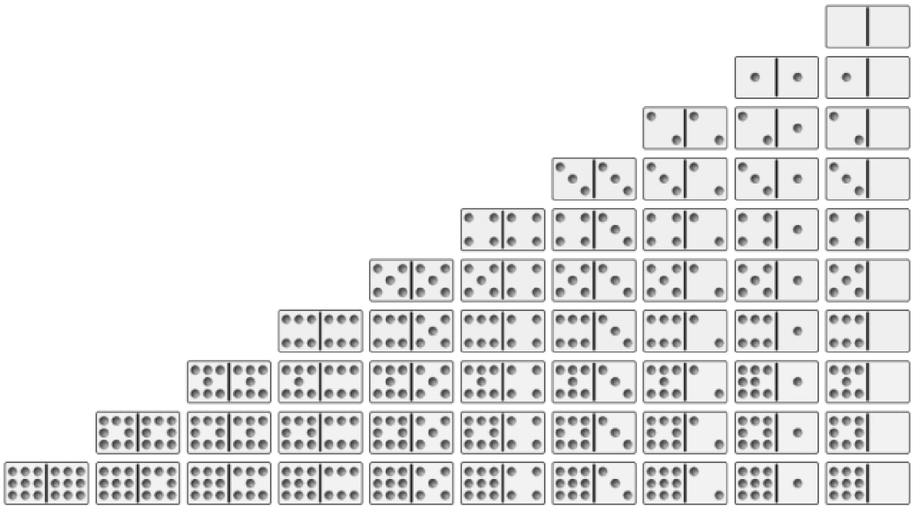
CS 351
Project 2
Mexican Train
Due 10-01-20

Joseph Haugh



Project Description

- Mexican Train is a game played with either a double-12 or double-9 set of dominoes
- The objective of the game is to be the first player to clear your hand of dominoes
- During each round a double domino is placed at the center of the table
- This double domino starts out with the highest double and goes down by 1 each round
- For example if you were playing with a double-9 set then you would start with the double-9 then go to double-8 then double-7 and so on
- Each player is given a set number of dominoes to start out
- Each player has a ”train” given to them with an additional public train, called the Mexican Train, also available
- Initially players must play a domino that matches the center domino either on their own train or the Mexican Train
- If a player is unable to play during their turn the must draw from the boneyard
- If they still cannot play then they must ”open”, which means other players can play on it, their train
- A round ends when a player has emptied their hand

- Then the total pips of each player are tallied and the next round begins
- To win you must have the least amount of total pips by the end of the last round, double-0
- We will be following the rules given here: `https://www.mexicantrainrulesandstrategies.com/`.

Project Requirements

- Write at least two versions of Mexican Train (more details below)
- Up to 4 players, with any mix of computers and humans possible
- Double-9 set of dominoes (shown above, 55 dominoes in total)
- Dominoes must form two parallel rows when shown (see example)
- AI capable of creating a combo and playing its highest value domino when given the opportunity
- Implement all rules given above and on the above website
- Design must be object-oriented, modular, and resistant to change
- Provide a design diagram for each version
- Must be sequential
- Must be built incrementally

Required Versions

You must submit a jar and design documents for at least the following two versions of the game. Make sure to document how to run each version in your README. When building the code make sure to build it incrementally. This means that each iteration should deliver a subset of the overall functionality of the application. Each subsequent iteration should build on the previous with minimal changes to existing code. Your game logic should be largely unchanged between the two versions.

- Version 1: Standard Input/Output
- Version 2: Graphics through JavaFX

Example Console Application

```
Welcome to Mexican Train!
If you are not familiar with the rules please browse to the following site: https://www.mexi
Up to 4 players can play with any mix of human and computer players.
Please enter the number of human players:
2
Please enter the number of computer players:
1
GameState:
Humans:
Player1: [[2 | 2], [1 | 1], [5 | 9], [0 | 2], [7 | 8], [1 | 7], [0 | 5], [1 | 9], [6 | 9],
Player2: [[2 | 7], [1 | 6], [8 | 8], [0 | 0], [4 | 6], [6 | 6], [3 | 3], [1 | 2], [7 | 7],

Computers:
Computer3: [[5 | 5], [0 | 8], [0 | 1], [1 | 8], [0 | 6], [2 | 8], [6 | 7], [8 | 9], [0 | 9],

Center: [9 | 9]
Board:
Player1(false):

Player2(false):

Computer3(false):

Mexican Train(true):

Boneyard:
[0 | 3] [1 | 5] [2 | 4] [3 | 4] [3 | 6] [3 | 8] [4 | 8] [5 | 7]
    [0 | 7] [2 | 3] [2 | 9] [3 | 5] [3 | 7] [4 | 7] [5 | 6] [9 | 9]
Current Player:1

Player1's Turn
[p] play domino
[d] draw from boneyard
[q] quit
p
Which domino?
2
Which train?
Player2
That train is not open, please try again.
Which train?
Player1
Flip?
Yes
```

```
GameState:
Humans:
Player1: [[2 | 2], [1 | 1], [0 | 2], [7 | 8], [1 | 7], [0 | 5], [1 | 9], [6 | 9], [1 | 4], [
Player2: [[2 | 7], [1 | 6], [8 | 8], [0 | 0], [4 | 6], [6 | 6], [3 | 3], [1 | 2], [7 | 7], [

Computers:
Computer3: [[5 | 5], [0 | 8], [0 | 1], [1 | 8], [0 | 6], [2 | 8], [6 | 7], [8 | 9], [0 | 9],

Center: [9 | 9]
Board:
Player1(false): [9 | 5]

Player2(false):

Computer3(false):

Mexican Train(true):

Boneyard:
[0 | 3] [1 | 5] [2 | 4] [3 | 4] [3 | 6] [3 | 8] [4 | 8] [5 | 7]
    [0 | 7] [2 | 3] [2 | 9] [3 | 5] [3 | 7] [4 | 7] [5 | 6] [9 | 9]
Current Player:2

...

GameState:
Humans:
Player1: [[2 | 2], [1 | 1], [0 | 2], [7 | 8], [1 | 7], [0 | 5], [1 | 9], [6 | 9], [1 | 4], [
Player2: [[2 | 7], [1 | 6], [8 | 8], [0 | 0], [4 | 6], [6 | 6], [3 | 3], [1 | 2], [7 | 7], [

Computers:
Computer3: [[5 | 5], [0 | 8], [0 | 1], [1 | 8], [0 | 6], [2 | 8], [6 | 7], [8 | 9], [0 | 9],

Center: [9 | 9]
Board:
Player1(false): [9 | 5]
                    [5 | 4]

Player2(false): [9 | 4]

Computer3(false): [9 | 3]

Mexican Train(true):

Boneyard:
```

```
[0 | 3] [1 | 5] [2 | 4] [3 | 4] [3 | 6] [3 | 8] [4 | 8] [5 | 7]
    [0 | 7] [2 | 3] [2 | 9] [3 | 5] [3 | 7] [4 | 7] [5 | 6] [9 | 9]
Current Player:2
```

Submitting Your Project

Submit the link to your CS git project to UNM Learn before the deadline. Make sure your project follows my submission guidelines as given on the course website.