Marvin Chedid (302446)– Johnny Borkhoche (296169)

Etape 12 Rapport

Main Changes:

The biggest change we've made is undoubtably adding a new mode of playing : the 3 players mode. This extension is optional, the users may still choose to play the original 2 players version of tChu. It works the following way:

- When starting up the game through ServerMain, a welcome screen pops up at the start. It will introduce the game's basics as well as contains a button that allows the player that will be hosting the game's server to choose whether to enter 2 players or 3 players mode.
- When pressed, the button will open up a new modal dialog box allowing the player to pick his choice for the game of tChu.
- When the player presses the confirm button, the JavaFX stages are hidden, and the hosting player will be awaiting either 1 or 2 proxy player's connections. If required, the correct program arguments must be provided to the connecting player's ClientMain (just like in Etape 11). By default, the ClientMain's will try to connect to the localhost.

Overall, the 2 players game has not changed, it still follows the exact same rules as Etape 11. However, some slight alterations were made to the rules of the 3 players mode:

- Double routes are not restricted to only 1 player anymore, 2 different players can now claim the two parallel routes in a double route. We felt like such a change made sense since with 3 players the map may feel slightly more crowded, so it was an opportunity to give the players a little more leeway.

Technical Details:

To accomplish the feat of allowing the host player to choose the game mode, we had to add some new classes and functions:

- Made a new WelcomeScreenView class that creates a JavaFX node containing a nice little tChu drawing shown in the welcome screen when the host player runs ServerMain.
- In both game modes, ServerMain doesn't simply create a game instance of tChu and runs it on a new thread anymore. Instead, it creates a new instance of what we called ApplicationMainAdapter, a class containing the adapted interface components for the game of tChu. As its name suggests, this class adapts ApplicationMain, a class which contains the interface components and has the following properties:
    - It's constructor creates a new stage which contains an introduction to the game as well as the button that allows the user to select a game mode.
    - The constructor also takes a new handler (added to the ActionHandler interface) as a parameter, which is adapted to handle the event of the player wanting to open the modal dialog box to choose a game mode.

- It's methods can present the game-mode selecting window, as well as run the game depending on the selected mode (which used to be done directly in ServerMain). More specifically, it's run() method takes one more newly defined ActionHandler that is used to receive the game mode chosen by the host player and passing it over so that the server gets started and awaits connections from the other players.

Luckily, most of our previous code could be very easily refactored to take into account more than 2 players (ex: by passing a list of players, we iterated over all players of the list, instead of dealing with each player individually). However, some special attention had to be taken to correctly set up the names of the 3 players (in the 3 graphical interfaces) as well as their corresponding colors; this was due to the fact that the serializer/deserializer wasn't configured to deal with 3 players.

We also made an extra style sheet used for the welcome screen, as well as added some CSS classes to the given resources.

Furthermore, for the 3 players game mode, the functionality of the canDrawTickets() method had to be changed:

- Since the game initially has 46 tickets, and with all players having to pick from 5 tickets at the start of the game, and knowing that unpicked tickets are discarded, there was an issue with the number of tickets left after the players picked their initial tickets. Indeed, in a 2 player game, the number of tickets left would be 46-10=36 which is fine (divisible by 2). However, for 3 players, the number of tickets left would be 46-15=31 which isn't divisible by 3. There were 2 possible solutions, either remove 1 ticket from the game or change the condition on canDrawTickets(). We found the latter more natural and implemented it accordingly.