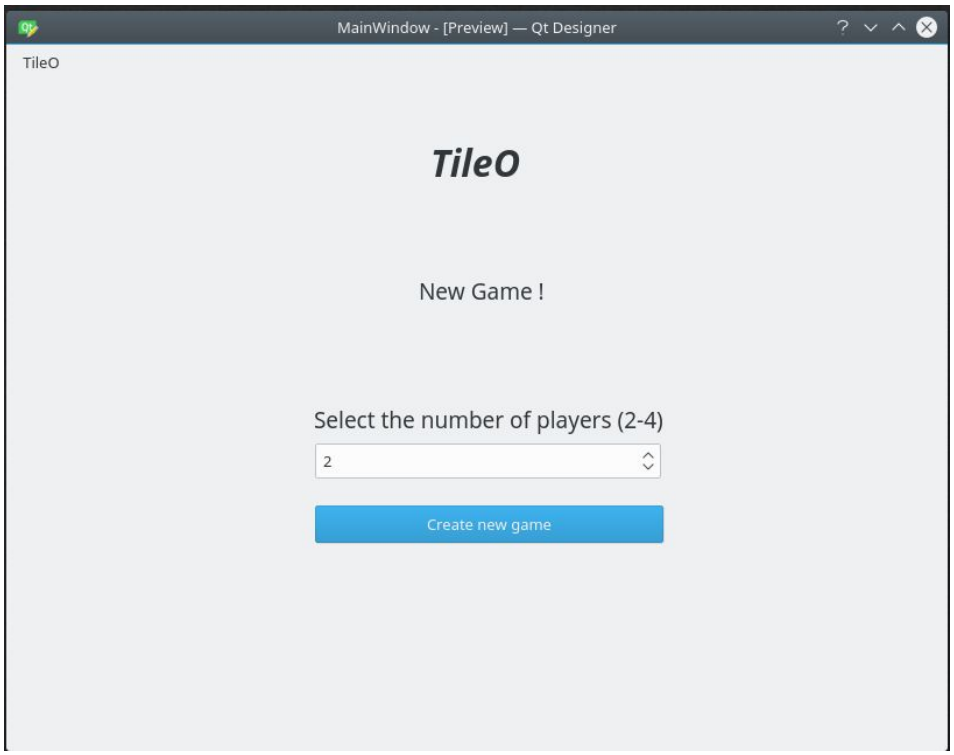
Group 14: Vincent D., Anthony L., Sam C., Younes B., James T., Gabriel T.

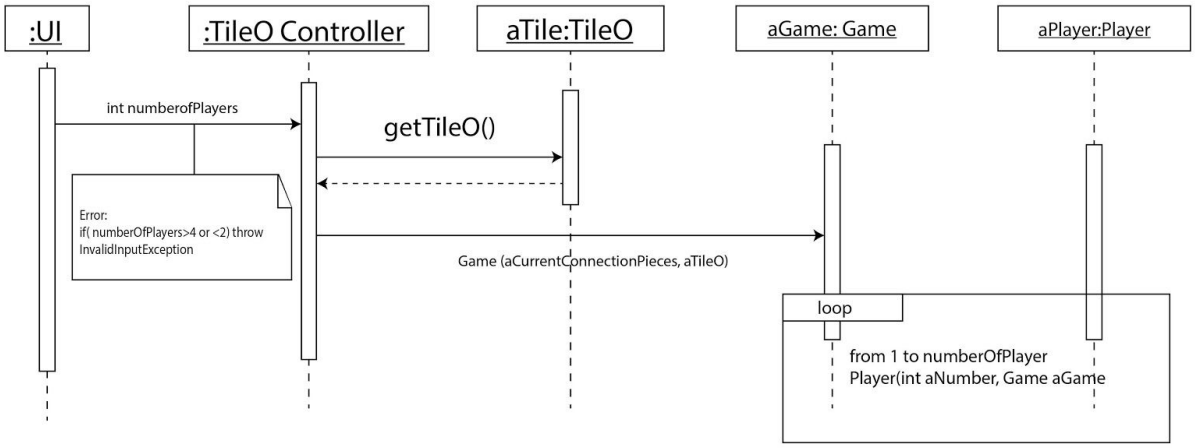
ECSE 223 Deliverable 2

Design Mode:

1. Create a game with a number of players: Younes
public void createGame(int numberOfPlayers)

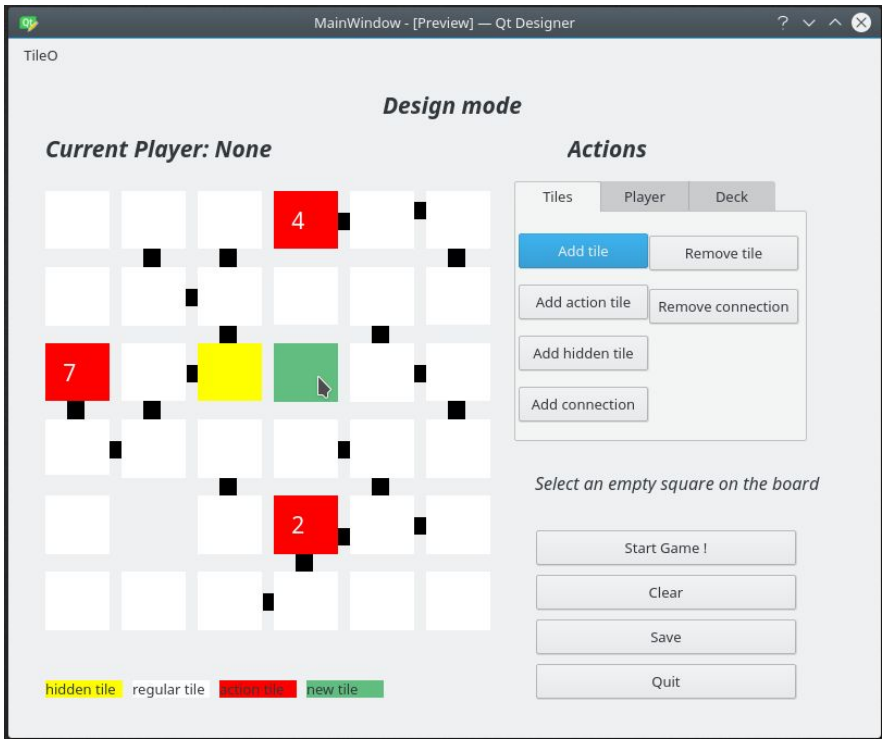


Sequence Diagram for Creating a Game with Number of Players

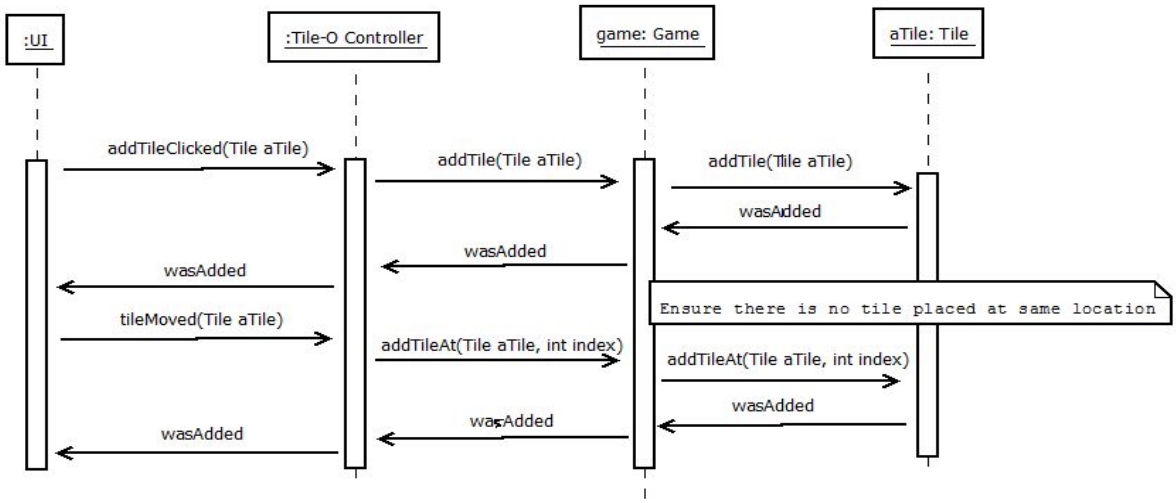


2. Place a tile on the game board: Anthony

User Interface:



Sequence Diagram:

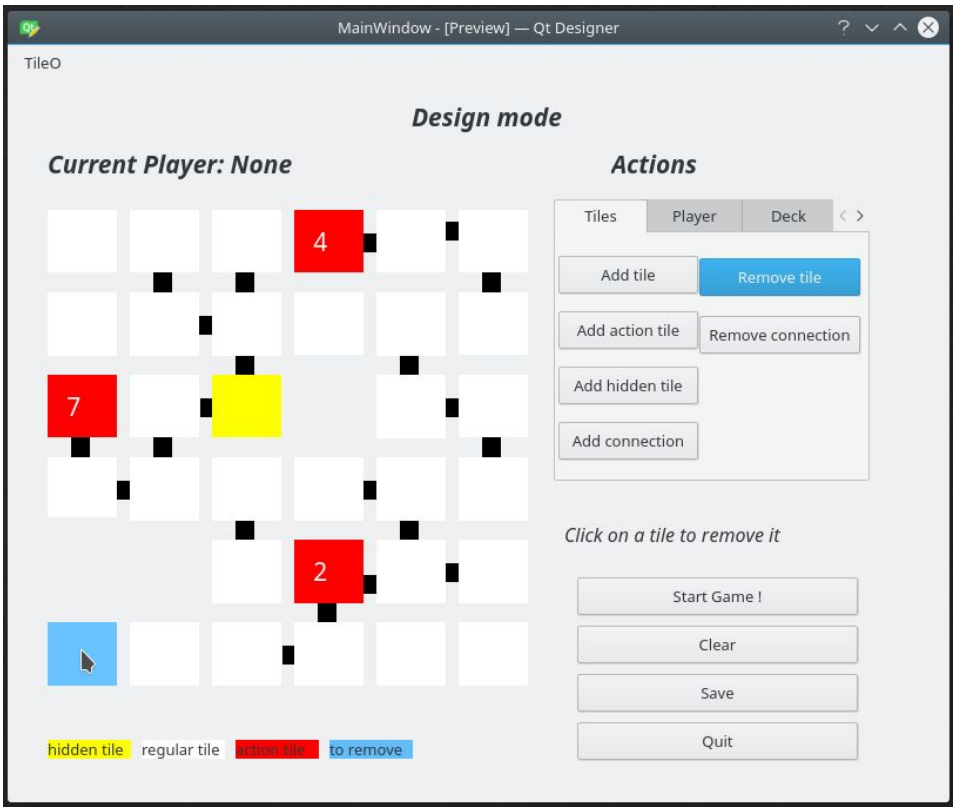


Controller Interface:

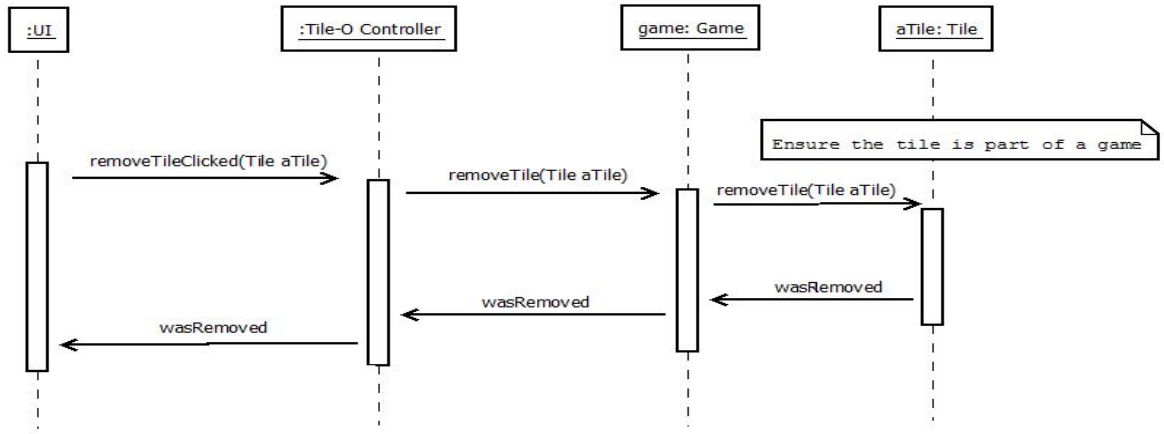
- public void addTileCllicked(Tile aTile)
- public void tileMoved(Tile aTile)

3. Remove a tile from the game board: Anthony

User Interface:



Sequence Diagram:

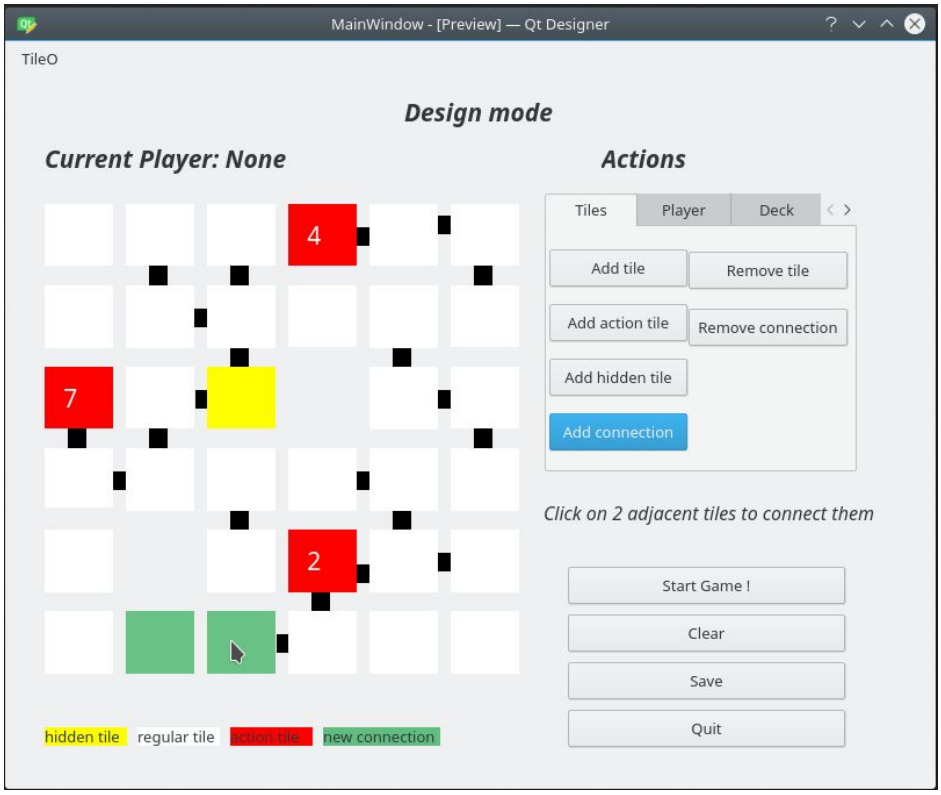


Controller Interface:

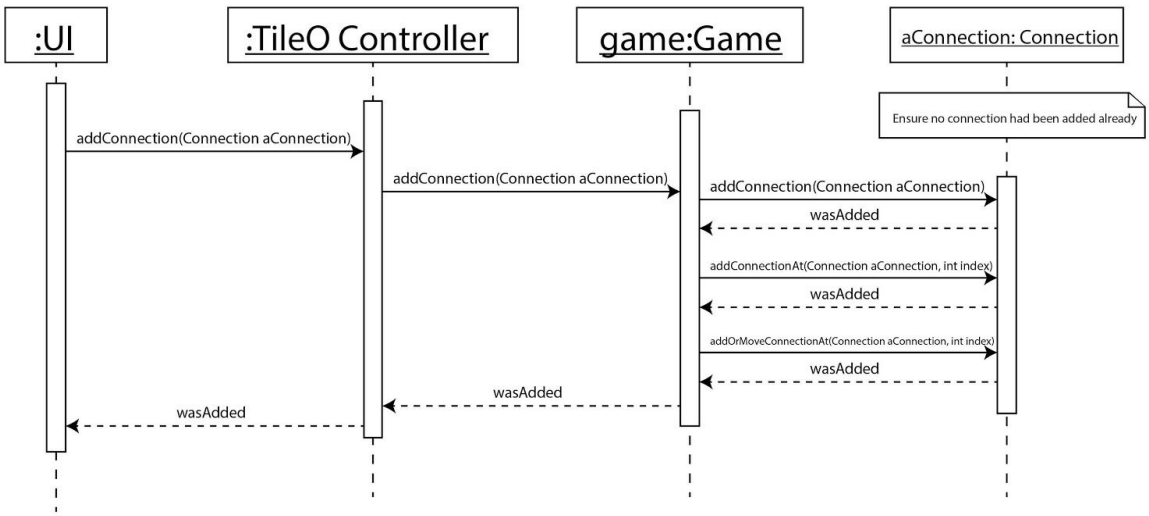
- public void removedTileClicked(Tile aTile)

4. Connect two tiles with a connection piece: James

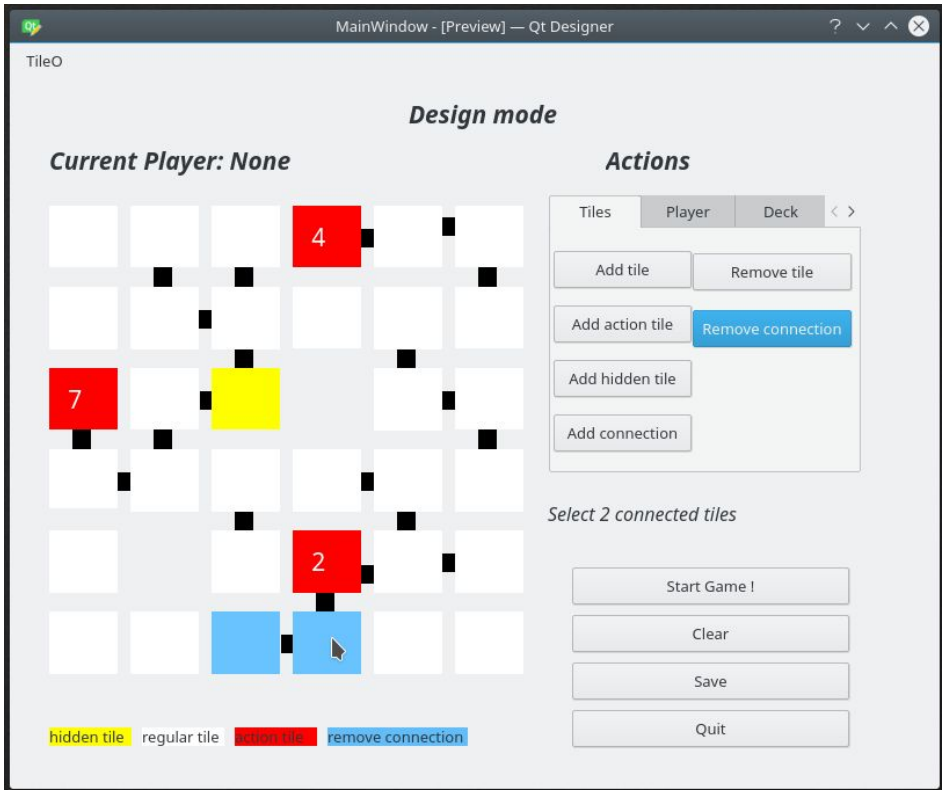
Controller: **public void connectTiles(Connection aConnection)**



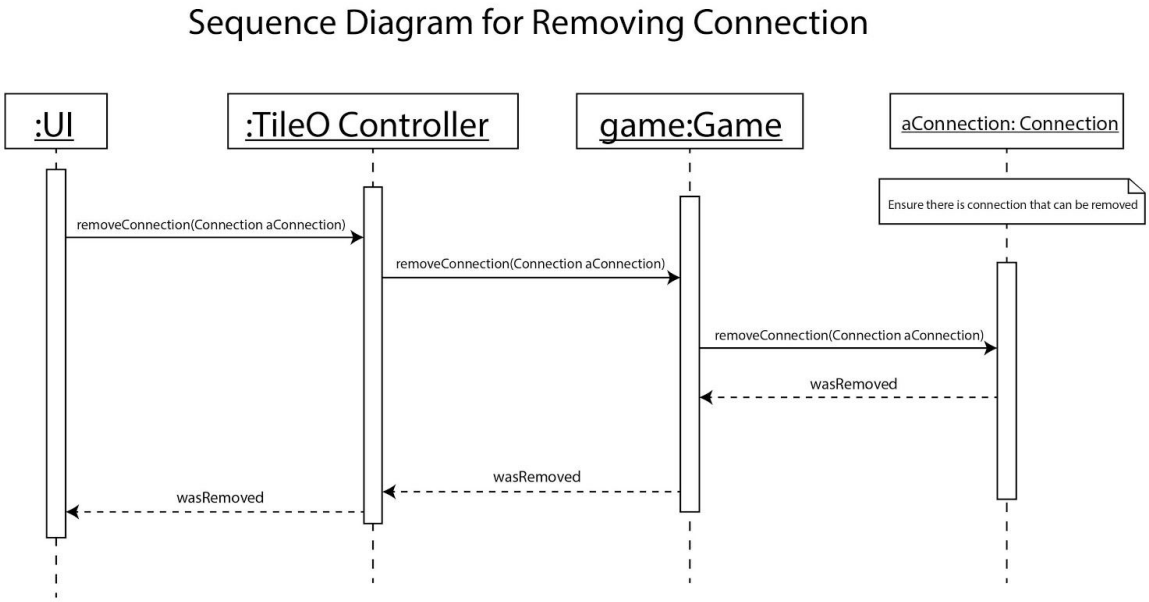
Sequence Diagram for Adding Connection



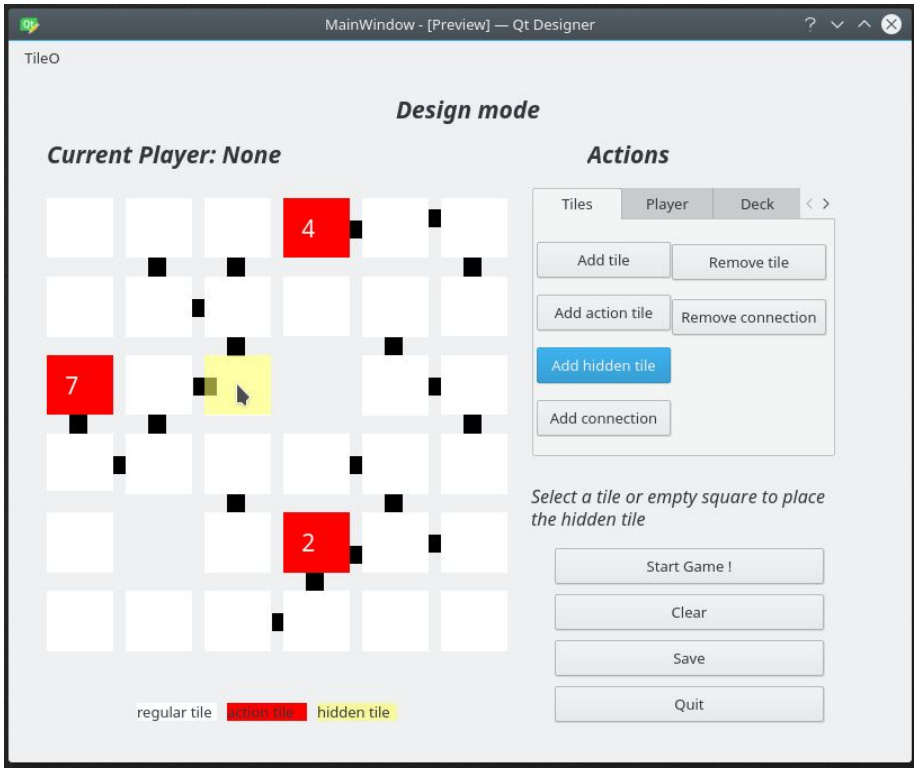
5. Remove connection between two tiles: James



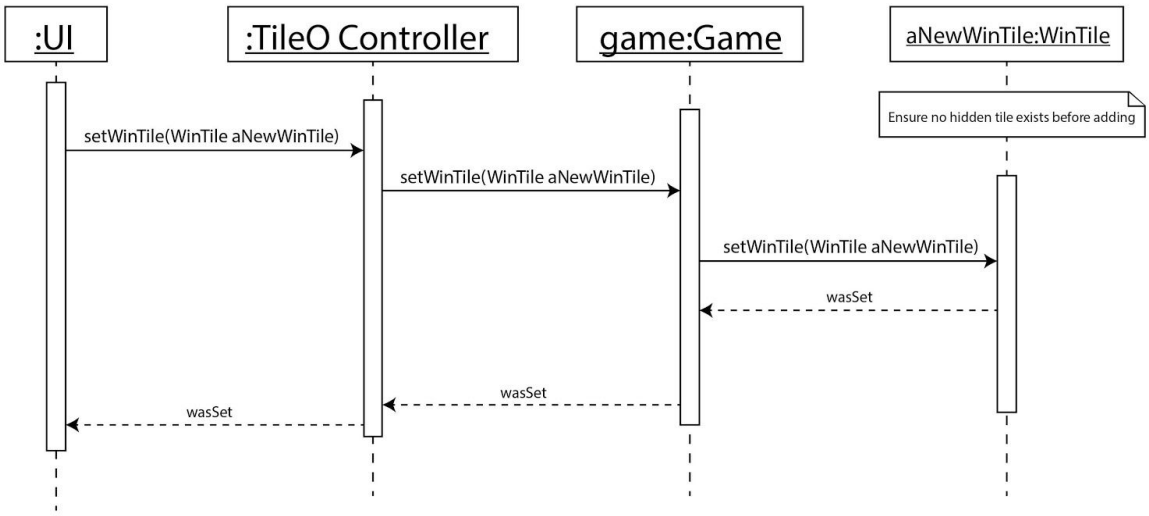
Controller: **public void removeTiles(Connection aConnection)**



6. Identify the hidden tile: James

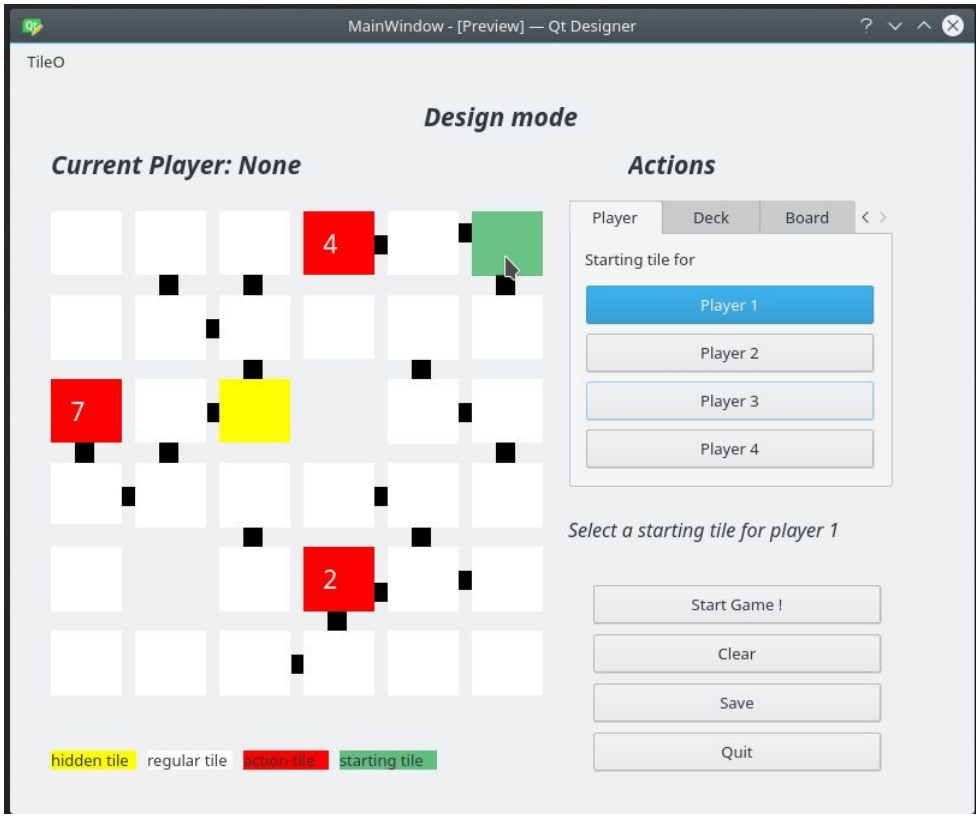


Sequence Diagram for Identifying the Hidden Tile

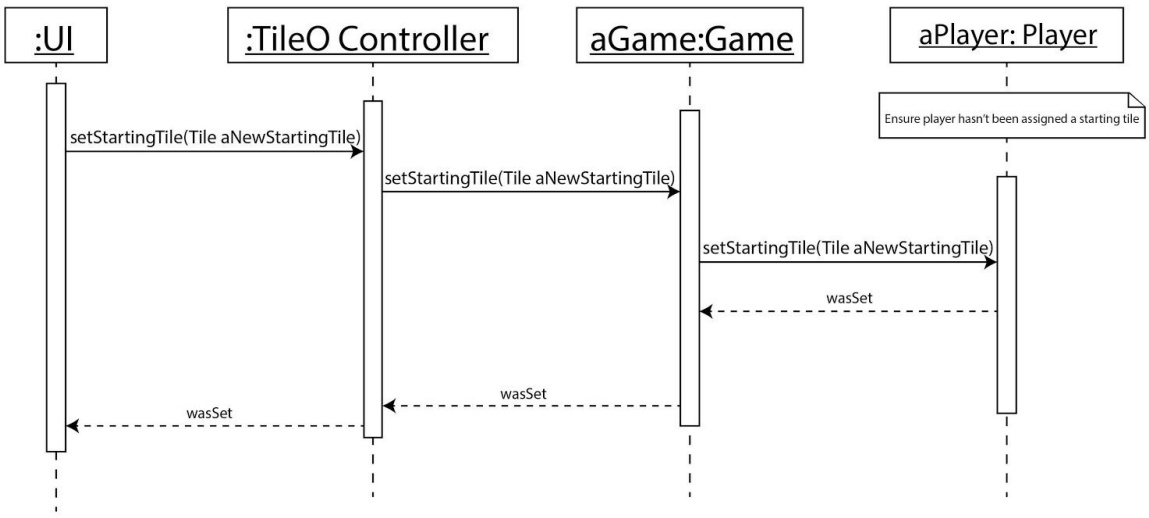


public void setHiddenTiles(WinTile aNewWinTile)

7. Identify the starting tile of a player: Younes

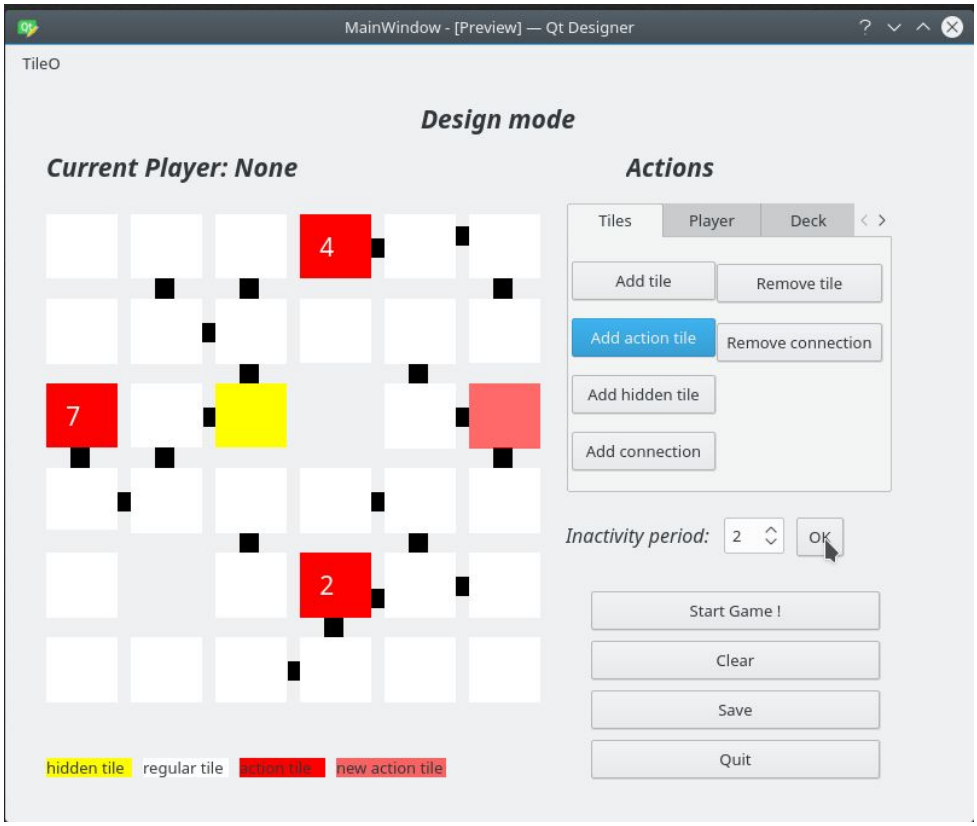


Sequence Diagram for Set Starting Tile of Player

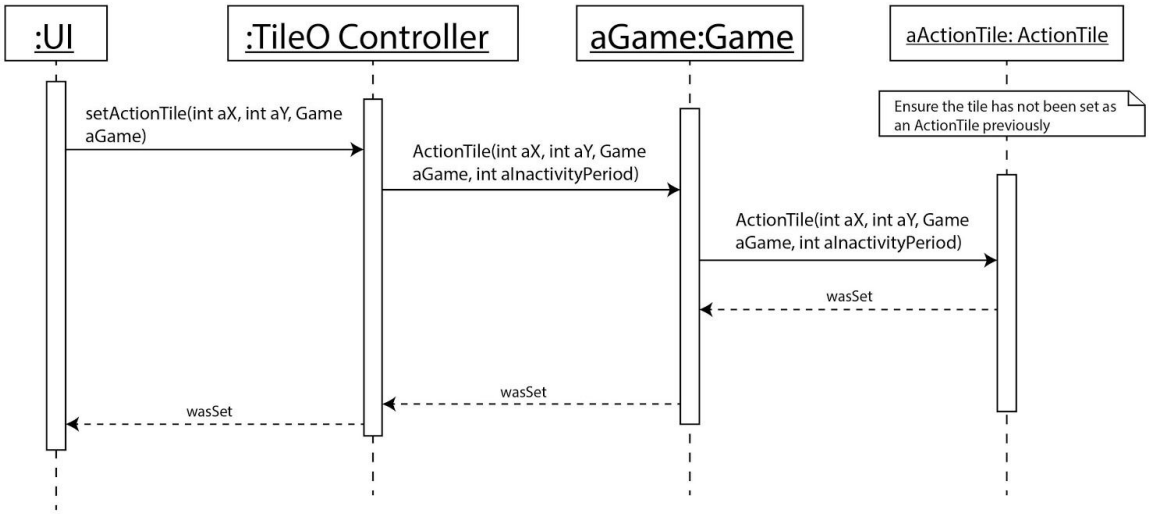


public void setStartingTile(Tile aNewStartingTile)

8. Identify an action tile: Younes

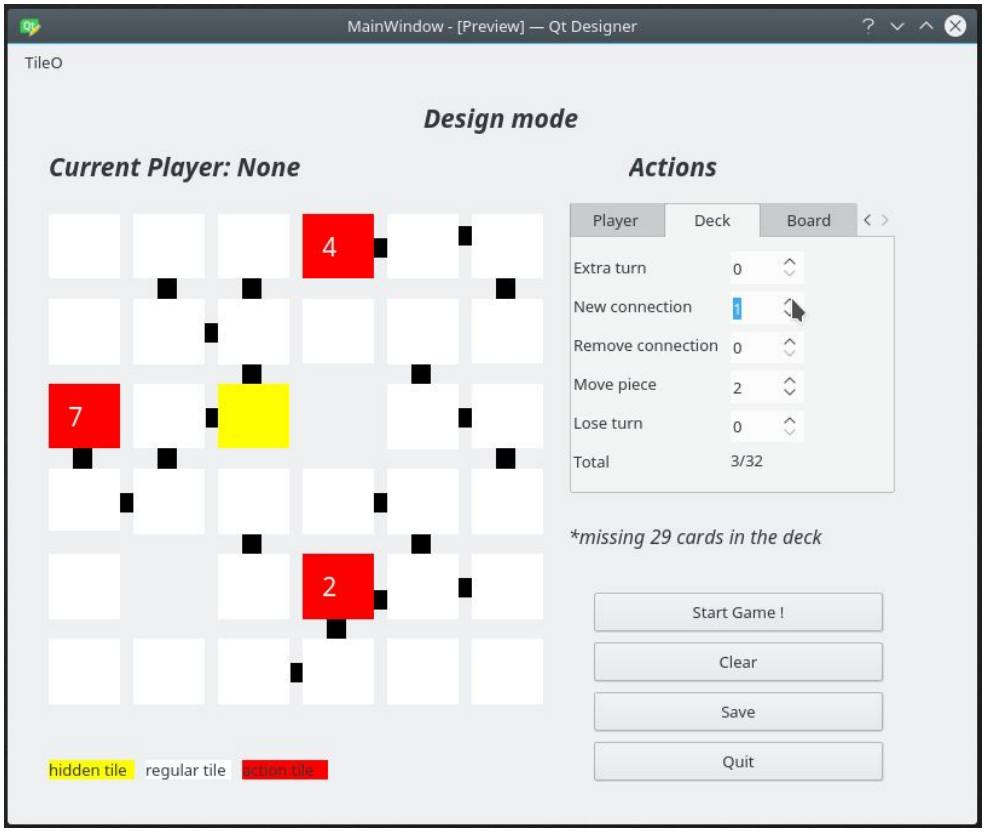


Sequence Diagram for Identifying Action Tiles



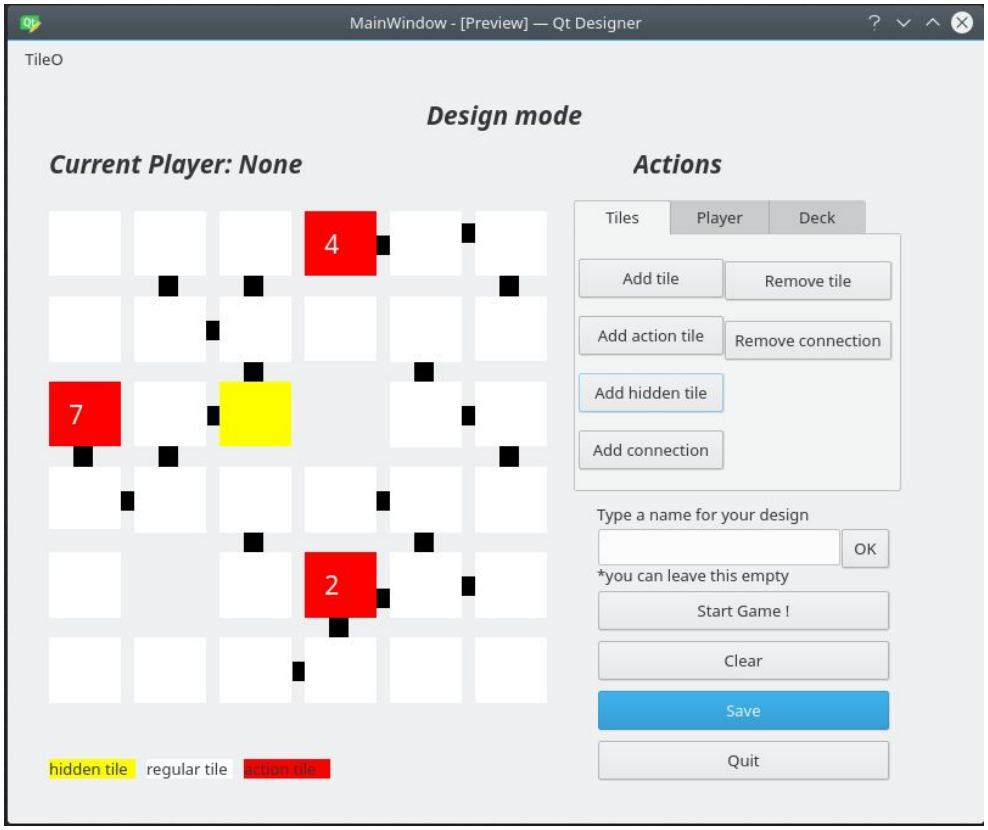
Public void setActionTile(int aX, int aY, Game aGame)

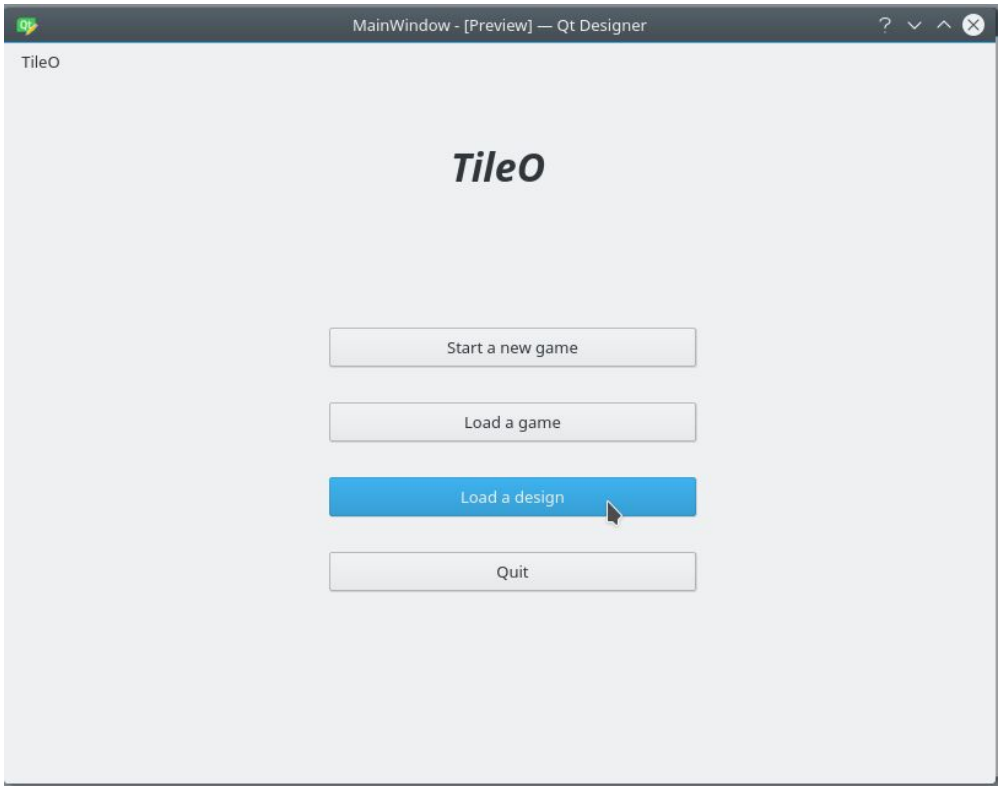
9. Select 32 cards from predefined choices: Gabriel

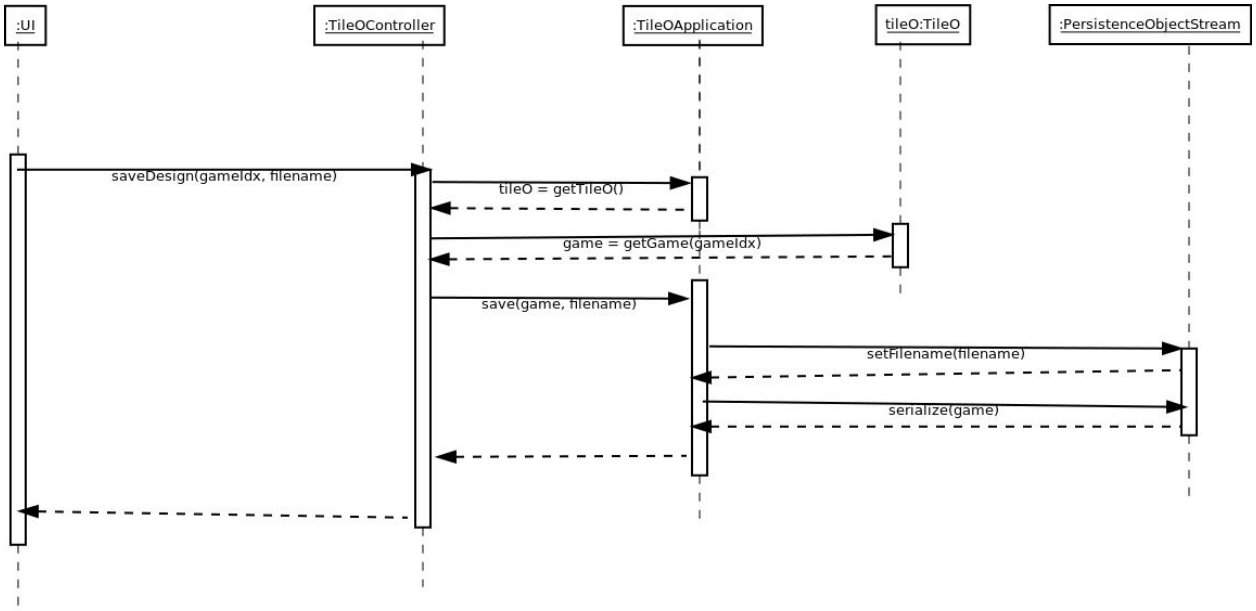


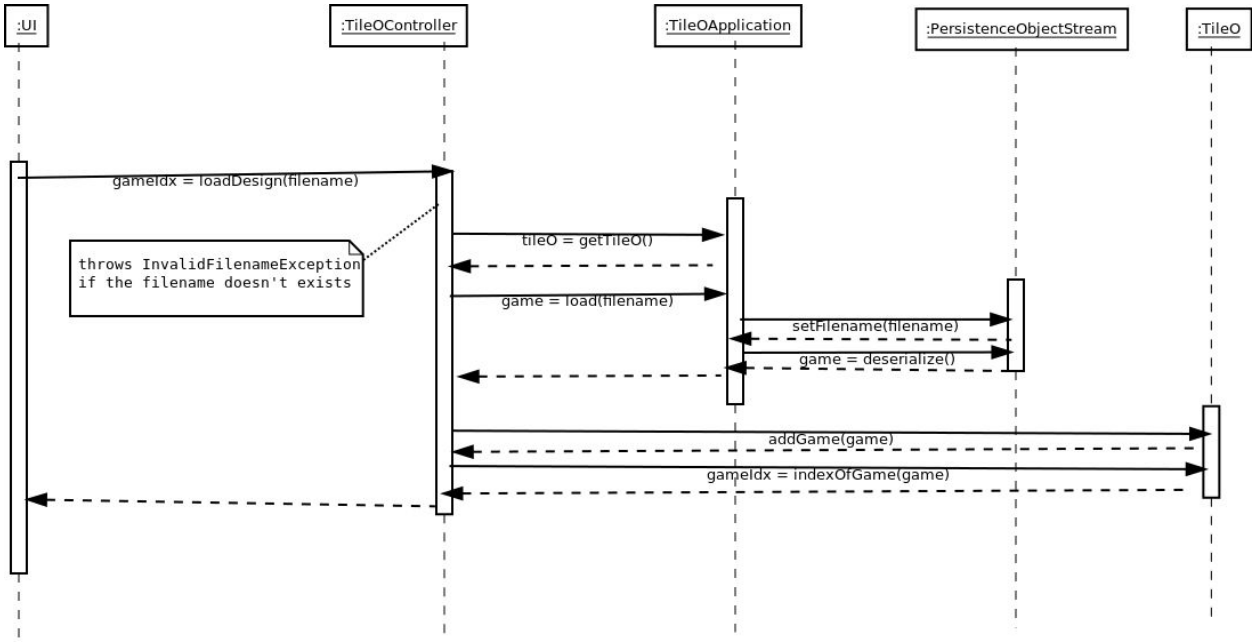
Public void createDeck(nExtra, nNewConn, nRemovConn, nMove, nLose)

10. Save/Load design: Vincent







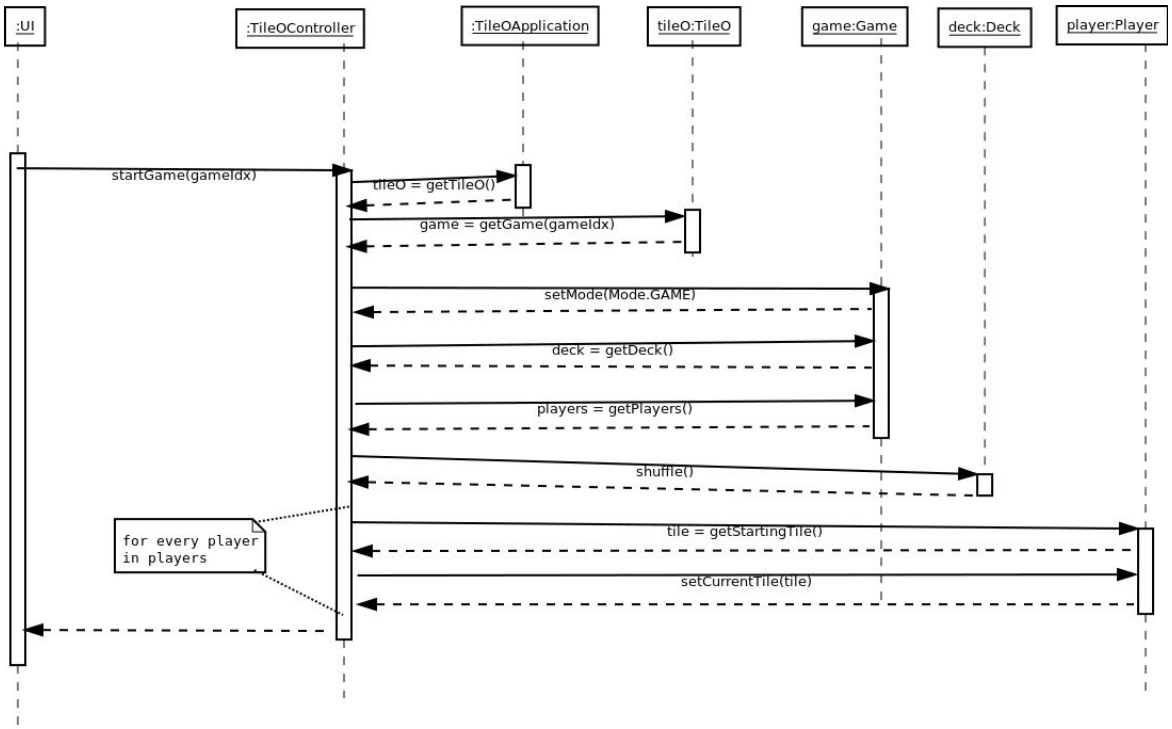


Controller Interface:

- public void saveDesign(int gameIdx, String filename)
- public int loadDesign(String filename)

Play Mode:

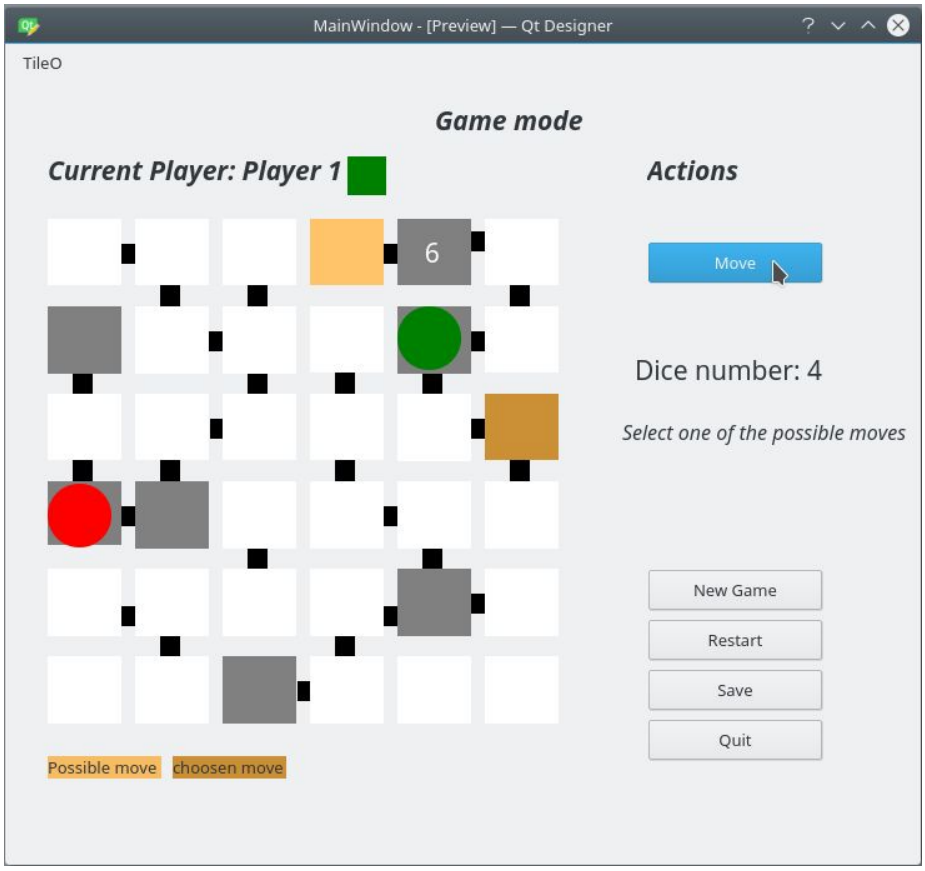
1. **Start game: Vincent**

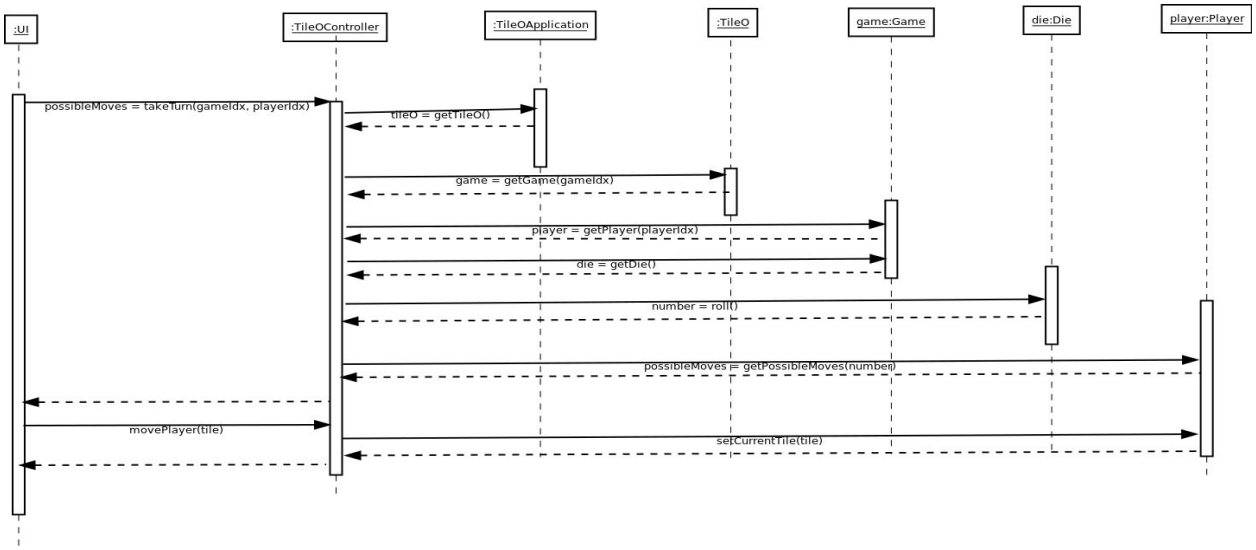


Controller Interface:

- public void startGame(int gameIdx)

2. Take a turn (roll the die, move to new position): Gabriel



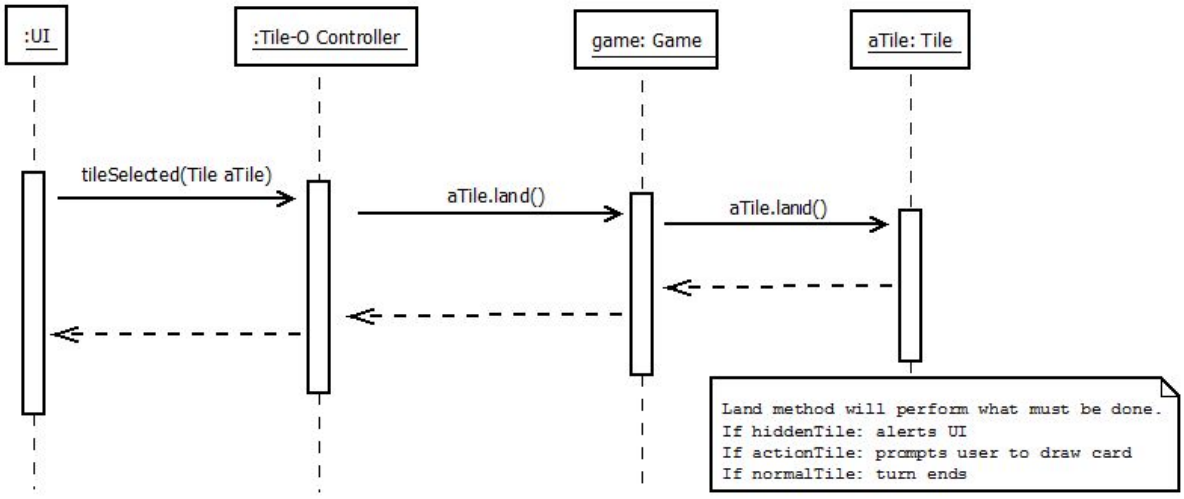


Controller Interface:

- Public List<Tile> takeTurn(gameIdx, playerIdx)
- Public void movePlayer(tile)
- **3. Land on a tile (basic behavior for hidden, regular, and action tiles): Anthony**

User Interface (Regular, Action, Hidden):

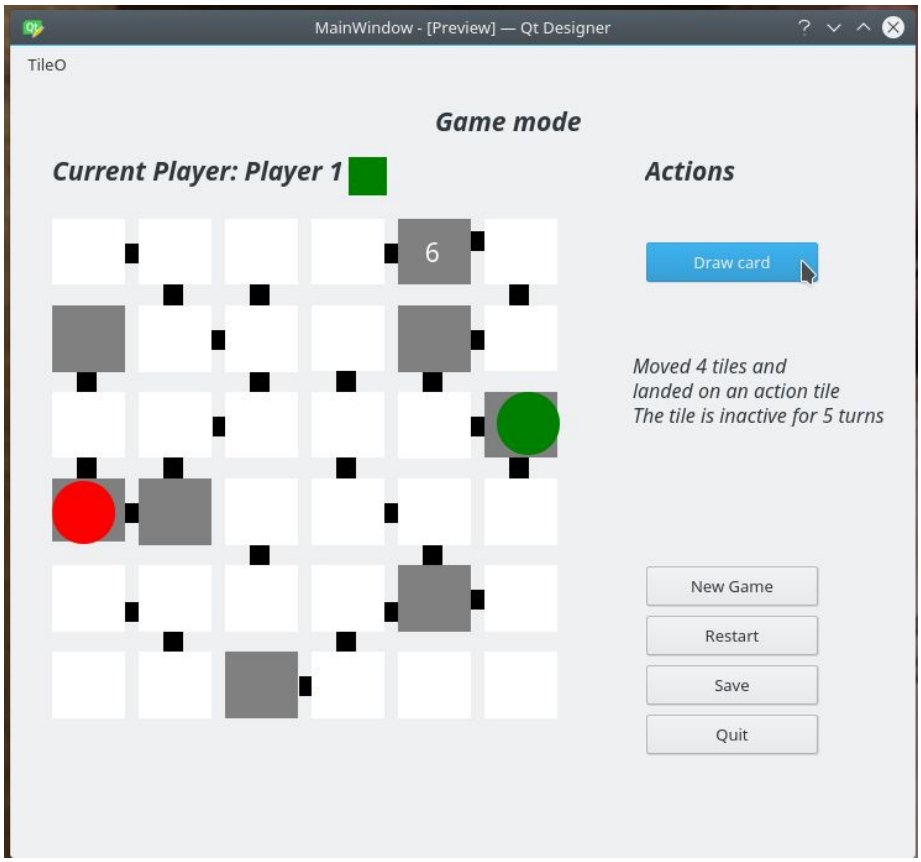
Sequence Diagram:

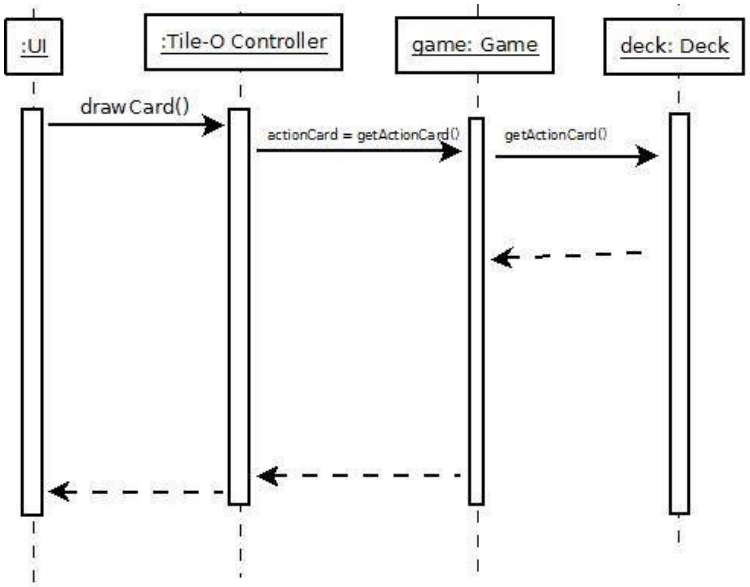


Controller Interface:

- public void tileSelected(Tile aTile)

4. Take the first card from the deck of cards: Gabriel



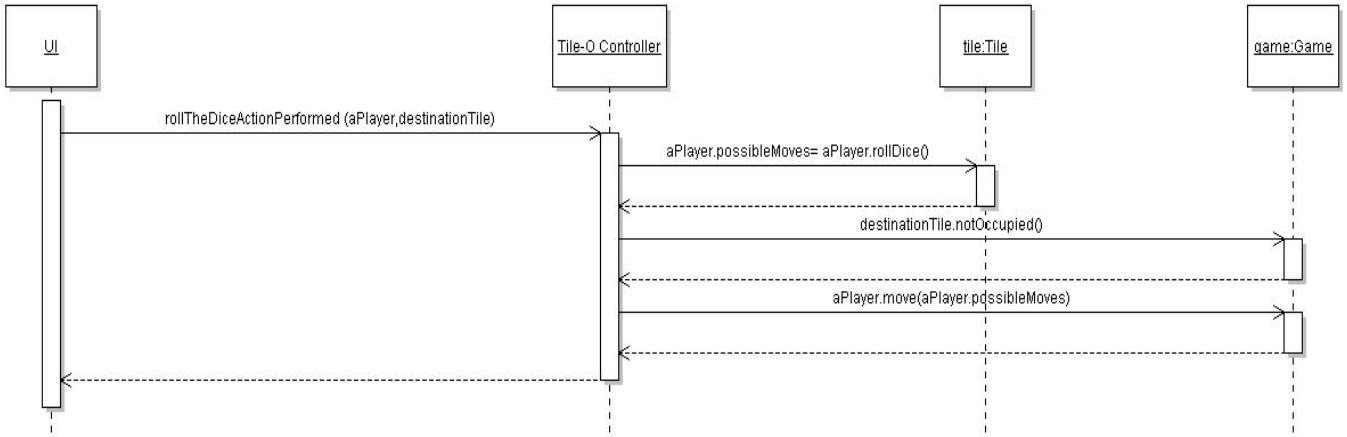


Public void drawCard()

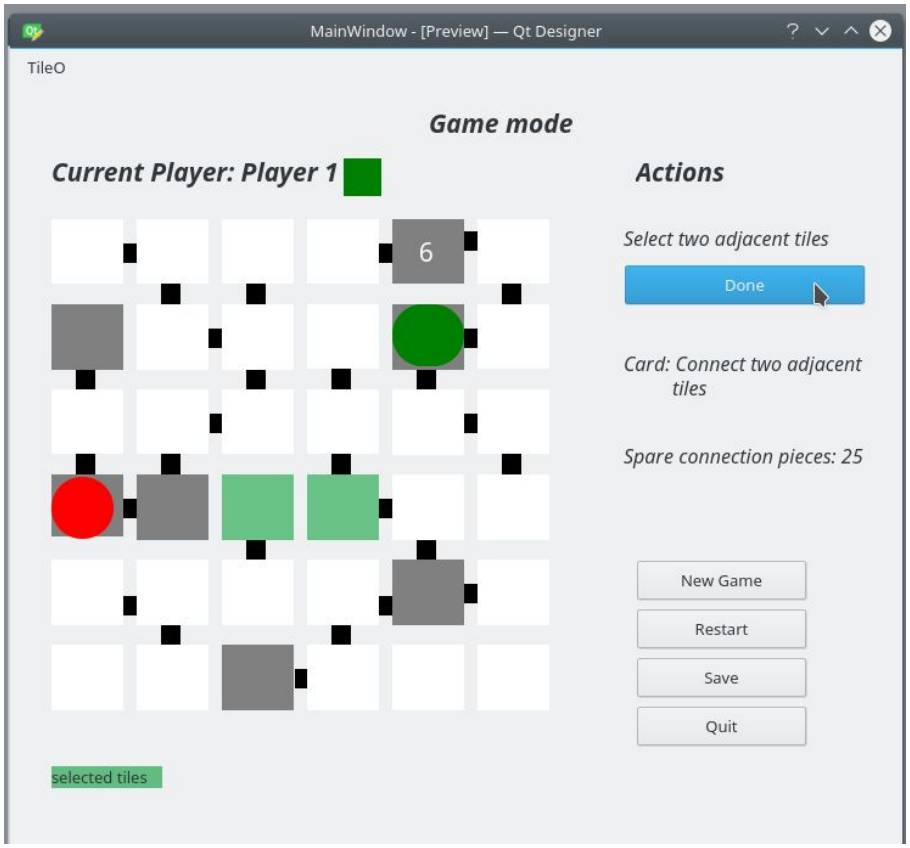
5. Action card "Roll the die for an extra turn": Sam

Controller:

- public void rollTheDiceActionPerformed (Player aPlayer, Die aDie)

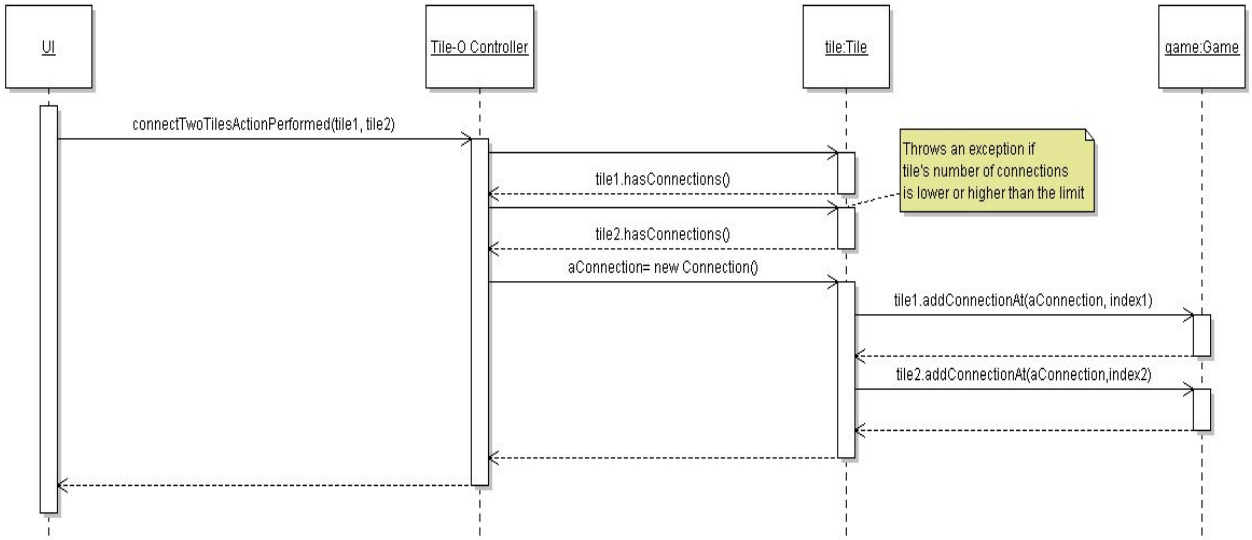


6. Action card "Connect two adjacent tiles with a connection piece from the pile of spare connection pieces": Sam

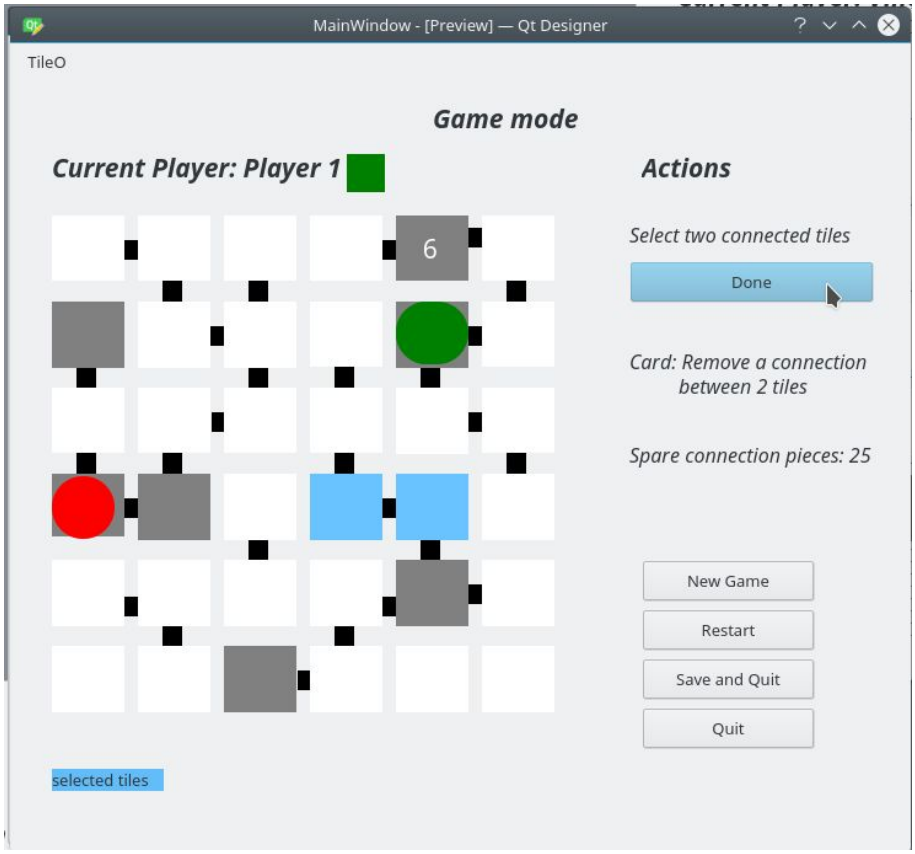


Controller:

- public void connectTwoTilesActionPerformed (Tile tile1, Tile tile2,)

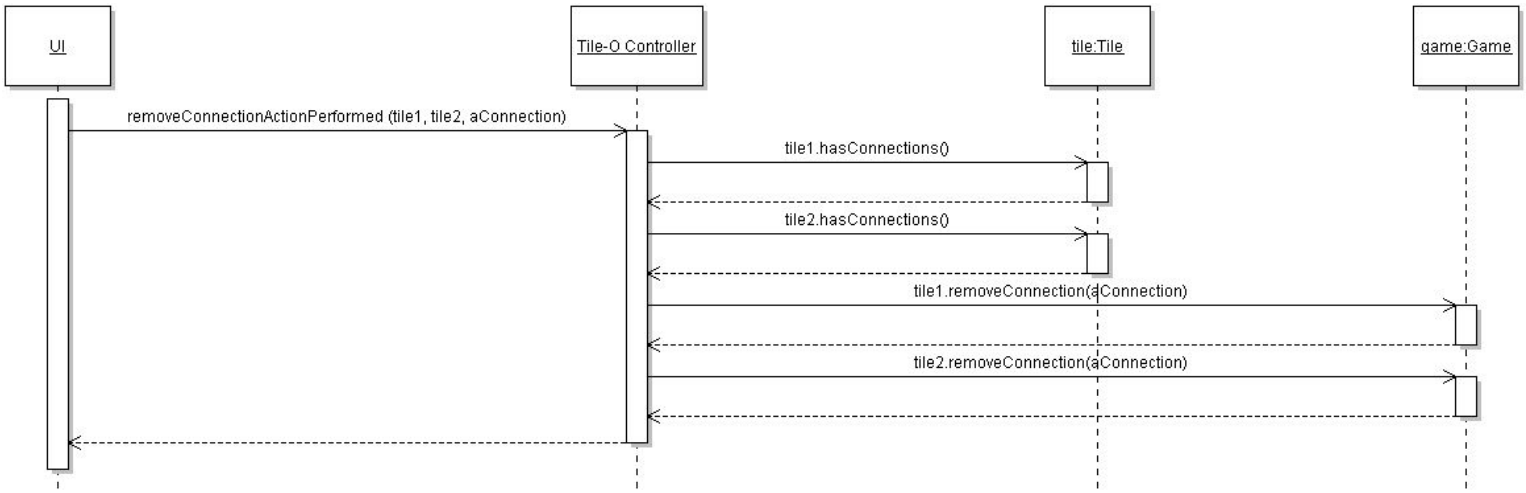


7. Action card "Remove a connection piece from the board and place it in the pile of spare connection pieces": Sam



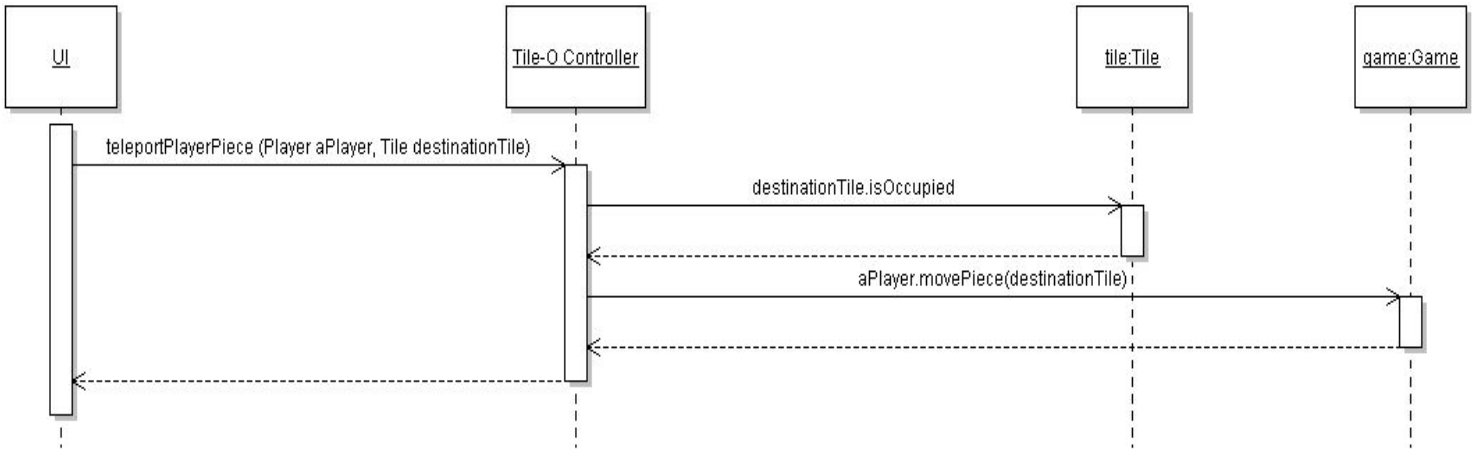
Controller:

- public void removeConnectionActionPerformed (Connection aConnection)



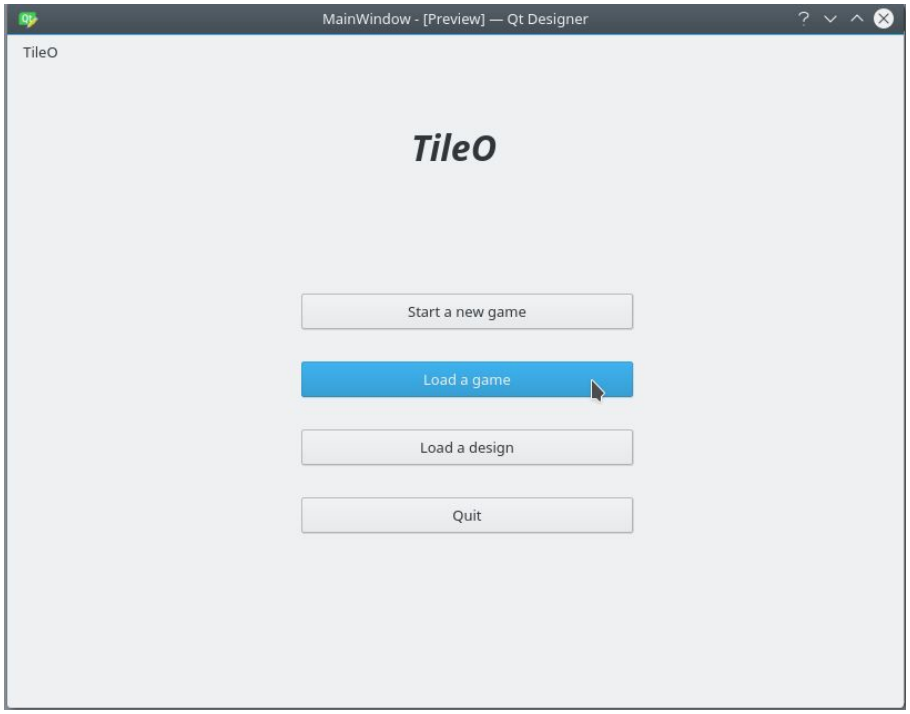
8. Action card "Move your playing piece to an arbitrary tile that is not your current tile": Sam

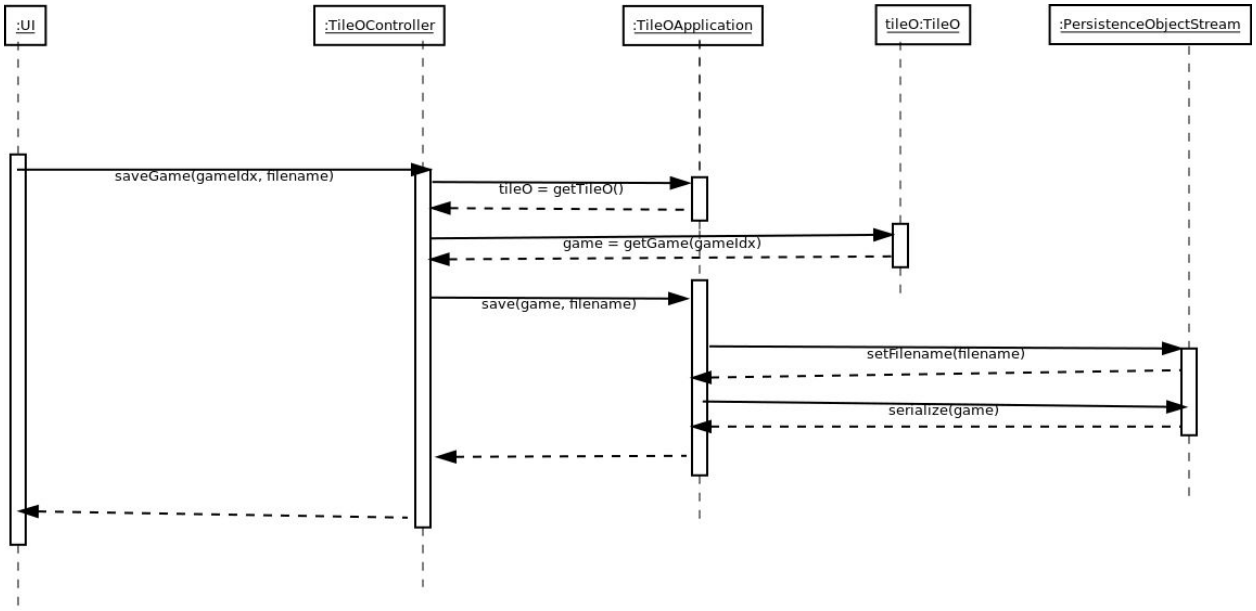
Controller:

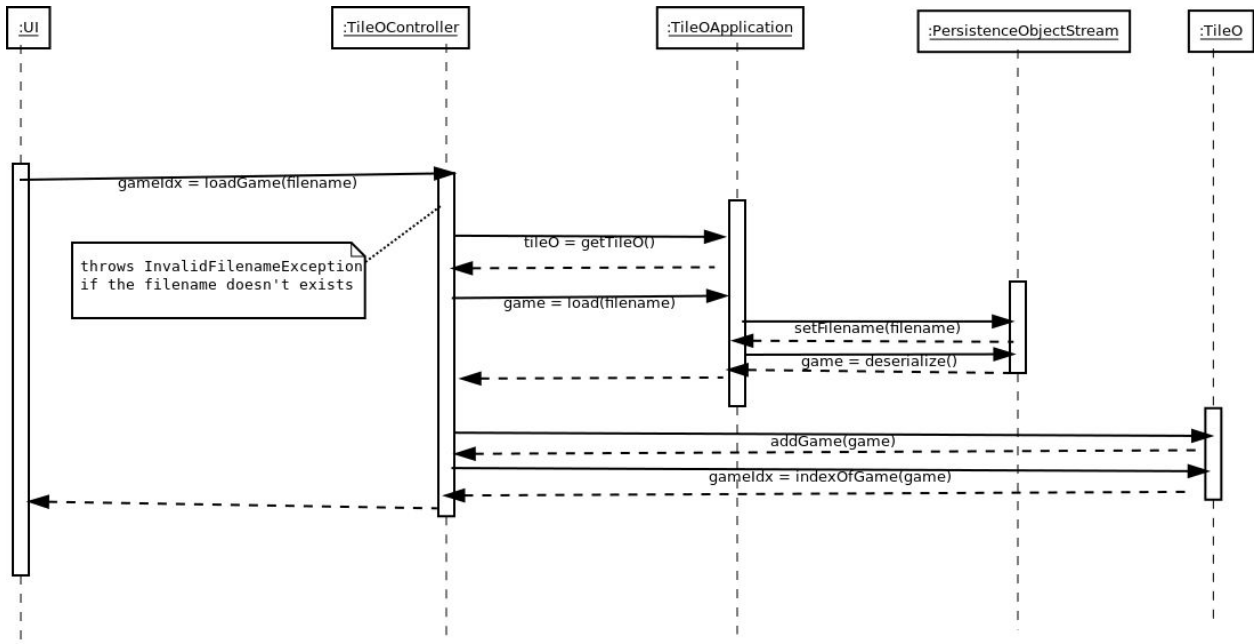


- public void teleportPlayerPiece (Player aPlayer, Tile destinationTile)

9. Save/Load game: Vincent


MainWindow - [Preview] — Qt Designer
TileO
TileO
Start a new game
Load a game
Load a design
Quit


:UI
:TileOController
:TileOApplication
tileO:TileO
:PersistenceObjectStream
saveGame(gameIdx, filename)
tileO = getTileO()
game = getGame(gameIdx)
save(game, filename)
setFilename(filename)
serialize(game)



Controller Interface:

- public void saveGame(int gameIdx, String filename)
- public int loadGame(String filename)