Comp1110/Comp6710 Presentation

Group 12g

Team

Xinyao Wang u6609958

Xinyi Zhang u6976740

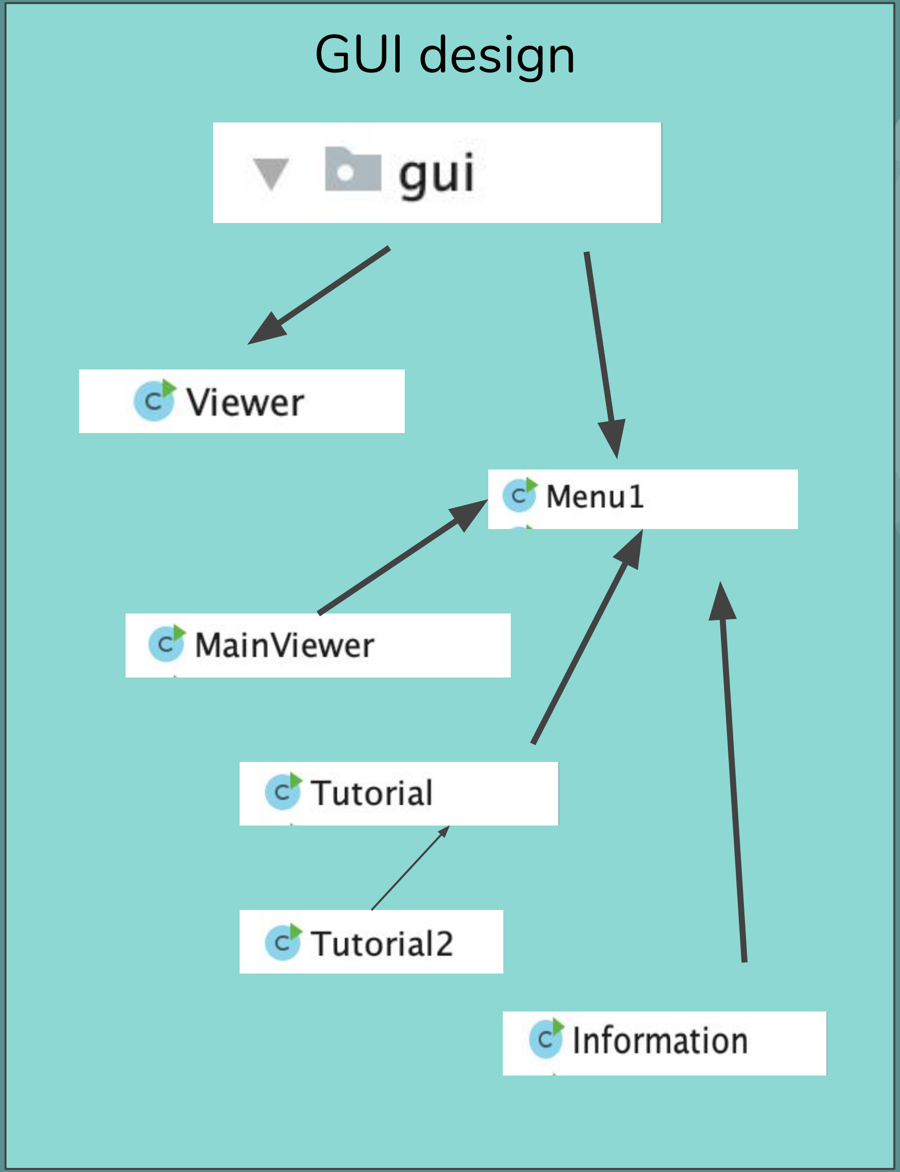
Ruiqiao Jiang u6818746

Structure

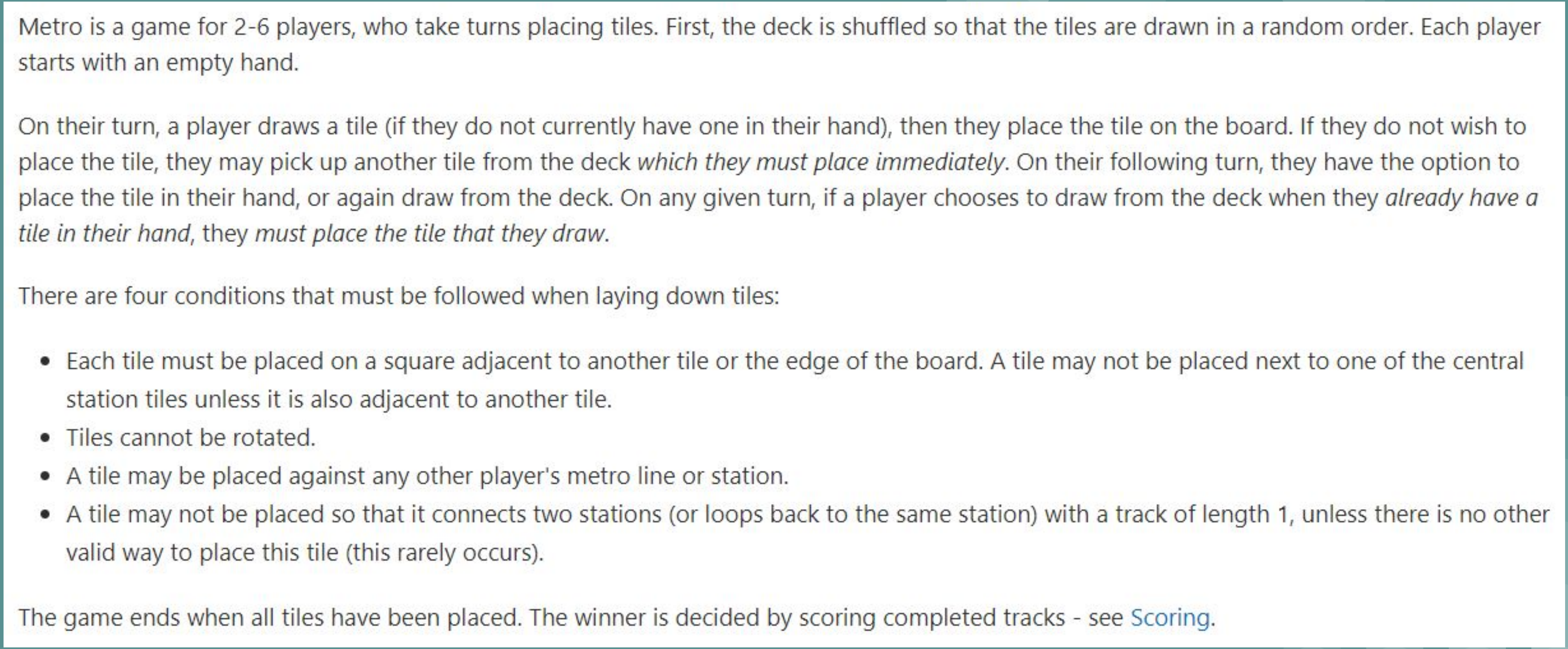
Rules of metro game

code

Diagram


GUI design
gui
Viewer
Menu1
MainViewer
Tutorial
Tutorial2
Information

Rules of Metro Game



Metro is a game for 2-6 players, who take turns placing tiles. First, the deck is shuffled so that the tiles are drawn in a random order. Each player starts with an empty hand.

On their turn, a player draws a tile (if they do not currently have one in their hand), then they place the tile on the board. If they do not wish to place the tile, they may pick up another tile from the deck *which they must place immediately*. On their following turn, they have the option to place the tile in their hand, or again draw from the deck. On any given turn, if a player chooses to draw from the deck when they *already have a tile in their hand*, they *must place the tile that they draw*.

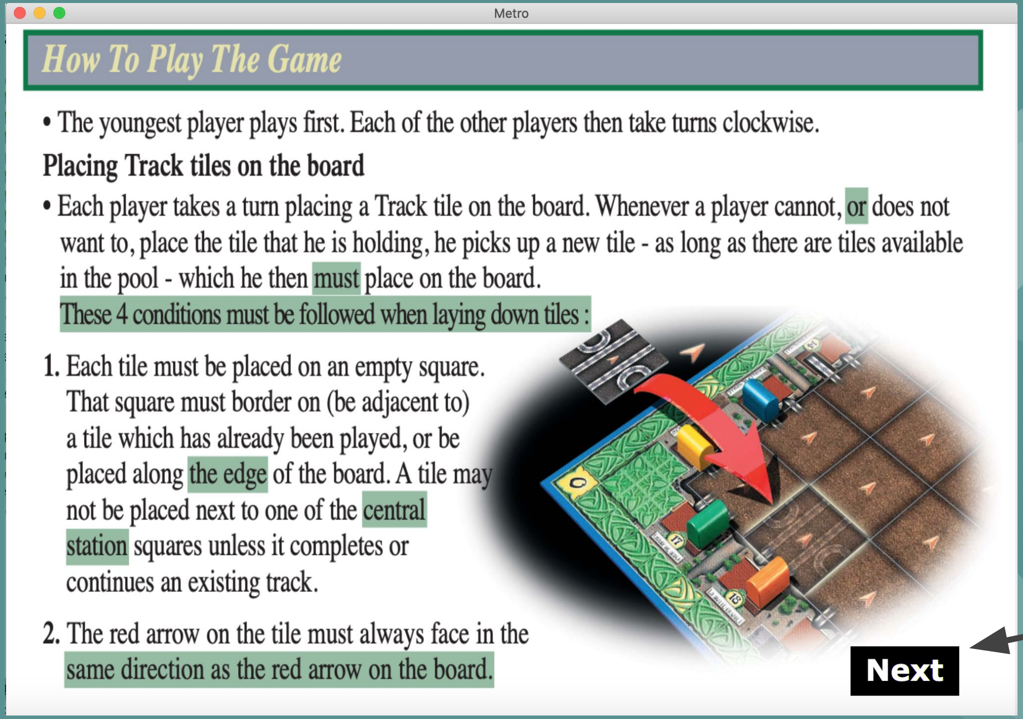
There are four conditions that must be followed when laying down tiles:

- Each tile must be placed on a square adjacent to another tile or the edge of the board. A tile may not be placed next to one of the central station tiles unless it is also adjacent to another tile.
- Tiles cannot be rotated.
- A tile may be placed against any other player's metro line or station.
- A tile may not be placed so that it connects two stations (or loops back to the same station) with a track of length 1, unless there is no other valid way to place this tile (this rarely occurs).

The game ends when all tiles have been placed. The winner is decided by scoring completed tracks - see Scoring.

Tutorial

brief tutorial of the game source from the Metro Queen Games



Metro

How To Play The Game

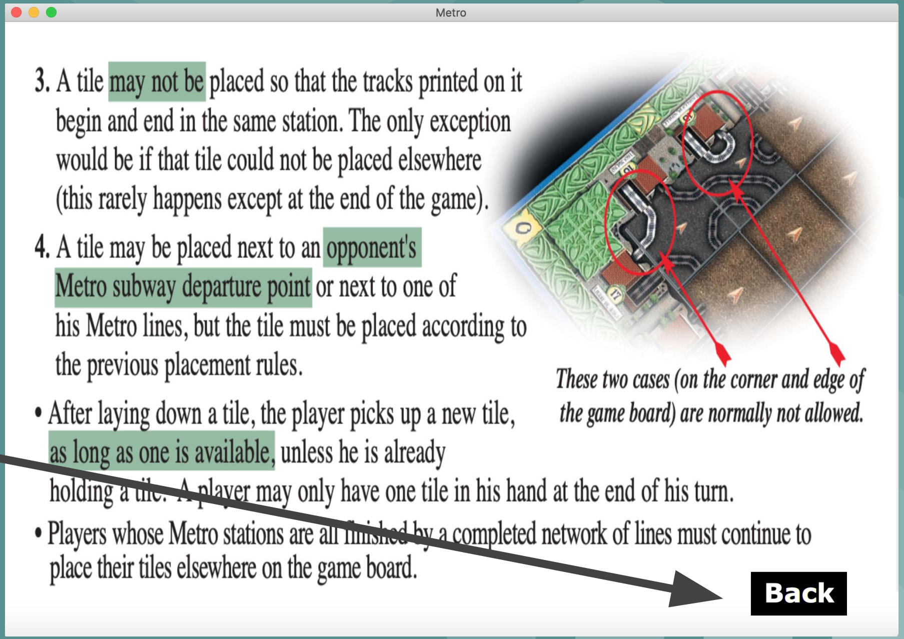
- The youngest player plays first. Each of the other players then take turns clockwise.

Placing Track tiles on the board

- Each player takes a turn placing a Track tile on the board. Whenever a player cannot, or does not want to, place the tile that he is holding, he picks up a new tile - as long as there are tiles available in the pool - which he then must place on the board.
  These 4 conditions must be followed when laying down tiles :

1. Each tile must be placed on an empty square. That square must border on (be adjacent to) a tile which has already been played, or be placed along the edge of the board. A tile may not be placed next to one of the central station squares unless it completes or continues an existing track.
2. The red arrow on the tile must always face in the same direction as the red arrow on the board.

To the next page



Metro

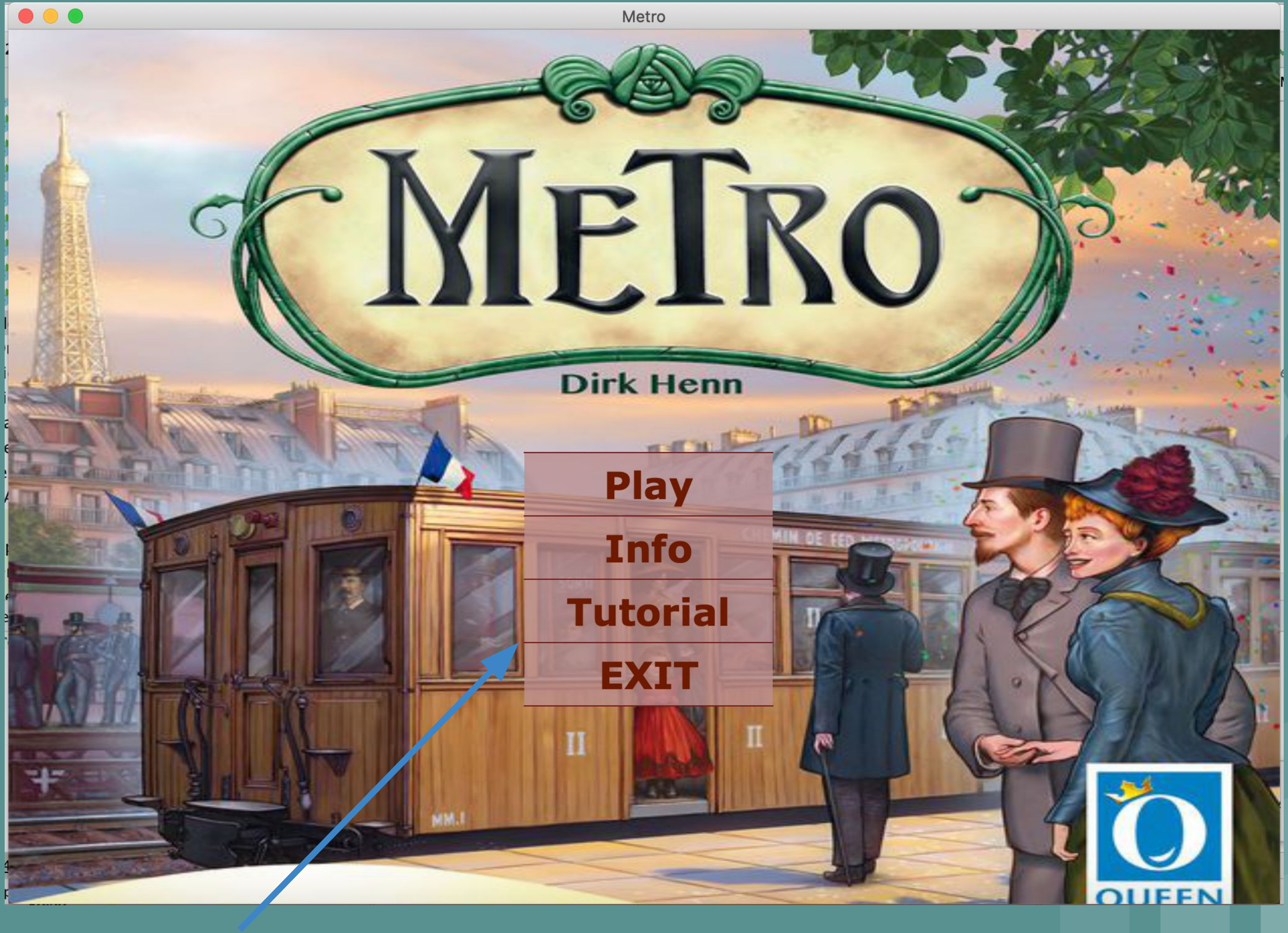
3. A tile may not be placed so that the tracks printed on it begin and end in the same station. The only exception would be if that tile could not be placed elsewhere (this rarely happens except at the end of the game).
4. A tile may be placed next to an opponent's Metro subway departure point or next to one of his Metro lines, but the tile must be placed according to the previous placement rules.

These two cases (on the corner and edge of the game board) are normally not allowed.

- After laying down a tile, the player picks up a new tile, as long as one is available, unless he is already holding a tile. A player may only have one tile in his hand at the end of his turn.
- Players whose Metro stations are all finished by a completed network of lines must continue to place their tiles elsewhere on the game board.

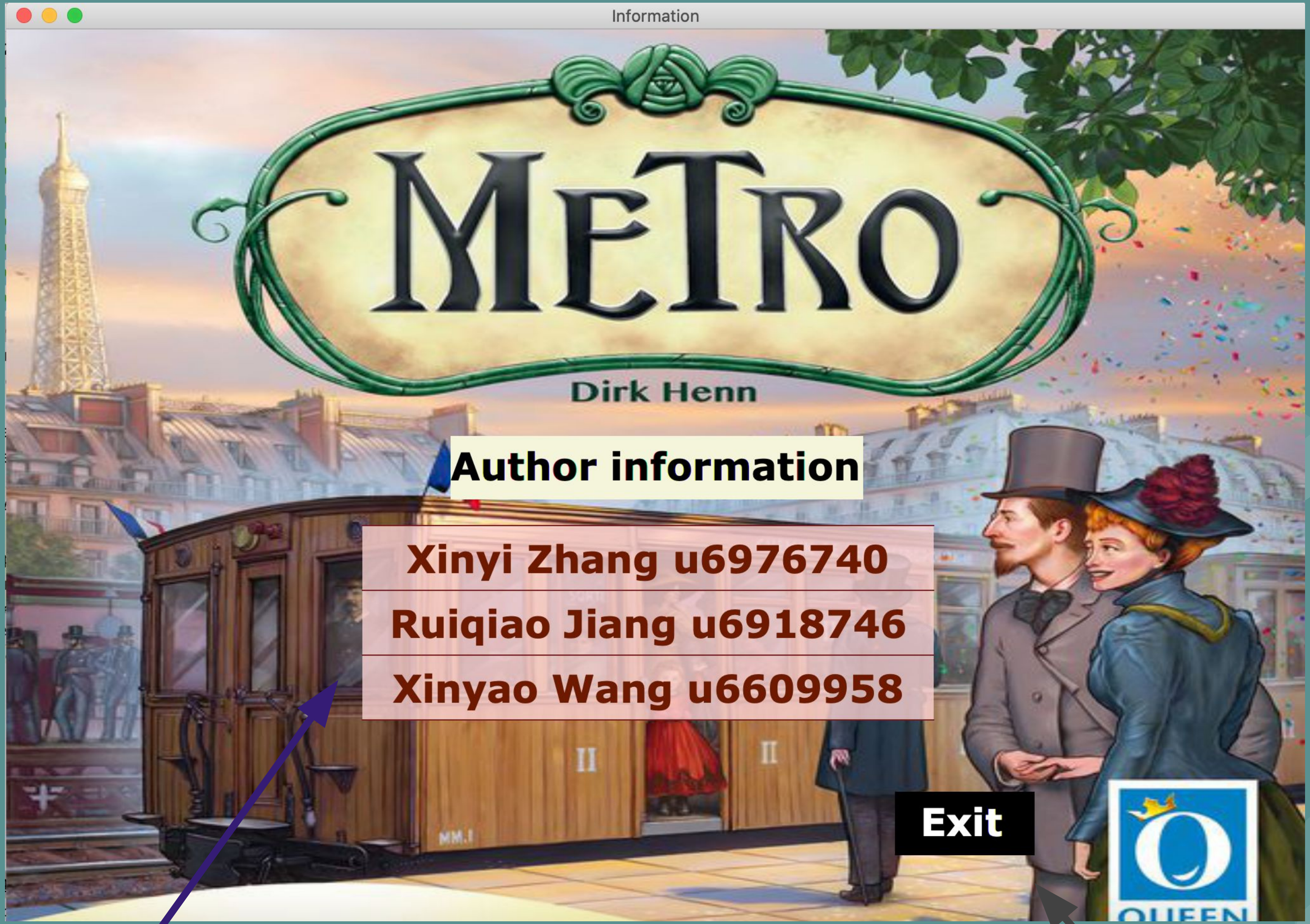
Back to Main viewer

Game Menu



Animation: the colour will change when the move the mouse and press the button

Information Page



Author information

Xinyi Zhang u6976740

Ruiqiao Jiang u6918746

Xinyao Wang u6609958

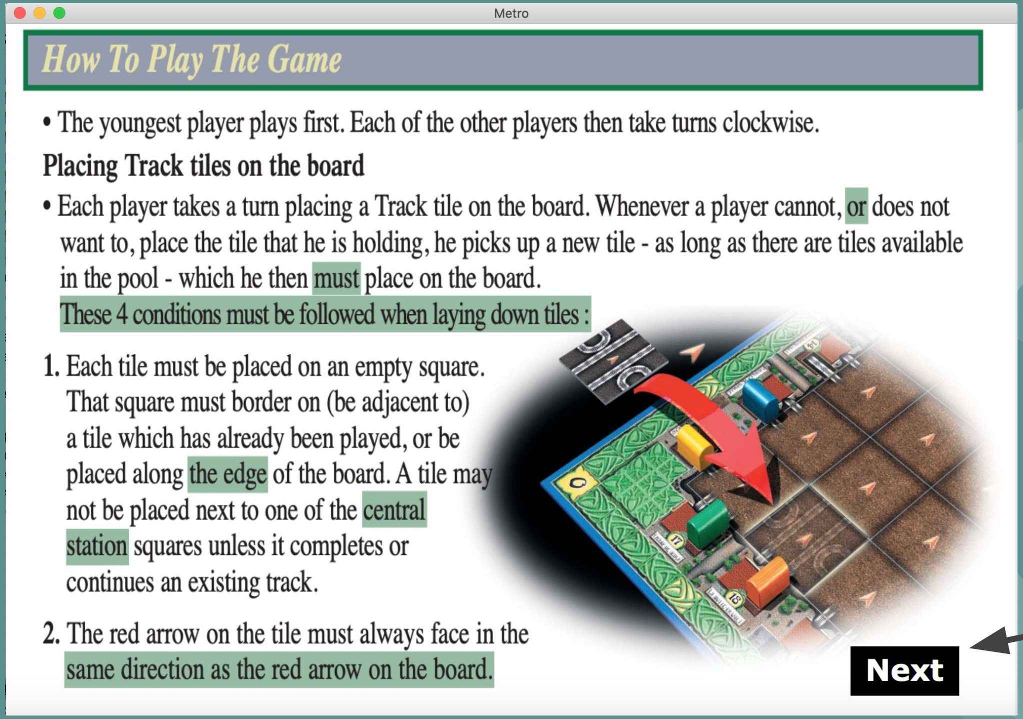
Exit

Indicated the author information

Press to back the main viewer

Tutorial

brief tutorial of the game source from the Metro Queen Games



How To Play The Game

• The youngest player plays first. Each of the other players then take turns clockwise.

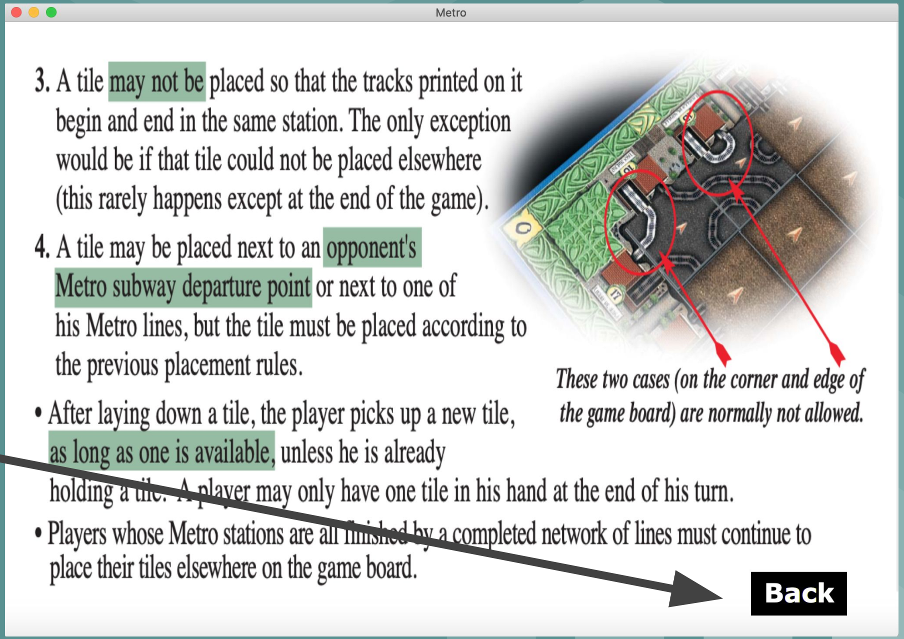
Placing Track tiles on the board

• Each player takes a turn placing a Track tile on the board. Whenever a player cannot, or does not want to, place the tile that he is holding, he picks up a new tile - as long as there are tiles available in the pool - which he then must place on the board.
These 4 conditions must be followed when laying down tiles :

1. Each tile must be placed on an empty square. That square must border on (be adjacent to) a tile which has already been played, or be placed along the edge of the board. A tile may not be placed next to one of the central station squares unless it completes or continues an existing track.

2. The red arrow on the tile must always face in the same direction as the red arrow on the board.

To the next page



3. A tile may not be placed so that the tracks printed on it begin and end in the same station. The only exception would be if that tile could not be placed elsewhere (this rarely happens except at the end of the game).

4. A tile may be placed next to an opponent's Metro subway departure point or next to one of his Metro lines, but the tile must be placed according to the previous placement rules.

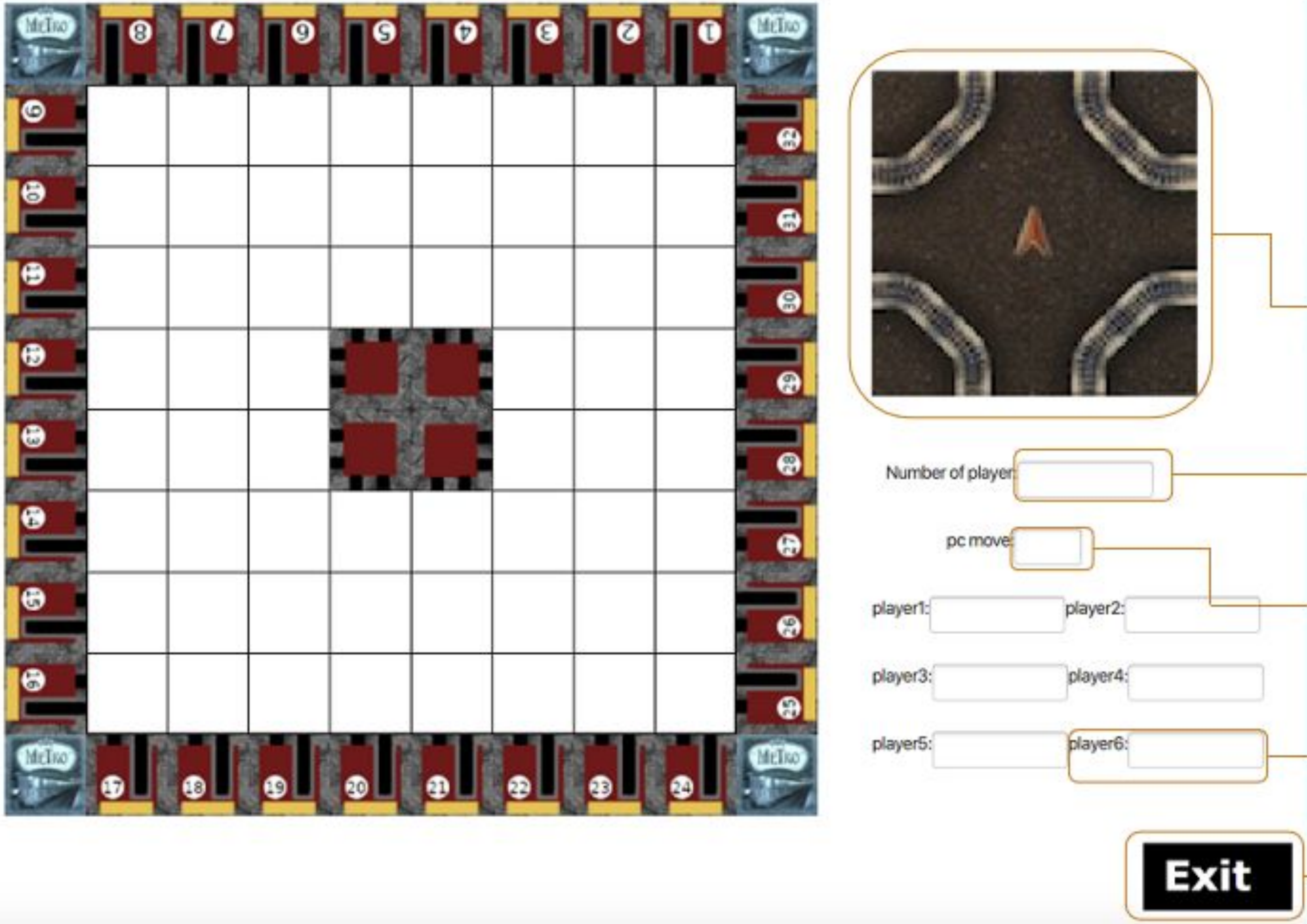
These two cases (on the corner and edge of the game board) are normally not allowed.

• After laying down a tile, the player picks up a new tile, as long as one is available, unless he is already holding a tile. A player may only have one tile in his hand at the end of his turn.

• Players whose Metro stations are all finished by a completed network of lines must continue to place their tiles elsewhere on the game board.

Back to Main viewer

Viewer



One tile randomly selected by the computer, When we clicked on the tile picture ,the size of the tile will become the same as the gird on the board, and it can be dragged onto the board.

Input the number of players here(1-6)

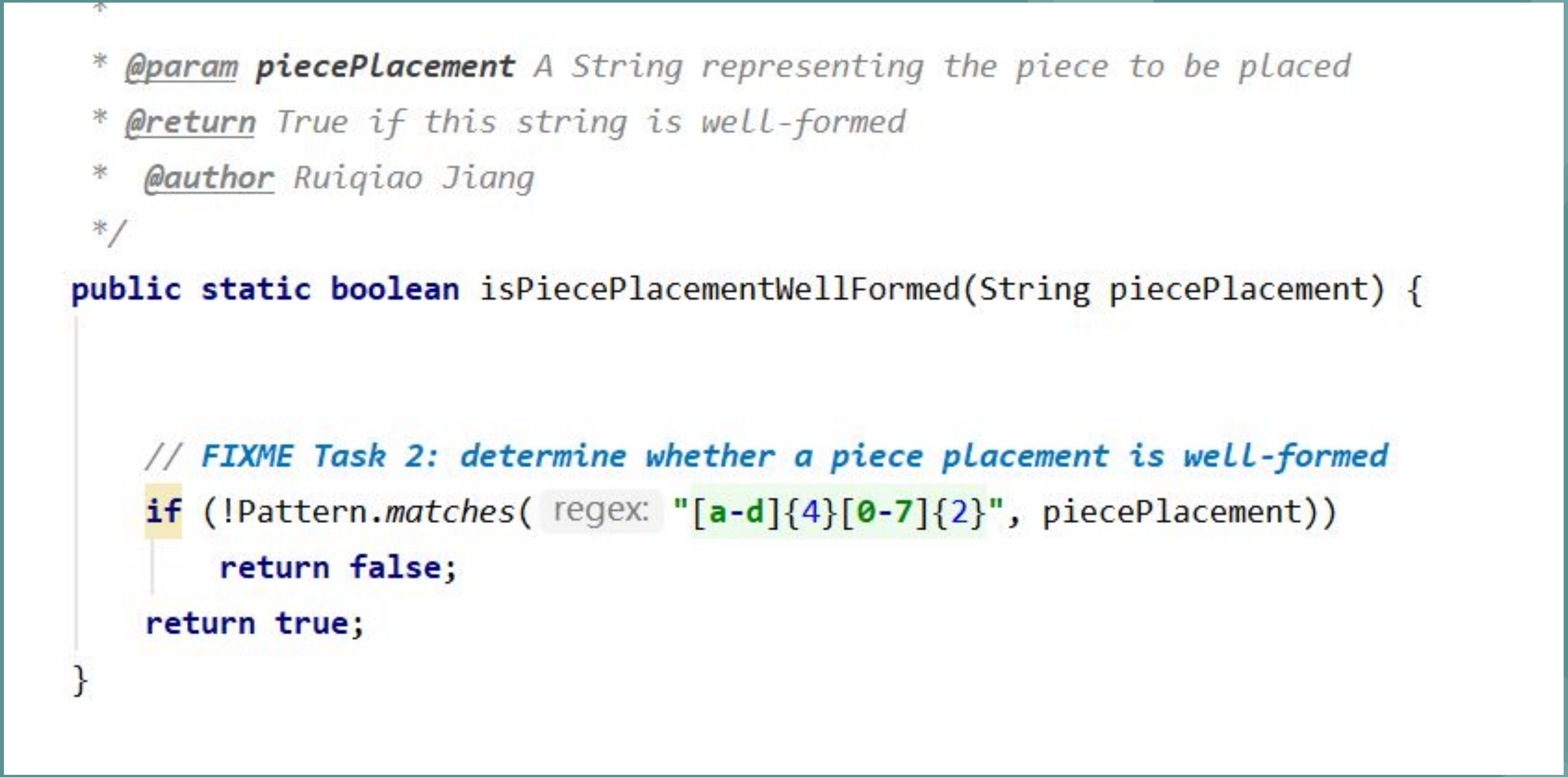
Input the number of tiles you want the computer to automatically generate on the board after you place a tile correctly

Display the score of each player, automatically update after dragging the tile to the correct position.

Return to main interface

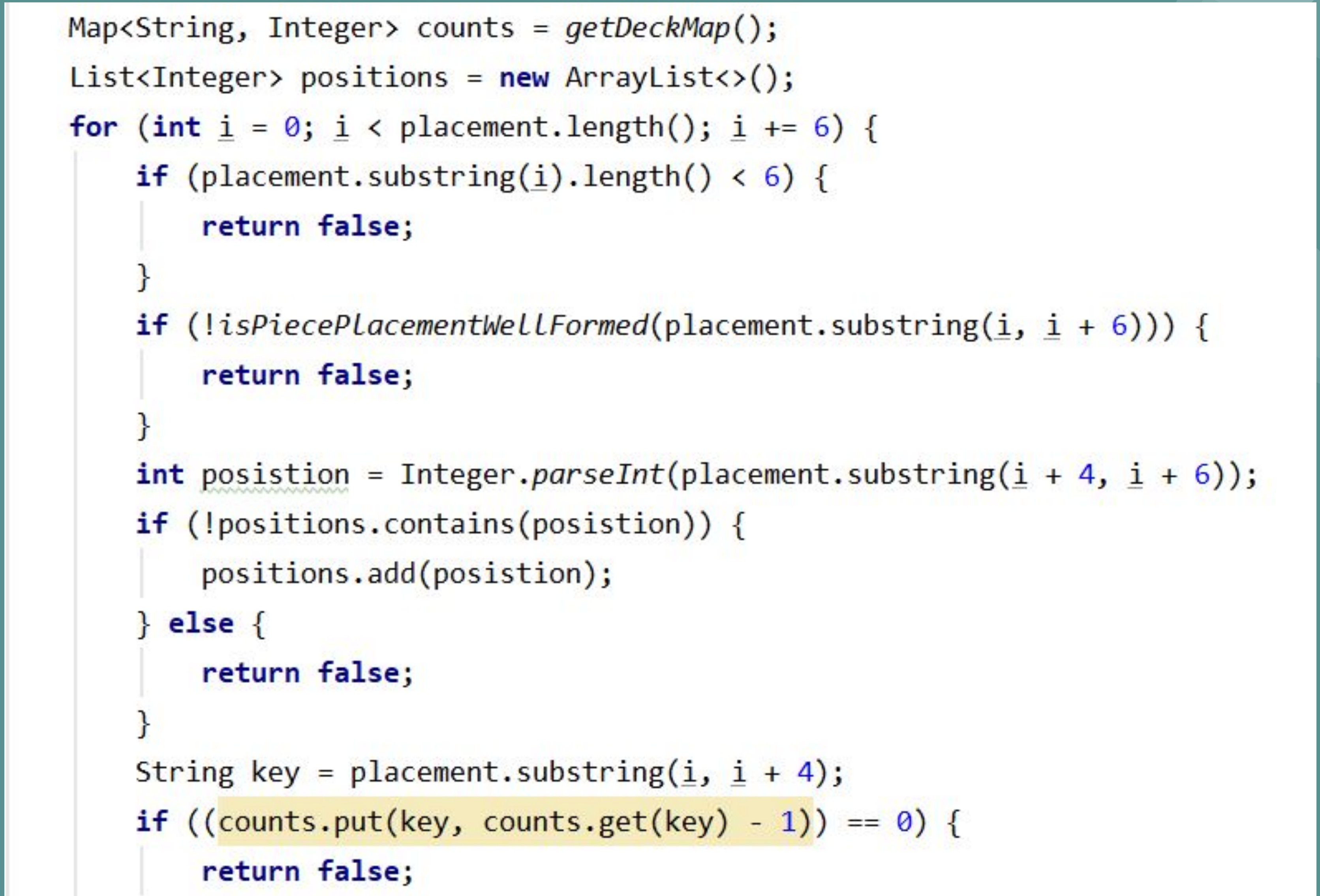
Task 2

determine whether a piece placement is well-formed



```java
 * @param piecePlacement A String representing the piece to be placed
 * @return True if this string is well-formed
 *  @author Ruiqiao Jiang
 */
public static boolean isPiecePlacementWellFormed(String piecePlacement) {


    // FIXME Task 2: determine whether a piece placement is well-formed
    if (!Pattern.matches( regex: "[a-d]{4}[0-7]{2}", piecePlacement))
        return false;
    return true;
}
```

Task 3

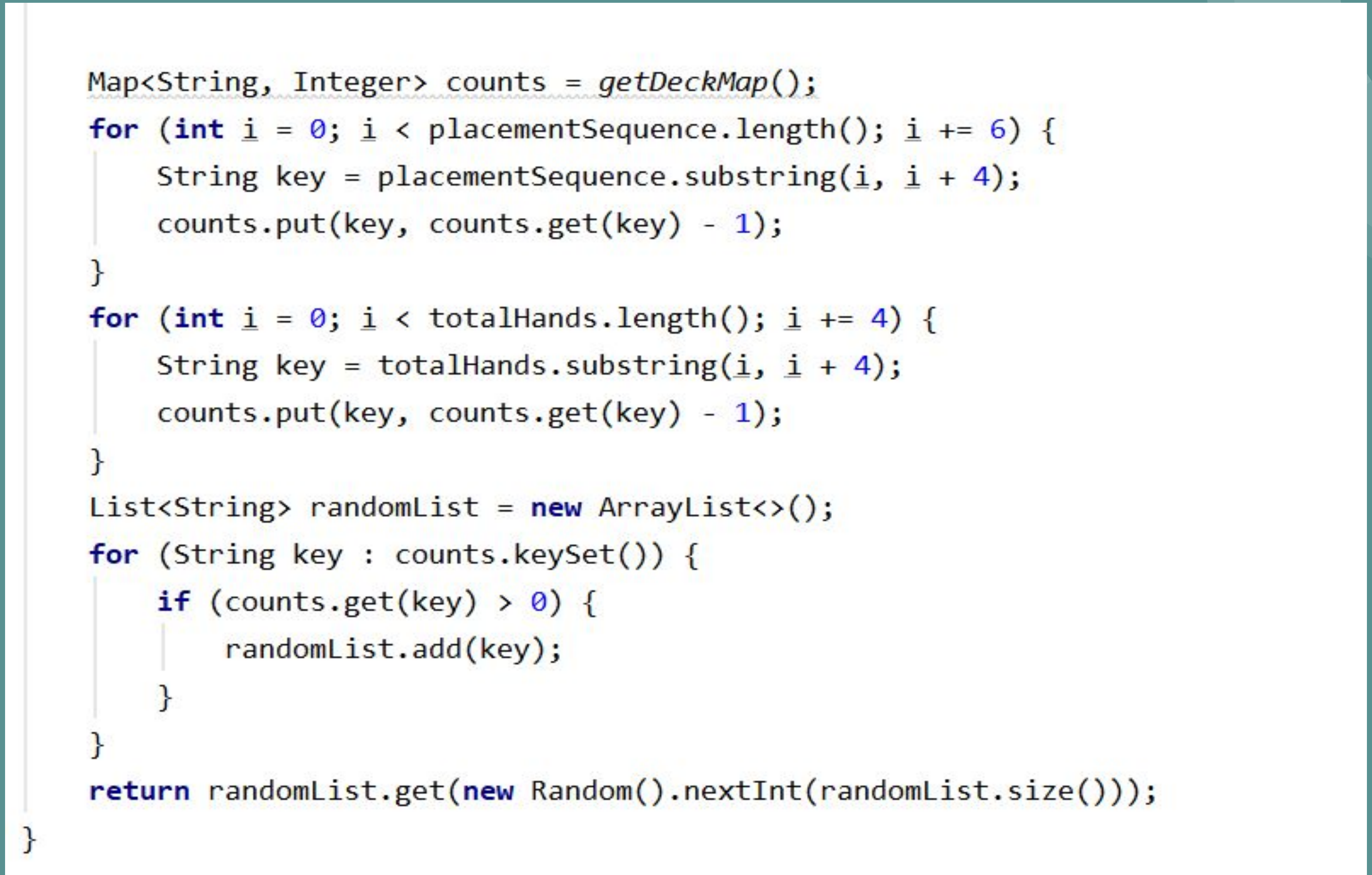


```
Map<String, Integer> counts = getDeckMap();
List<Integer> positions = new ArrayList<>();
for (int i = 0; i < placement.length(); i += 6) {
    if (placement.substring(i).length() < 6) {
        return false;
    }
    if (!isPiecePlacementWellFormed(placement.substring(i, i + 6))) {
        return false;
    }
    int posistion = Integer.parseInt(placement.substring(i + 4, i + 6));
    if (!positions.contains(posistion)) {
        positions.add(posistion);
    } else {
        return false;
    }
    String key = placement.substring(i, i + 4);
    if ((counts.put(key, counts.get(key) - 1)) == 0) {
        return false;
```

determine whether a placement sequence is well-formed

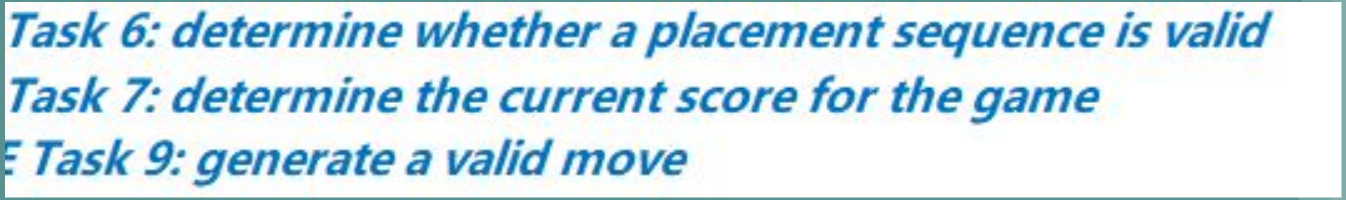
Task 5

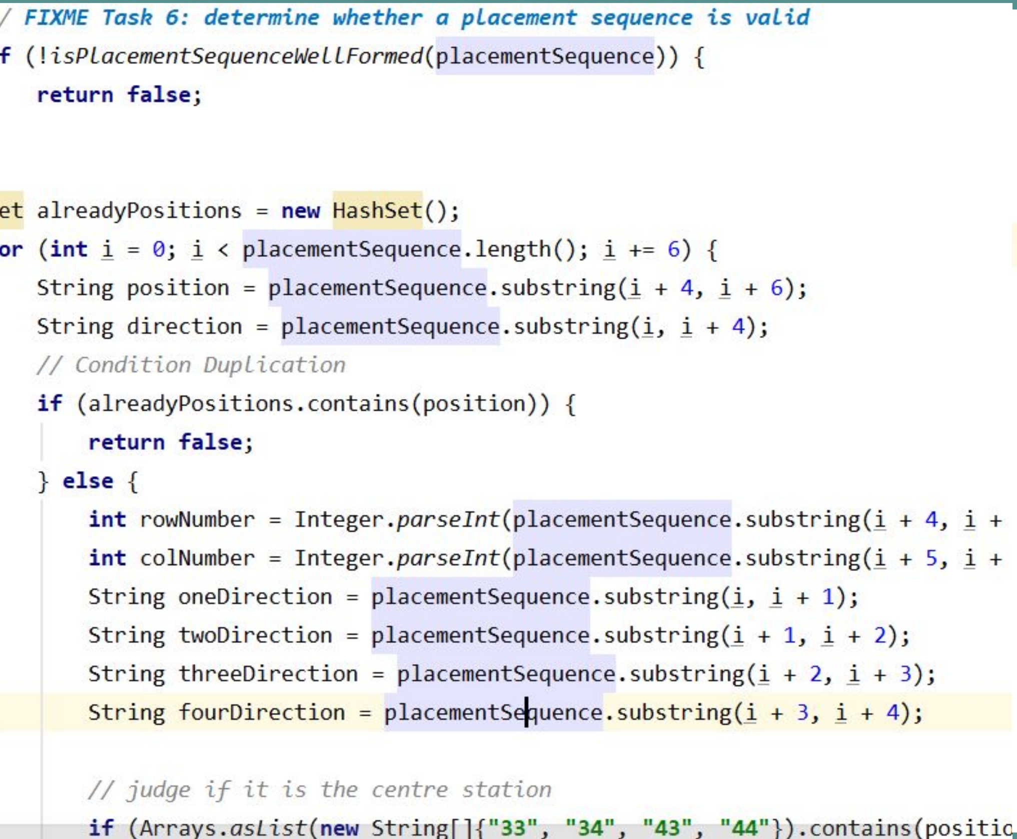
draw a random tile from deck

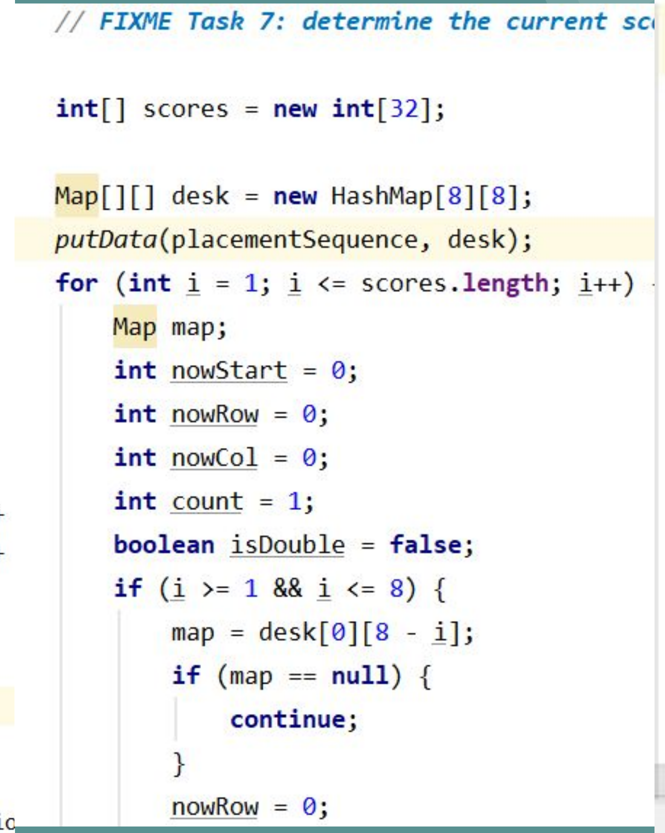


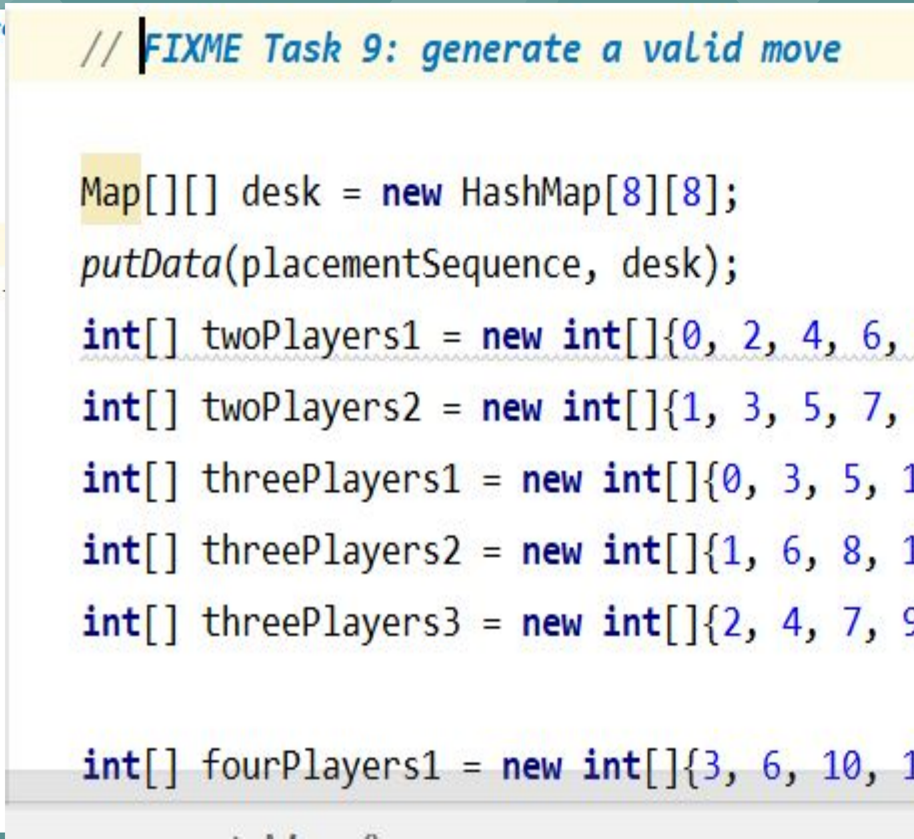
```
    Map<String, Integer> counts = getDeckMap();
    for (int i = 0; i < placementSequence.length(); i += 6) {
        String key = placementSequence.substring(i, i + 4);
        counts.put(key, counts.get(key) - 1);
    }
    for (int i = 0; i < totalHands.length(); i += 4) {
        String key = totalHands.substring(i, i + 4);
        counts.put(key, counts.get(key) - 1);
    }
    List<String> randomList = new ArrayList<>();
    for (String key : counts.keySet()) {
        if (counts.get(key) > 0) {
            randomList.add(key);
        }
    }
    return randomList.get(new Random().nextInt(randomList.size()));
}
```

Task6-9


Task 6: determine whether a placement sequence is valid
Task 7: determine the current score for the game
E Task 9: generate a valid move


```
/ FIXME Task 6: determine whether a placement sequence is valid
f (!isPlacementSequenceWellFormed(placementSequence)) {
   return false;


et alreadyPositions = new HashSet();
or (int i = 0; i < placementSequence.length(); i += 6) {
   String position = placementSequence.substring(i + 4, i + 6);
   String direction = placementSequence.substring(i, i + 4);
   // Condition Duplication
   if (alreadyPositions.contains(position)) {
       return false;
   } else {
       int rowNumber = Integer.parseInt(placementSequence.substring(i + 4, i +
       int colNumber = Integer.parseInt(placementSequence.substring(i + 5, i +
       String oneDirection = placementSequence.substring(i, i + 1);
       String twoDirection = placementSequence.substring(i + 1, i + 2);
       String threeDirection = placementSequence.substring(i + 2, i + 3);
       String fourDirection = placementSequence.substring(i + 3, i + 4);

       // judge if it is the centre station
       if (Arrays.asList(new String[]{"33", "34", "43", "44"}).contains(positic
```


```
// FIXME Task 7: determine the current sc

int[] scores = new int[32];

Map[][] desk = new HashMap[8][8];
putData(placementSequence, desk);
for (int i = 1; i <= scores.length; i++) 
    Map map;
    int nowStart = 0;
    int nowRow = 0;
    int nowCol = 0;
    int count = 1;
    boolean isDouble = false;
    if (i >= 1 && i <= 8) {
        map = desk[0][8 - i];
        if (map == null) {
            continue;
        }
        nowRow = 0;
```


```
// FIXME Task 9: generate a valid move


Map[][] desk = new HashMap[8][8];
putData(placementSequence, desk);
int[] twoPlayers1 = new int[]{0, 2, 4, 6,
int[] twoPlayers2 = new int[]{1, 3, 5, 7,
int[] threePlayers1 = new int[]{0, 3, 5, 1
int[] threePlayers2 = new int[]{1, 6, 8, 1
int[] threePlayers3 = new int[]{2, 4, 7, 9


int[] fourPlayers1 = new int[]{3, 6, 10, 1
```

Thanks