DELIVERABLE #2

UI Mock-Up, Sequence Diagram, Controller Interfaces

TEAM #6

CHAN, Chris – 260430627

HUANG, Yin Jun (Charles) – 260743592

LEI, Jun Yu (Justin) – 260738036

LIANG, Chun Ming – 260744492

VUONG, Victor – 260742996

ZHANG, Li – 260743980

ECSE 223 Model-Based Programming sect. 1, Winter 2017

Professor Gunter Mussbacher

McGill University, Montreal

Report submitted on Friday, February 10th, 2017

TABLE OF CONTENTS

**Design Mode** …………………………………………………………………………………………………………………………………………. 2

Feature D1: Create a game with a number of players ...…………………………………………………………….. 2

Feature D2: Place a tile on the game board ………………………………………………………………………………. 3

Feature D3: Remove a tile from the game board ………………………………………………………………………. 4

Feature D4: Connect two tiles with a connection piece ……………………………………………………………. 5

Feature D5: Remove connection between two tiles ……………………………………………………………….....6

Feature D6: Identify the hidden tile ……………………………………………………………………………………………7

Feature D7: Identify the starting tile of a player …………………………………………………………………………8

Feature D8: Identify an action tile ………………………………………………………………………………………………9

Feature D9: Select 32 cards from predefined choices ………………………………………………………………10

Feature D10: Save and load the game design …………………………………………………………………………..11

**Play Mode** ……………………………………………………………………………………………………………………………………….……13

Feature P1: Start a game ………………………………………....…………………………………………………………….. 13

Feature P2: Take a turn …………………………….……………………………………………………………………………… 14

Feature P3: Land on a tile ………………………………….…………………………………………………………………… 15

Feature P4: Take the first card from the deck of cards ..…………………………………………………………… 18

Feature P5: Action card “Roll a die for an extra turn” .……………………………………………………………... 19

Feature P6: Action card “Connect two adjacent tiles with a connection piece from the pile of spare connection piece”.…………………………….………………………………………………………………………………..20

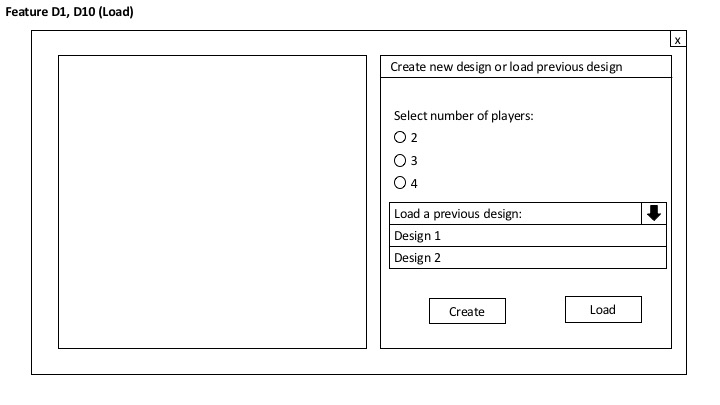
Feature P7: Action card “Remove a connection piece from the board and place it in the pile of spare connection pieces” ..………………………..………………………………………………………………………………21

Feature P8: Action card “Move your playing piece to an arbitrary tile that is not your current tile” ……………………………………………………………………………………….…………………………………………………22

Feature P9: Save and load game to continue playing at a later point ……………………………………….23

**Design Mode**

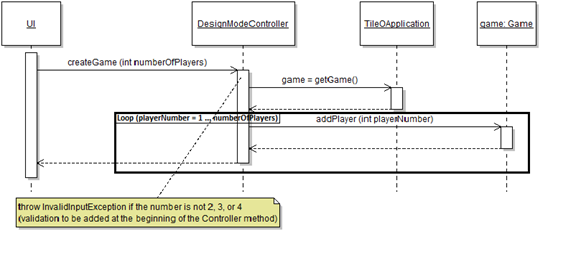
1. Create a game with a number of players - Jun Yu (Justin) Lei



Feature: Create game, given the desired number of players

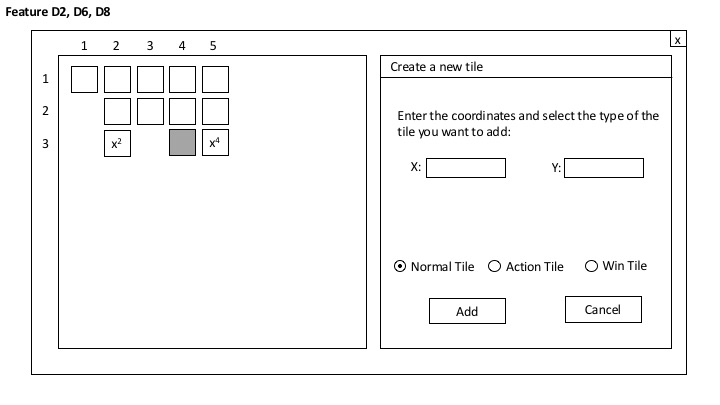
Specification of Controller Interface:

public void createGame(int numberOfPlayers) throws InvalidInputException



The designer creates one game, by specifying the number of players (2, 3 or 4). Then, players are added to the game one by one, while the player numbers are assigned at the same time.

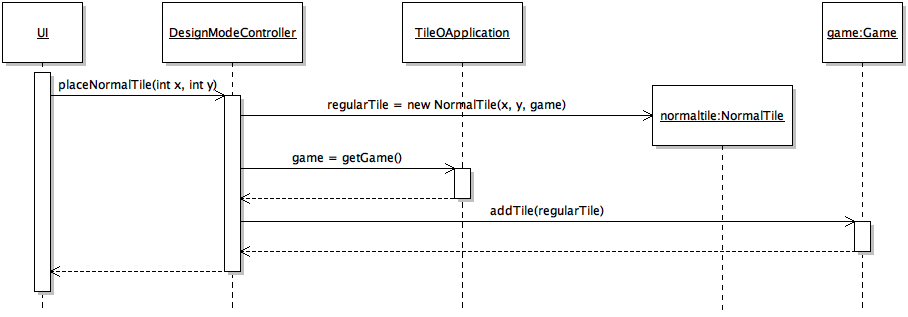
2. Place a tile on the game board - Chris Chan



Feature: Placing a new tile on the board game during design mode

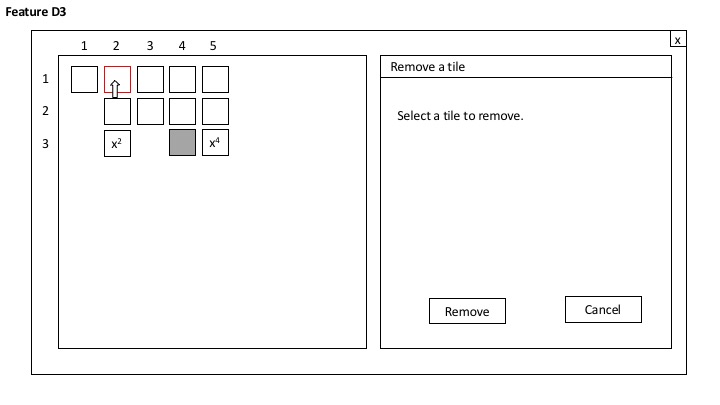
Specification of Controller Interface:

public void placeNormalTile(int x, int y) throws InvalidInputException



The designer creates a new tile from the normal class and sets it on the game board by indicating the x and y coordinates. He cannot input negative numbers as x and y coordinates.

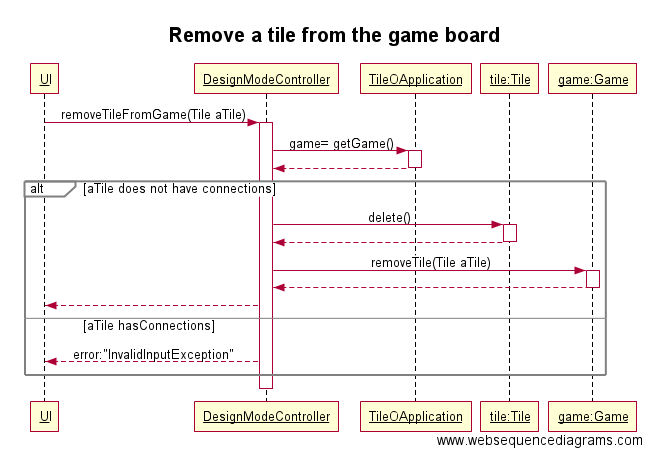
3. Remove a tile from the game board - Li Zhang



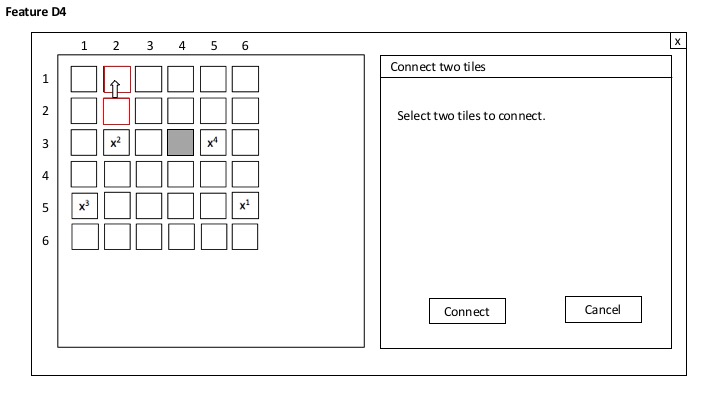
Feature: Removes a given tile from the game (from a mouseClick event). Cannot be deleted if it has connections - must have 0 connections before deleting

Specification of Controller Interface:

public void removeTileFromGame(Tile aTile) throws InvalidInputException



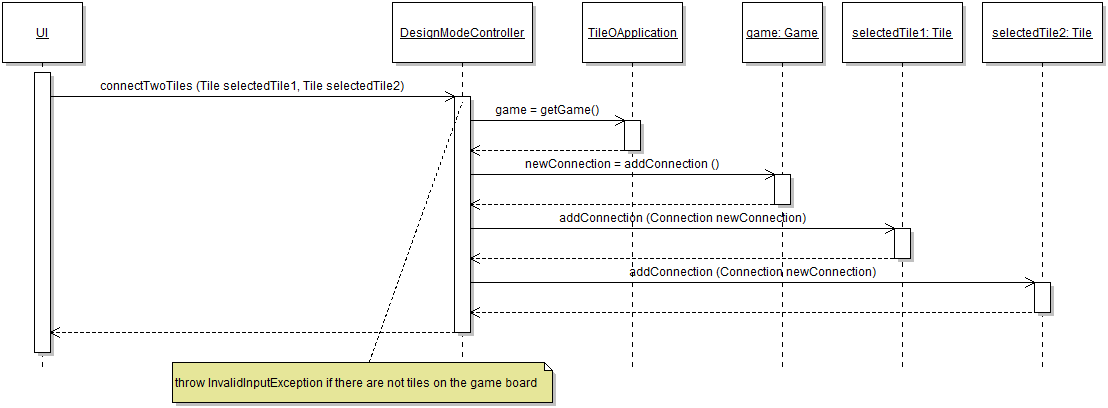
4. Connect two tiles with a connection piece - Jun Yu (Justin) Lei



Feature: Add a connection between two selected tiles in design mode

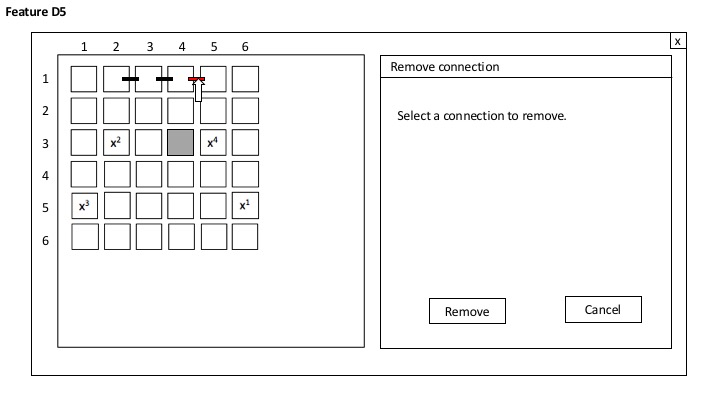
Specification of Controller Interface:

public void connectTwoTiles(Tile selectedTile1, Tile selectedTile2) throws InvalidInputException



The designer selects two adjacent tiles from the board game, and a connection between the two tiles appears; after the connection is created, it is added to the two tiles, and the connection is created.

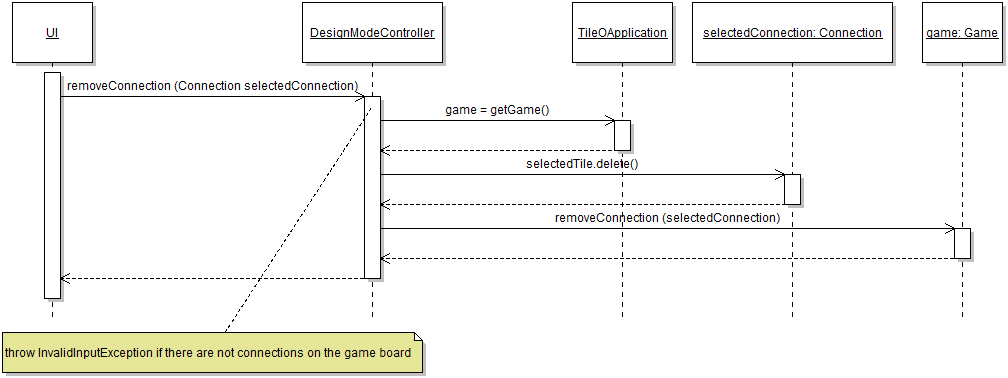
5. Remove connection between two tiles - Jun Yu (Justin) Lei



Feature: Remove connection between two tiles

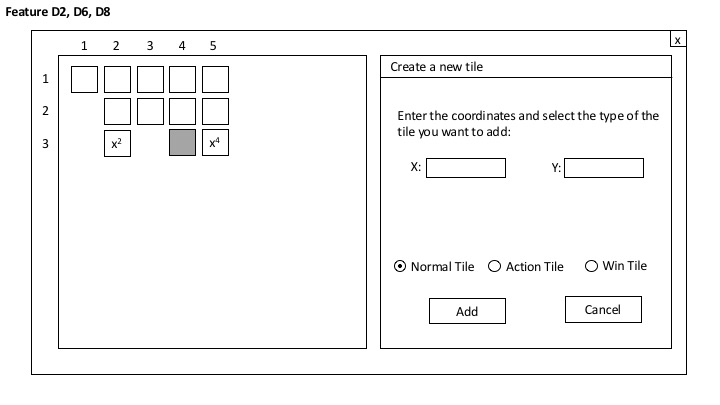
Specification of Controller Interface:

public void removeConnection(Connection selectedConnection) throws InvalidInputException



The designer selects a connection from the game board that he (or she) wants to delete. First, the connection is deleted because each connection has to be associated to a game at all times. Then, the connection can actually be removed.

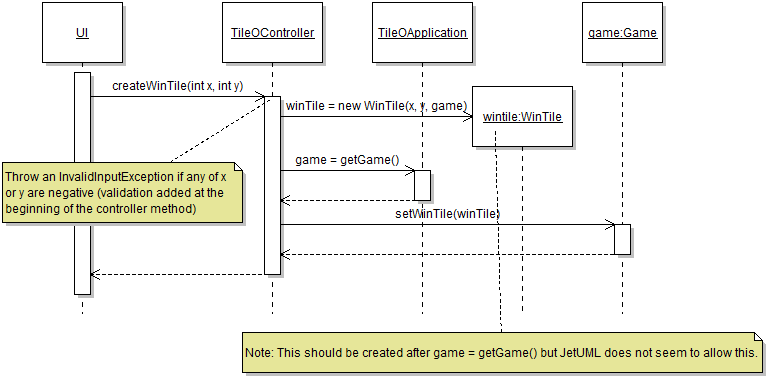
6. Identify the hidden tile - Chun Ming Liang



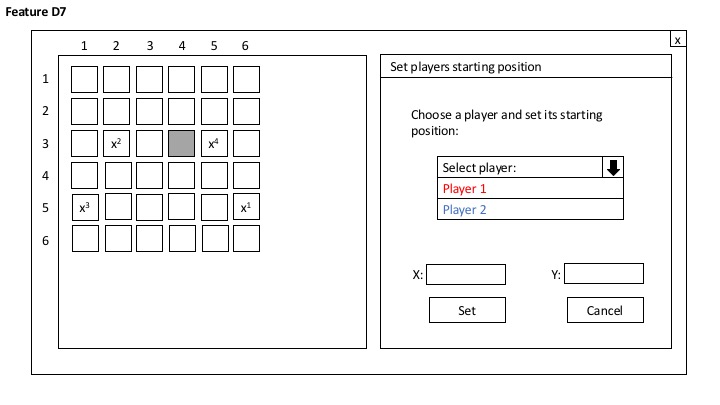
Feature: Create a new tile during design mode, and identify it as the hidden (win) tile.

Specification of Controller Interface:

public void createWinTile(int x, int y) throws InvalidInputException



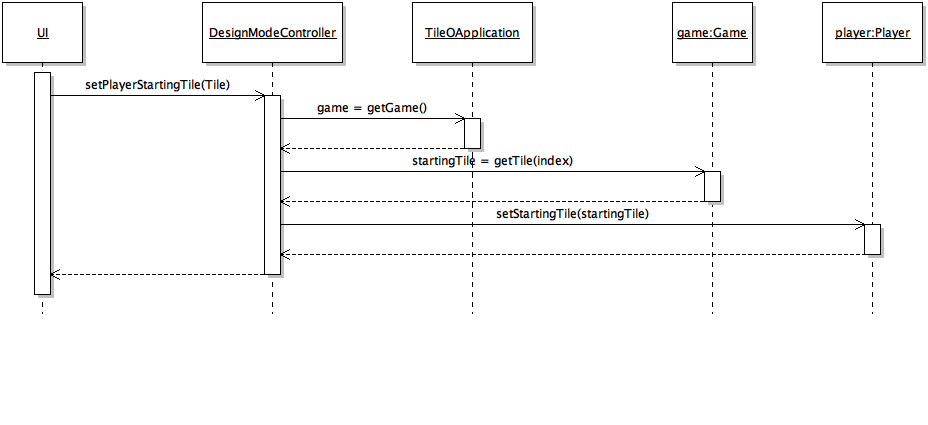
7. Identify the starting tile of a player - Chris Chan



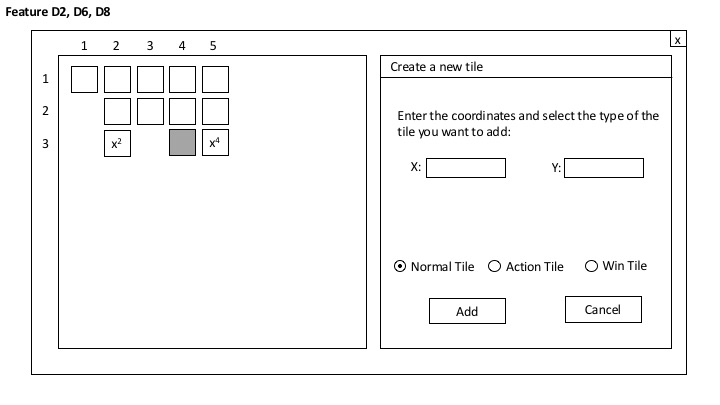
Feature: identifying the starting position of each player during design mode

Specification of Controller Interface:

public void setPlayerStartingTile(Tile aTile) throws InvalidInputException

The designer gets an index of a tile from the game class and sets it as the starting tile of each player. The designer does this every single player in the game.

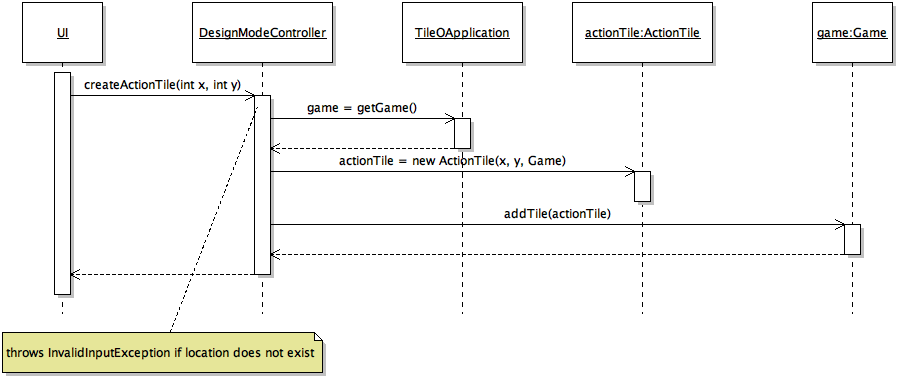
8. Identify an action tile (inactivity period not required for this iteration) - Victor Vuong



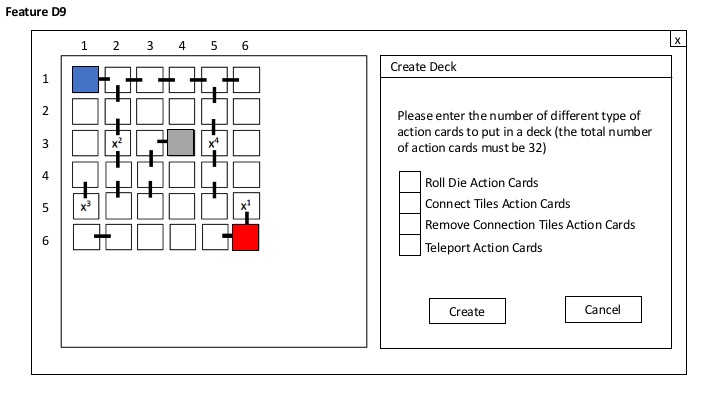
Feature: Create a new tile during design mode and identify it as an action tile

Specification of Controller Interface:

public void createActionTile(int x, int y) throws InvalidInputException



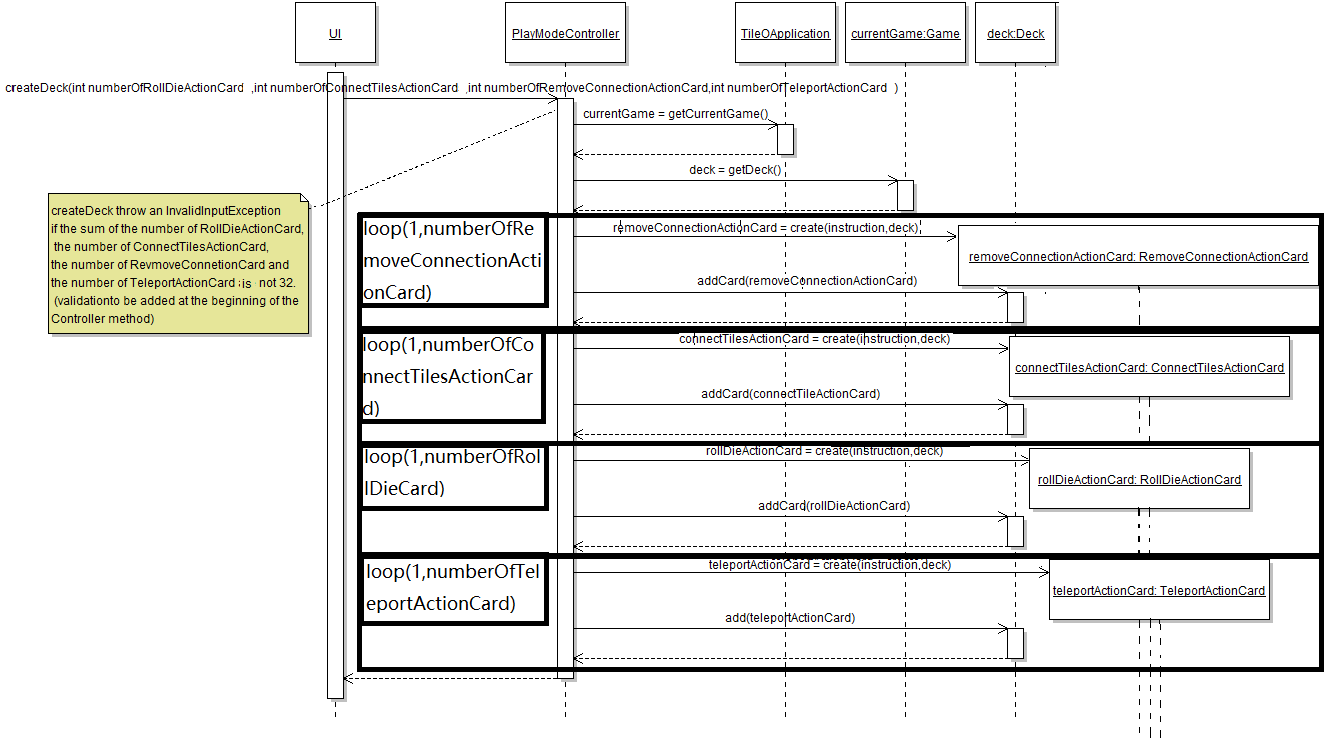
Note that the user will choose a specific location (x, y) and will create a Tile of type ActionTile at the desired location. An error will be thrown if the user chooses an invalid location such as an existing tile location or a non-existing location.

9. Select 32 cards from predefined choices - Yin Jun (Charles) Huang

Feature: Create Deck with 32 action cards that are selected by the designer

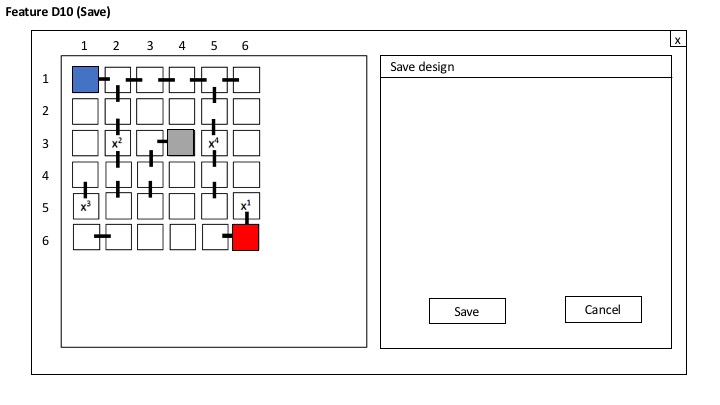
Specification of Controller Interface:

public void createDeck(int numberOfRollDieActionCard, int numberOfConnectTilesActionCard, int numberOfRemoveConnectionActionCard, int numberOfTeleportActionCard) throws InvalidInputException



The designer need to enter the number of different types of action card that he wants to put in a deck. Then, in the DesignModeController Class, the instances of different types of action card will be created and will be added in the Deck by loops.

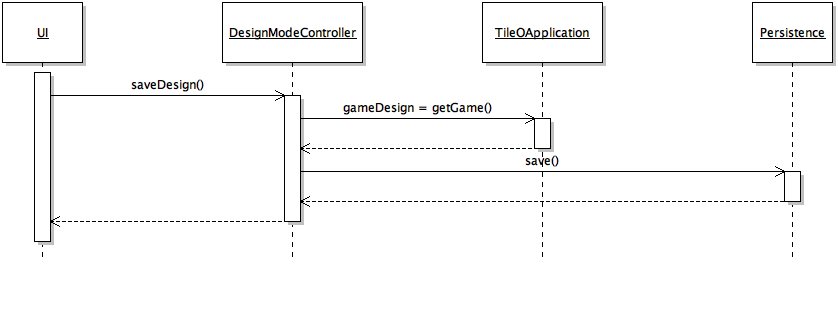
10. Save and load the game design - Victor Vuong



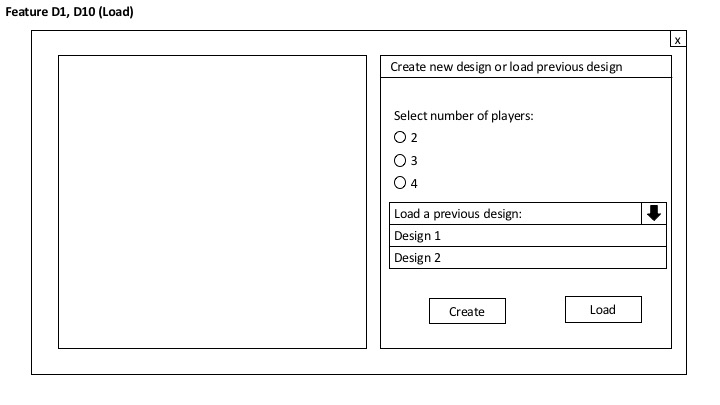
Feature: Saving a design. The designer can choose to save a game design.

Specification of Controller Interface:

public void saveDesign()



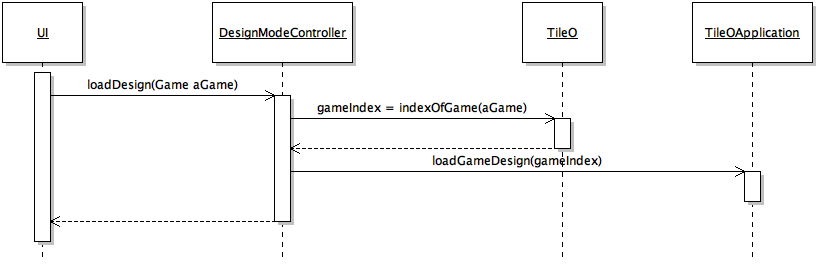
Note that there will be a Persistence class that allows us to serialize in order to save.

****

Feature: Loading a design. The designer can choose to load a game design.

Specification of Controller Interface:

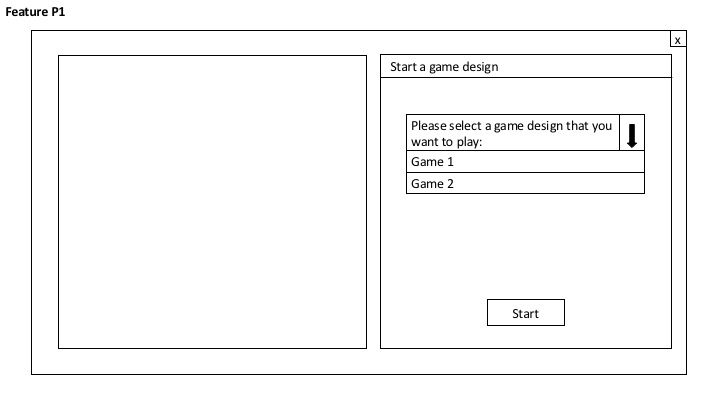
public void loadDesign(Game aGame) throws InvalidInputException



Note that the user will choose a game design from a drop down list to load. Therefore, the loadDesign method takes in a Game as argument. Also note that there will be a Persistence class that will allow us to serialize and deserialize in order to save and load.

**Play Mode**

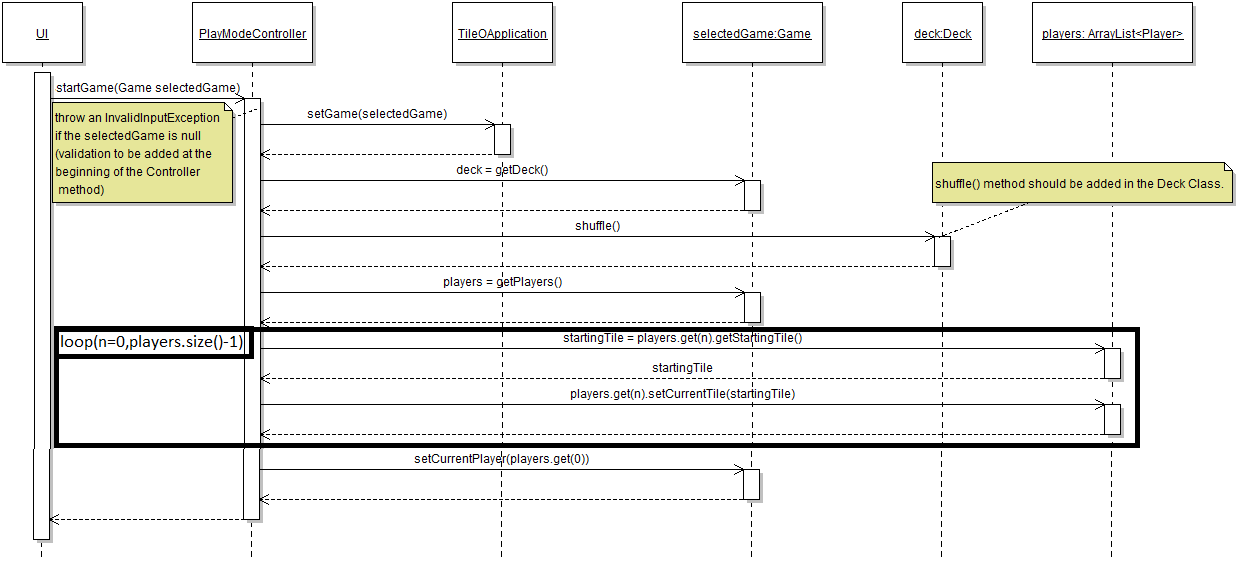
1. Start a game (shuffle the action cards, place players on board) - Yin Jun (Charles) Huang



Feature: start a game from a designed game list. Then, the action cards will be shuffle automatically, all the players will be automatically placed on the board, and player 1 will be set as current player.

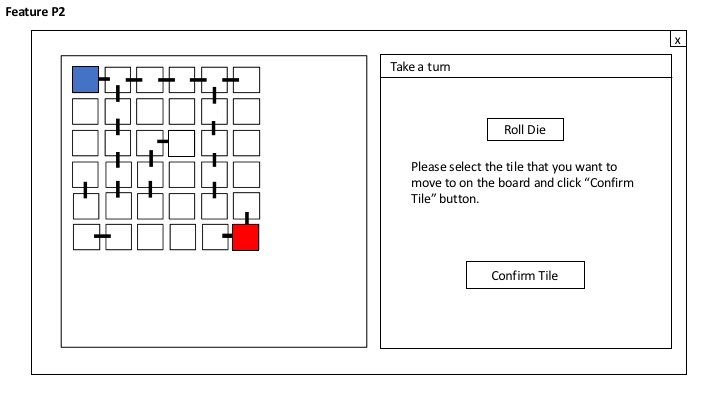
Specification of Controller Interface:

public void startGame(Game selectedGame) throws InvalidInputException

****

The player needs to select a designed game from the list, and then press the “Start” button in order to start a game. After he starts a game, the action cards in the deck will be shuffled automatically, and all the players will be placed to their startingTile, then player 1 will be set as current player.

2. Take a turn (roll the die, move to new position) - Yin Jun (Charles) Huang

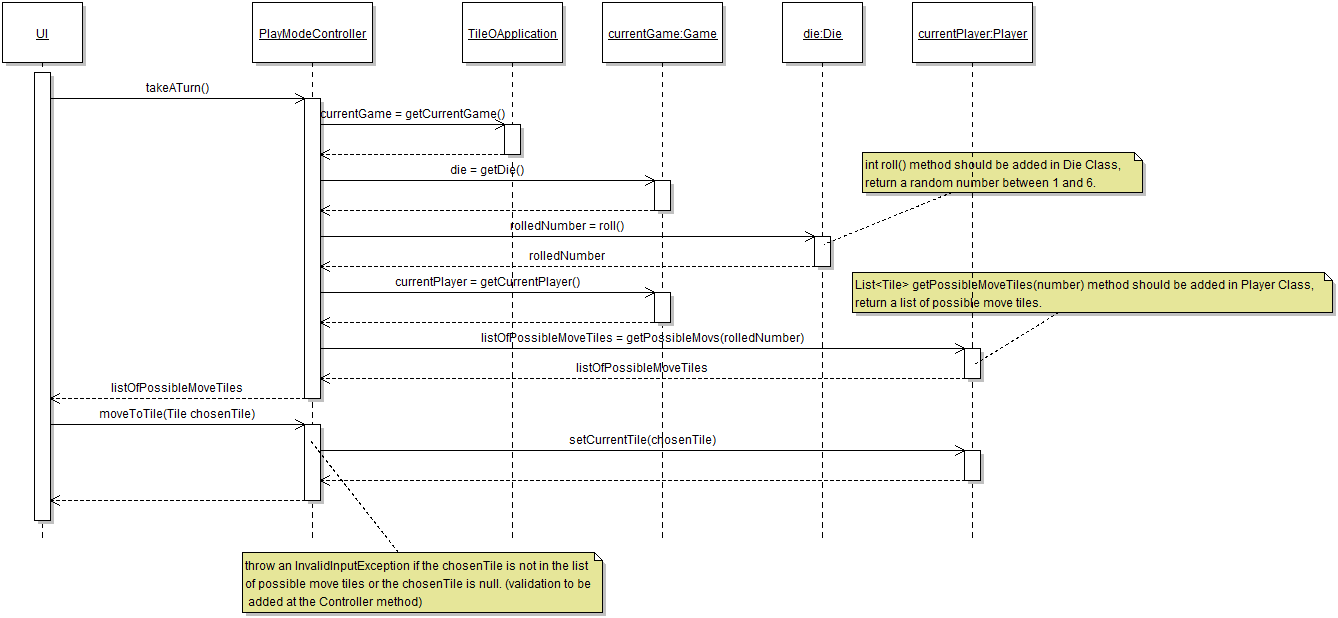


Feature: The player rolls the die, then the board will show the list of possible moves, then the player needs to choose one tile from the list, in order to move to this tile.

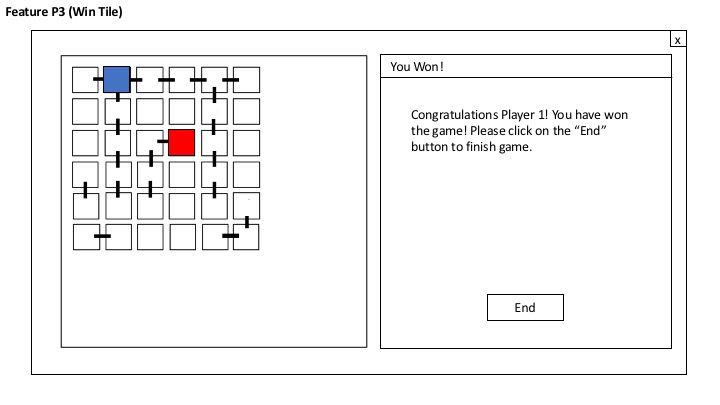
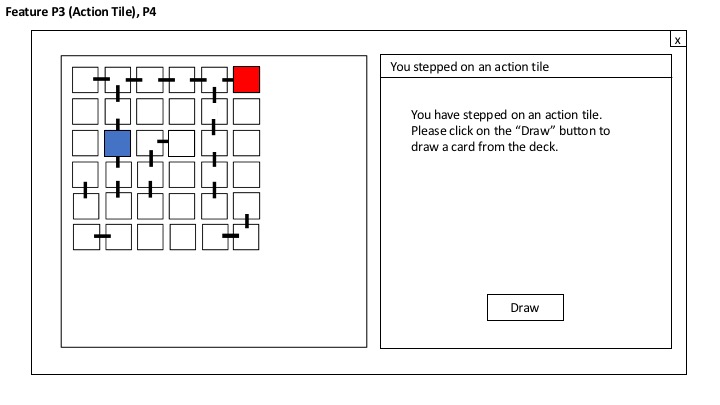
Specification of Controller Interface:

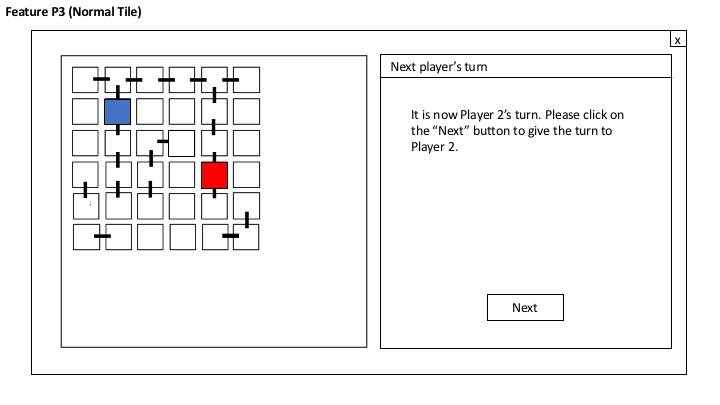
public List <Tile> takeATurn()

public void moveToTile(Tile chosenTile) throws InvalidInputException



The current player needs to press “Roll Die” button in order to take a turn. After he presses the “Roll Die” button, the list of possible moves will be shown on the board. Then, the current player needs to choose one from the list of the possible move tiles and press the “Confirm Tile” button to move to this tile.

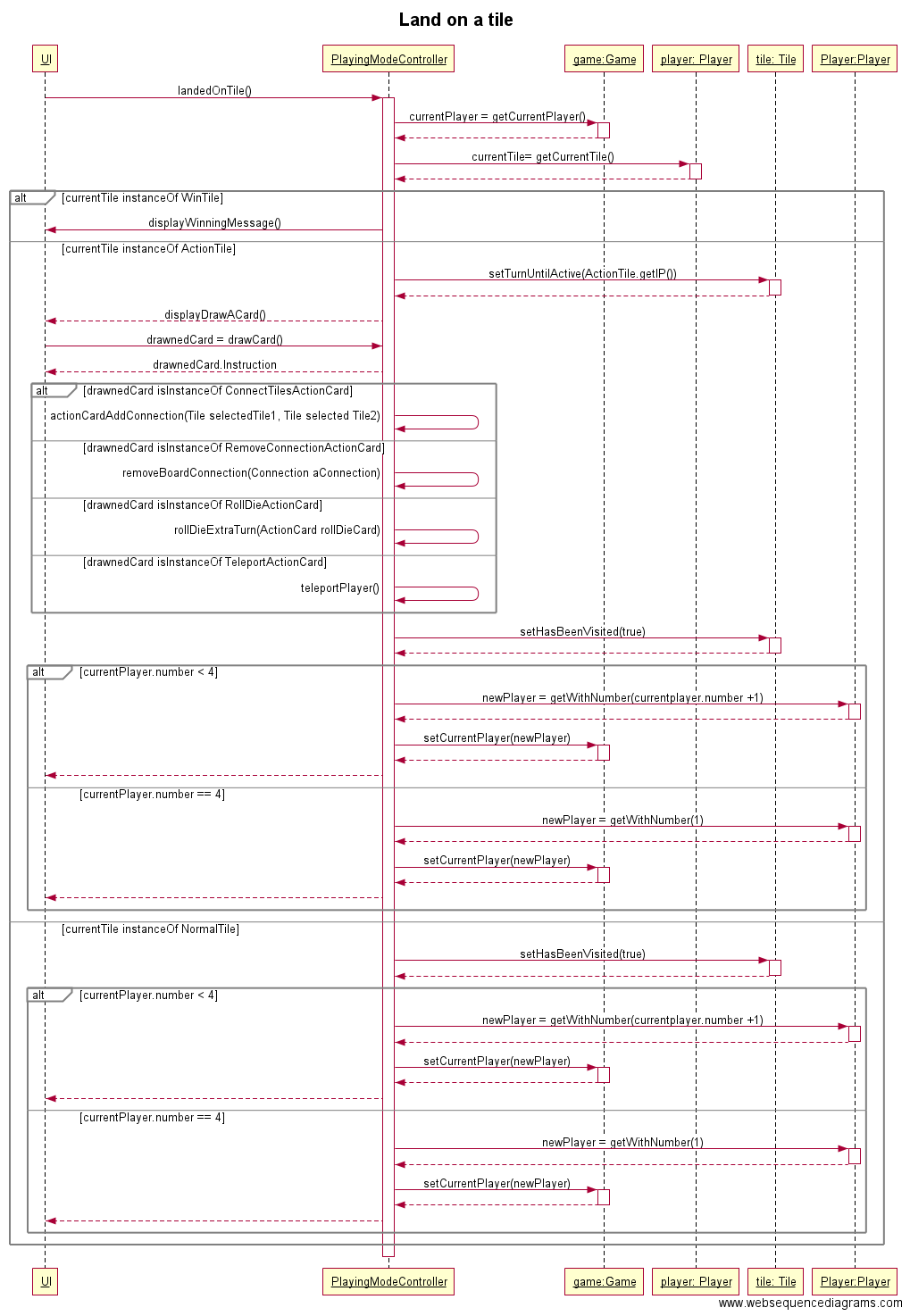
3. Land on a tile (basic behavior for hidden, regular, and action tiles) - Chris Chan

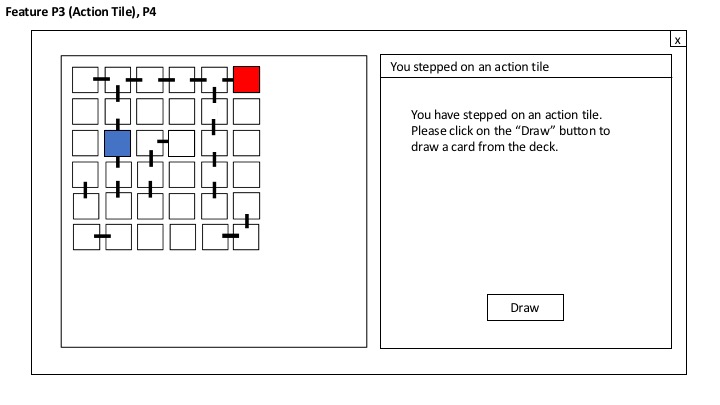


Feature: when a player lands on a tile, depending on the tile, the player has to either end his turn if the player lands on a normal tile, draw from the deck if the player lands on an action tile, or end the game (win the game) if the player lands on the hidden tile during play mode.

Specification of Controller Interface:

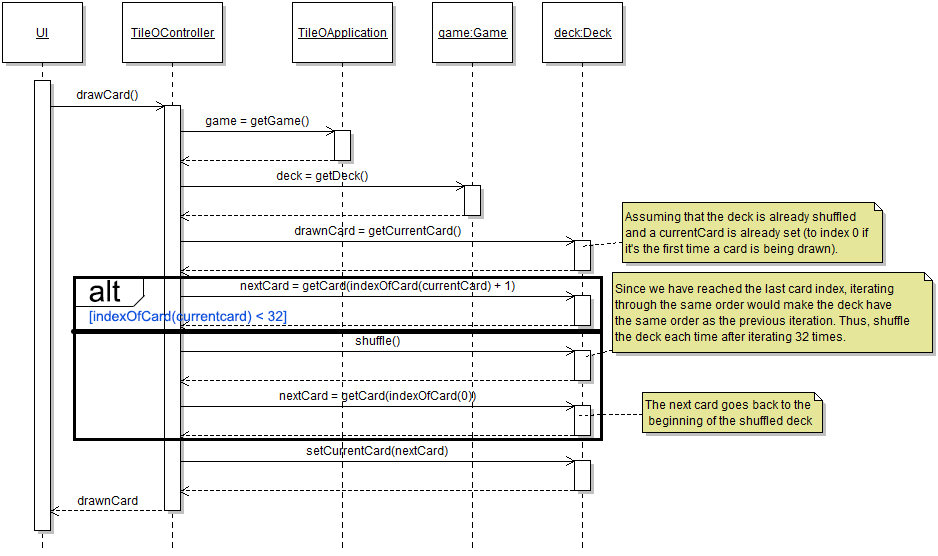
public void landedOnTile()

4. Take the first card from the deck of cards - Chun Ming Liang

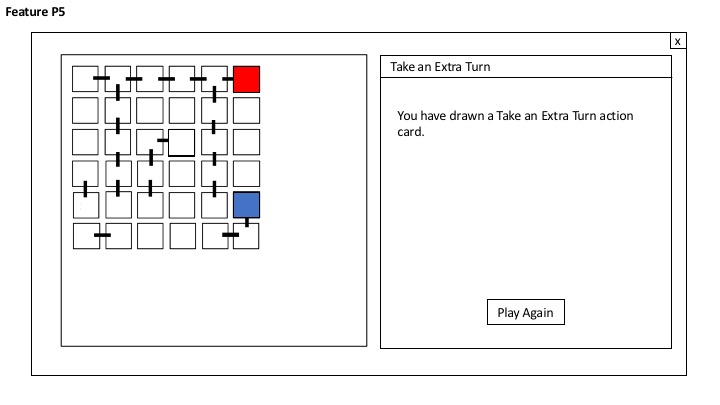
Feature: Create a new tile at a specific location during design mode, and identify it as the hidden (win) tile. Set this tile as the winTile of the game the game. Checks for correct input.

Specification of Controller Interface:

public void createWinTile(int x, int y) throws InvalidInputException



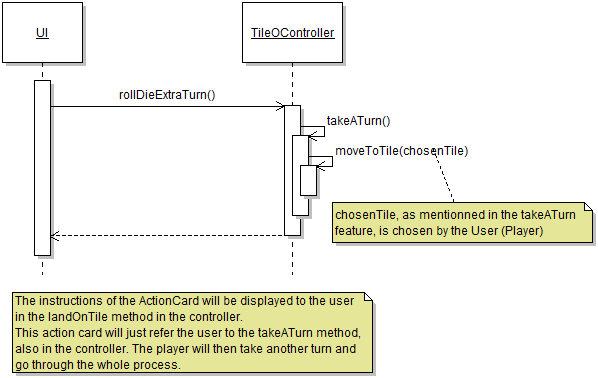
5. Action card "Roll the die for an extra turn" - Chun Ming Liang



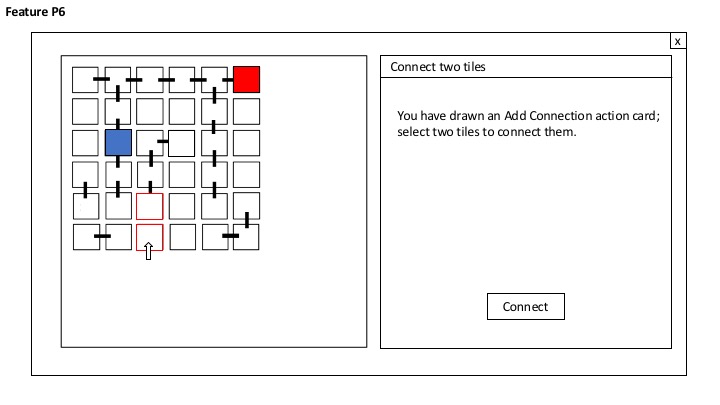
Feature: Execute the effect mentioned in the instructions string of the rollDieExtraTurn: roll the die and get the list of possible moves; choose the tile that the player wants to move to (if possible).

Specification of Controller Interface:

public void rollDieExtraTurn()

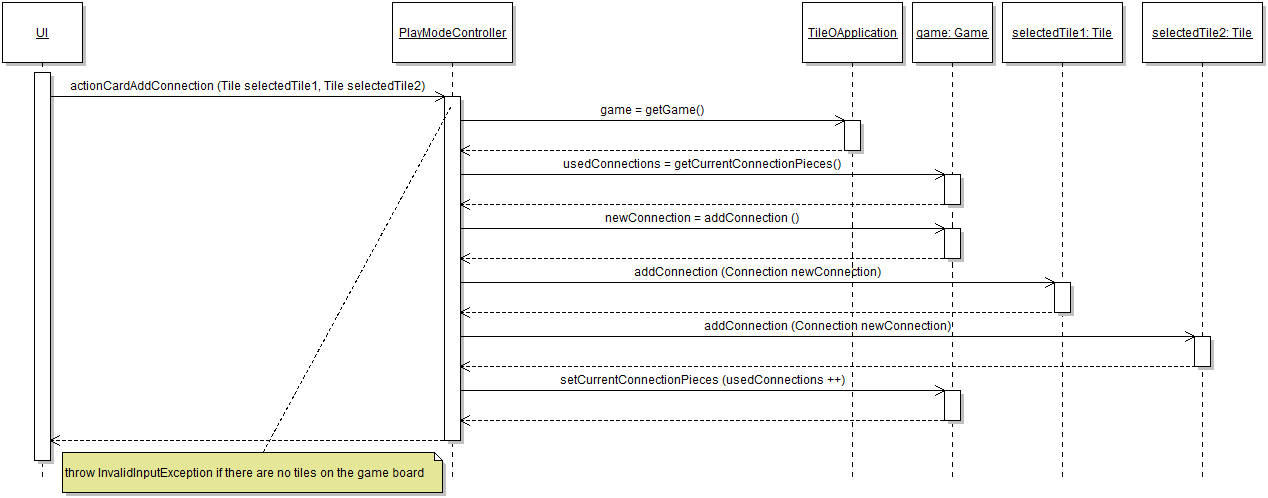


6. Action card "Connect two adjacent tiles with a connection piece from the pile of spare connection pieces" – Jun Yu (Justin) Lei



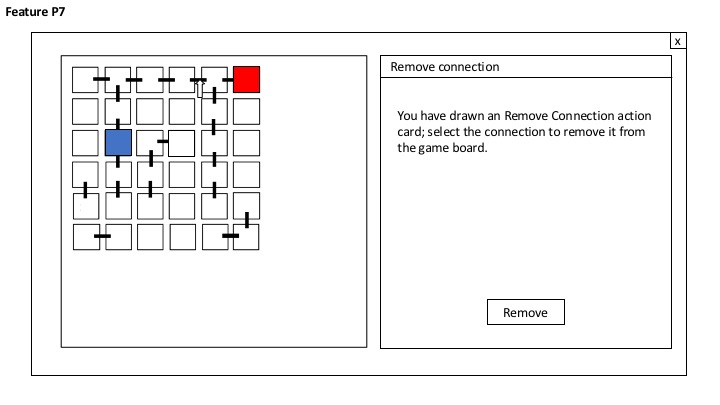
Feature: Add a connection from a pack of connections between two selected tiles, after drawing an action card

Specification of Controller Interface: actionCardAddConnection (Tile selectedTile1, Tile selectedTile2) throws InvalidInputException



The number of used connections is counted from the board. The number of used connection is compared to the spare connection pieces (32). If the number of used connections exceeds 32, nothing happens. If the number of used connections is lower than 32, then the player selects two tiles from the game, and a connection is added.

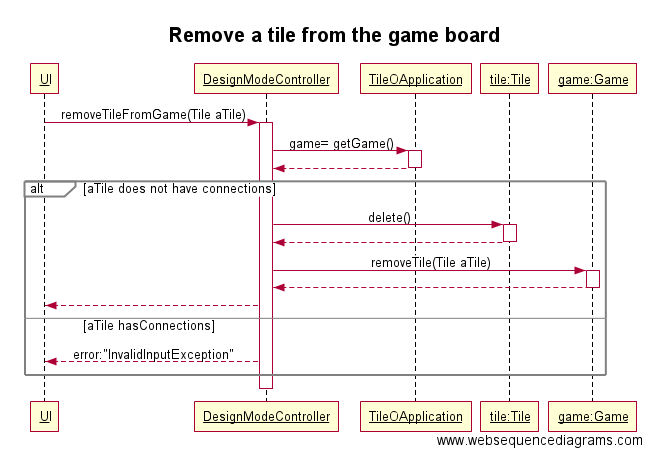
7. Action card "Remove a connection piece from the board and place it in the pile of spare connection pieces" - Li Zhang



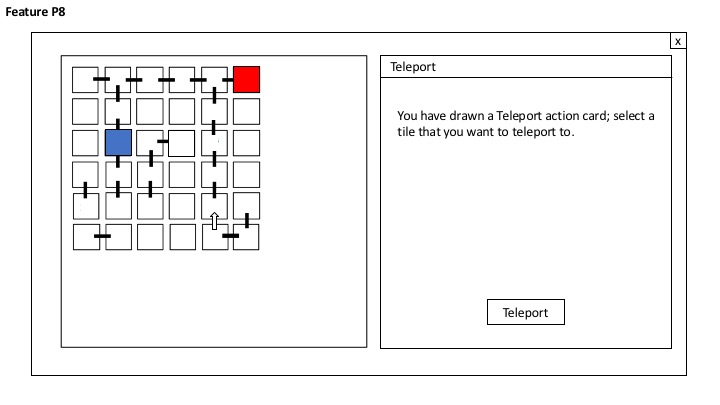
Feature: Removes a current connection piece on the game due to an action card

Specification of Controller Interface:

public void removeBoardConnection(Connection aConnection) throws InvalidInputException



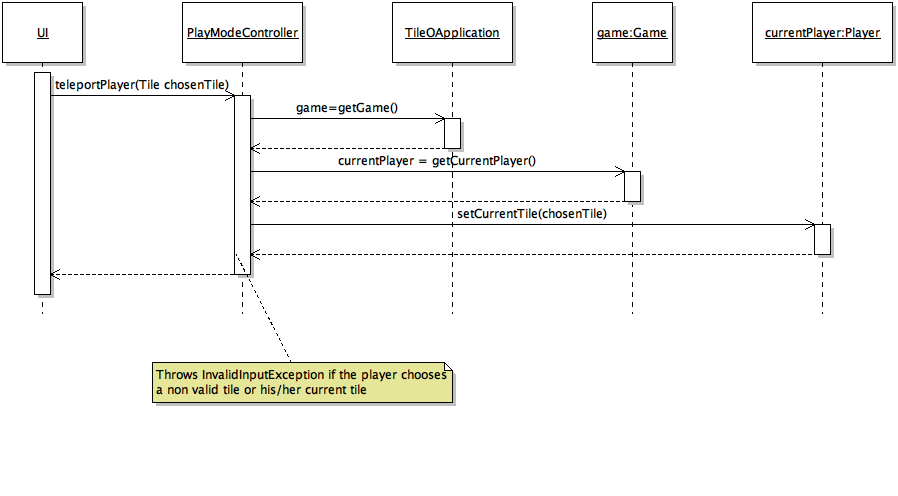
8. Action card "Move your playing piece to an arbitrary tile that is not your current tile" - Victor Vuong



Feature: The player chooses an arbitrary tile to teleport to that is not his or her current tile.

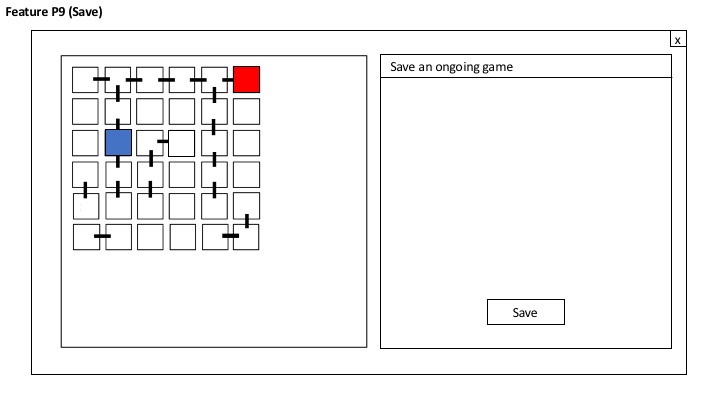
Specification of Controller Interface:

public void teleportPlayer(Tile chosenTile) throws InvalidInputException



Note that the player will choose a chosenTile by interacting with the user interface with a mouse listener. Therefore, the method takes in the chosenTile as an argument. The player’s current tile will then be set to that tile.

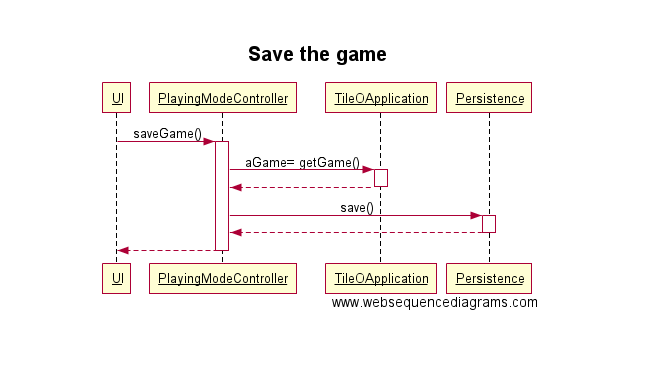
9. Save and load game to continue playing at a later point - Li Zhang



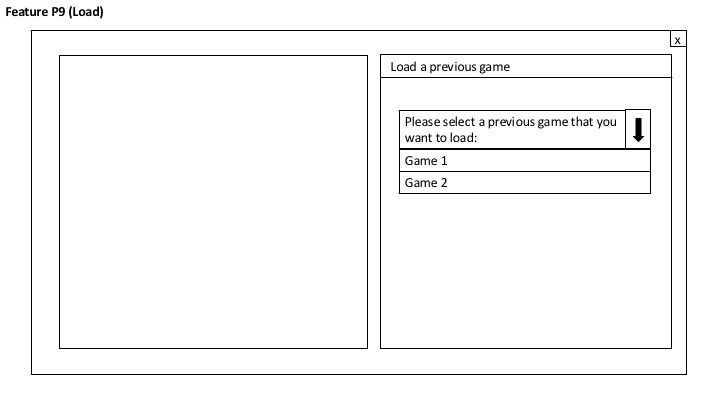
Feature: Saves the current game

Specification of Controller Interface:

public void saveGame()



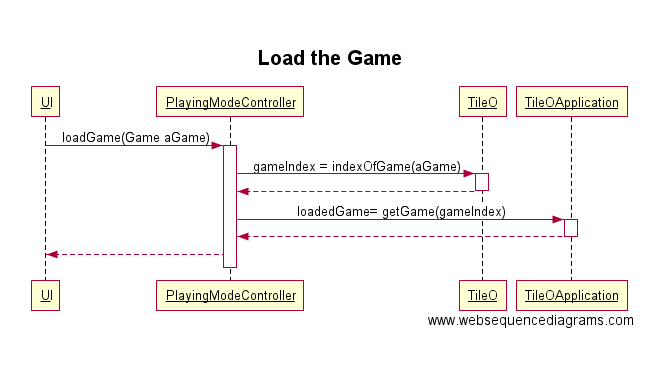
The UI saves the current game from the TileOApplication and then saves it using the persistence.



Feature: Loads a game given a selected game

Specification of Controller Interface:

public Game loadGame(Game aGame)



The UI, given aGame of class Game, will find the index of that game and then load it