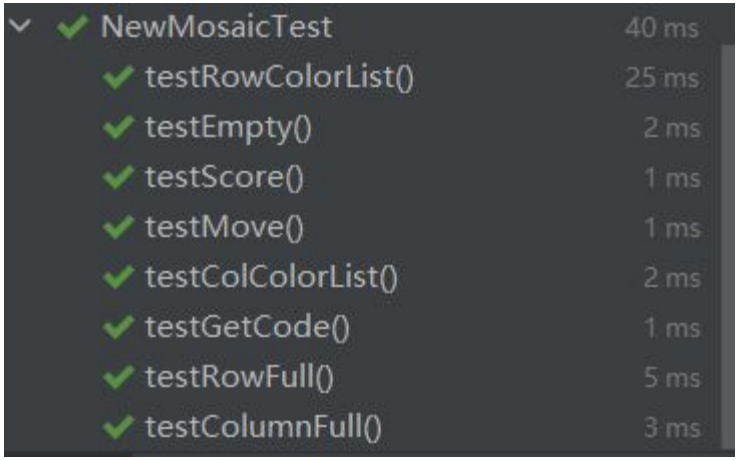
Design Ideas

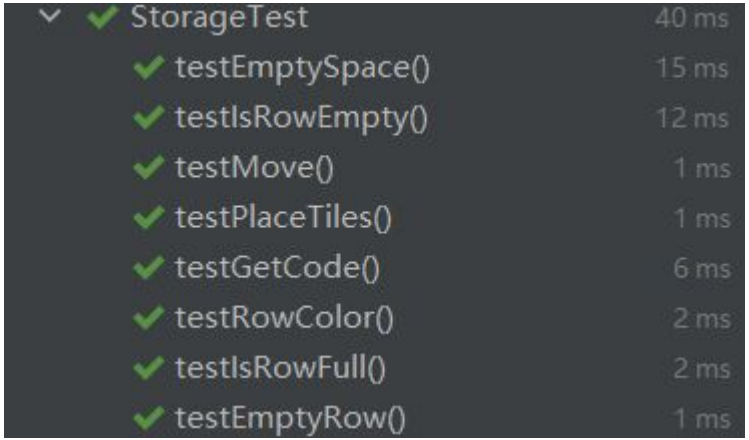


Checking

We designed test cases in three aspects:

1) Normal conditions.
2) Empty conditions.
3) Full conditions.
4) Wrong conditions.



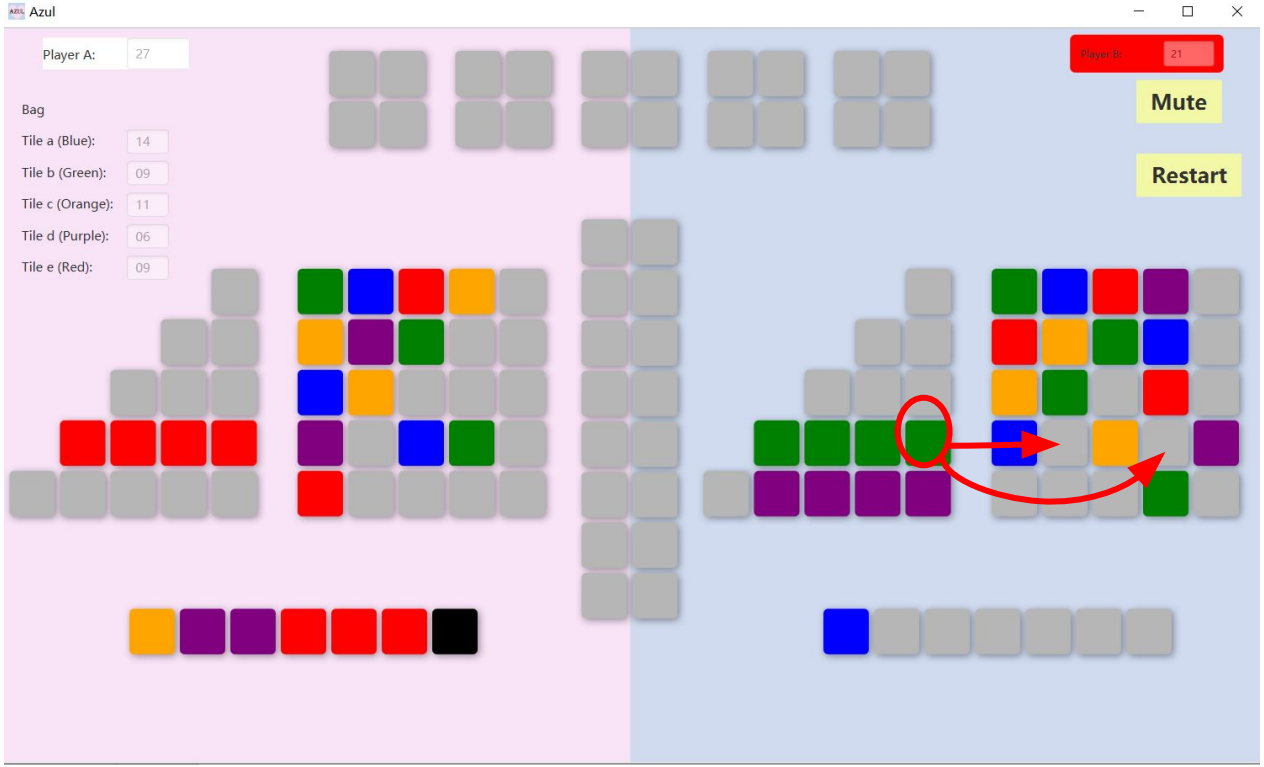


Tricky rules

Case:

Green tile cannot move to Mosaic.

We used console to debug the game state step by step.



How can we improve code

In order to expand the game from two-player mode to multi-player mode. We need to define the position information a more flexible way.

(Use a list to store the position information, and use a initialize method to initialize the position according to the number of players.)

```
private static final int AMOSAIC_X_LAYOUT = 280; // XIndex of mosaic of player A ,player B +600
private static final int AMOSAIC_Y_LAYOUT = 227; // YIndex of mosaic/storage of A and B player
private static final int ASTORAGE_X_LAYOUT = 198; // XIndex of storage of A, B +600
private static final int ASTORAGE_Y_LAYOUT = 227; //YIndex of storage of A
private static final int AFLOOR_X_LAYOUT = 120; //XIndex of floor of A, B+600
private static final int AFLOOR_Y_LAYOUT = 550; //YIndex of floor of A and B
private static final int BMOSAIC_X_LAYOUT = 940; // XIndex of mosaic of player B
private static final int BMOSAIC_Y_LAYOUT = 227; // YIndex of mosaic of player B
private static final int BSTORAGE_X_LAYOUT = 858; // XIndex of storage of B
private static final int BSTORAGE_Y_LAYOUT = 227; //YIndex of storage of B
private static final int BFLOOR_X_LAYOUT = 780; //XIndex of floor of B
private static final int BFLOOR_Y_LAYOUT = 550; //YIndex of floor of B
private static final int CENTER_X_LAYOUT = 550; //XIndex of center
private static final int CENTER_Y_LAYOUT = 180; //YIndex of center
private static final int FACTORIES_X_LAYOUT = 310;
private static final int FACTORIES_Y_LAYOUT = 20; //YIndex of Factories
private static final int NUMBER_OF_TILETYPE = 5;
private final Label[] bagLable = new Label[NUMBER_OF_TILETYPE]; // included in player board.
private final TextField[] bagField = new TextField[NUMBER_OF_TILETYPE]; // included in player board.
private static final int BAG_TEXT_FIELD_X_LAYOUT = 120; // set the position of bag
private static final int BAG_TEXT_FIELD_Y_LAYOUT = 97;
private static final int BAG_TEXT_FIELD_HEIGHT = 20;
private static final int BAG_TEXT_FIELD_WEIGHT = 40;
private static final int BAG_LABEL_X_LAYOUT = 20;
private static final int BAG_LABEL_Y_LAYOUT = 100;
private static final int BAG_X_LAYOUT = 20;
private static final int BAG_Y_LAYOUT = 70;
```

improvement

- Need to be more Object-oriented.

- Avoid processing String to much, hand it to the members.

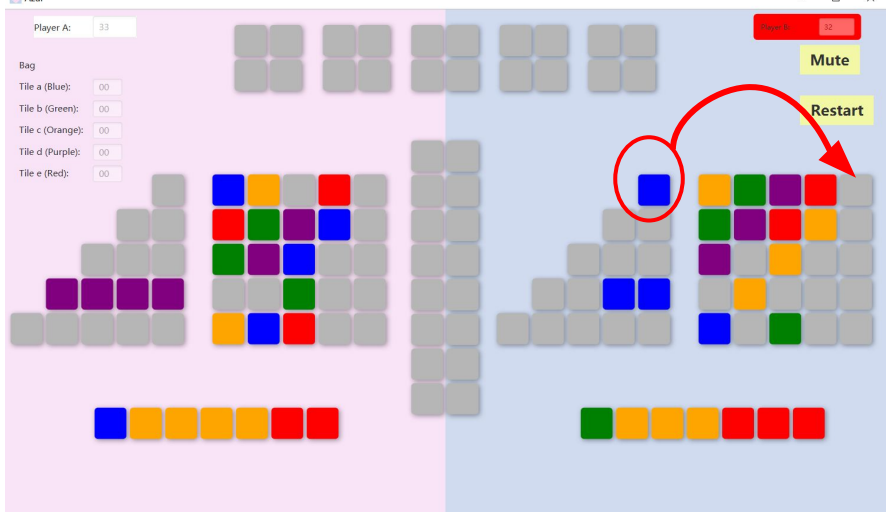
```
 *
 * @param gameState the current game state
 * @return the tile drawn from the bag, or 'Z' if the bag and discard pile are empty.
 * TASK 5
 */
public static char drawTileFromBag(String[] gameState) {
    // FIXME Task 5
    String a = gameState[0];
    a = a.substring(3);
    int b = a.indexOf("B");
    int c = a.indexOf("D");
    String d = a.substring(b, b + 11);
    String e = a.substring(c);
```

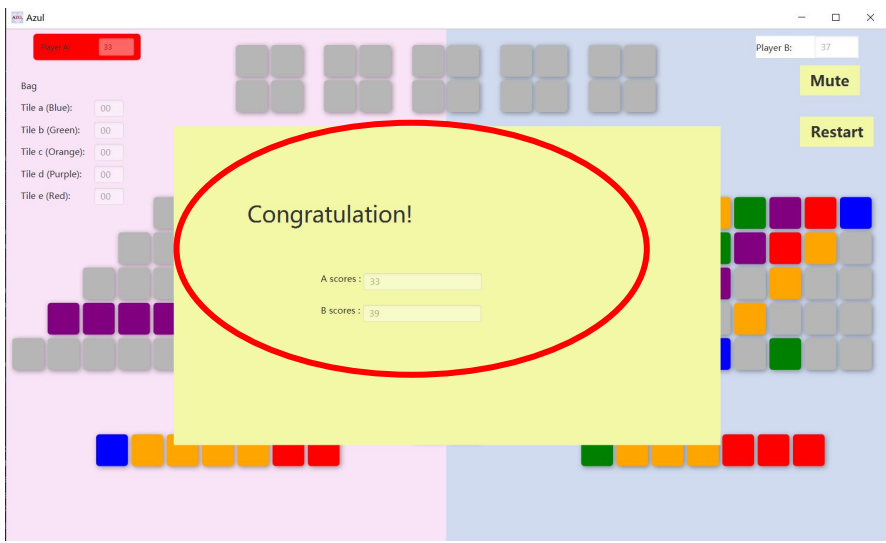
Game End & Score

The Scoreboard shows the total score for each player.

Total score contains:

- Basic point
- Floor point
- Bonus point





Interesting Aspects

1. Background music
2. Tile-move music
3. Background image
4. Icon image
5. ‘Restart’ button

Thanks

Authored by
Ruizheng Shen
Xiao Xu
Yixin Ge