ANIMATIONS

User end: For this part, we wanted to make the GUI more engaging and user-friendly by adding animations to objects like cards and routes :
- an animation when the player draws a card from the face-up cards
- one when a card is added or removed from the cards in the hand of the player (either by drawing a card or by claiming a route)
- one for when a route is claimed by a player.

Java implementation: In order to implement the above-mentioned animations, we used some helpful tools provided by JavaFX. More specifically the different transition classes such as TranslateTransition, ScaleTransition, FadeTransition as well as SequentialTransition and ParallelTransition which help group and execute multiple transitions together. All the animations were implemented as static void methods inside the classes DecksViewCreator for the cards, and MapViewCreator for the roads. Many methods were implemented, one for the face up cards, one for the cards in the player's hand, and one for the route groups.
Also when creating animations, because they run in parallel to the game it was essential to make sure they are synchronized with the the game. For example, disabling face-up cards while the animation is running and making them change color after the first part (swipe) of the animation is done.

SOUNDS

User end: In order to bring the GUI to life we also added sound effects. The set of sounds includes :
- a sound when cards are being drawn or placed,
- one when the player claims a route,
- one when the player draws a ticket or a card,
- one when buttons are clicked,
- one when a message is sent through the chat,
- one when the current player's mouse fly over a claimable route,
- the background music.

Java implementation: In order to implement the sounds we added two classes. The first one, named Sound, represents a sound. The class contains two public methods play() and playOnLoop() that play the sound clip one time or on loop respectively. The second abstract class named Sounds stores as parameters all the sounds used in the game and has static methods to play each sounds. In order for the extension to function the resources folder should contain many .wav files. Source for the background music : https://www.youtube.com/watch?v=bGCKLFsDDVw&t=110s.

CHAT

User End: We wanted the players to have the possibility to simply discuss together through the game itself.

Java implementation: We created two classes, namely RemoteChatProxy and RemoteChatClient that implement both the interface ChatSystem with a sole method 'send(String)', to be overridden. The interface Player was also given a new method 'receiveMessage(String)'. RemoteChatProxy consists of multiple methods :
- addClient(Socket) which adds the new socket's to a list,
- sendAll(String) which call the receiveMessage's function of the player and send the message to each socket's writer of the list,

- startReading(Socket) that starts reading the socket's buffered reader, and call sendAll(String) when a message comes in,
- send(String) that calls sendAll with the serialized message.

Besides send(String), RemoteChatClient has run() that starts reading and use receiveMessage(String).
When ServerMain is launched, it instantiate RemoteChatProxy and wait for connections in a new thread, same goes for ClientMain (whose arguments are the hostname and the proxy for the game, plus the same two for the chat) and RemoteChatClient, in which the call of the method 'run()' is done in a new thread. Finally we created the chat graphical interface in GraphicalPlayer.

LONGEST TRAIL

User End: At the end of the game, we wanted the longest(s) trail(s) to be highlighted by blinking.

Java implementation: We attached a new argument to the PlayerState's constructor, specifically the longestTrail (Trail). If it is not null, then the player's trail is one of longest ones in the game. To add one, the class Game calls the new method in GameState withAddedLongestTrail(PlayerState, Trail) at the end. Then we joined the new serdes for stations and trails in Serdes, and modified publicGameStateSerde as a consequence. Furthermore a new object property was inserted in ObservableGameState for each player, and in MapViewCreator we added a listener to it, that makes all the unpossessed routes turn white, and the longest(s) trail(s) blinking indefinitely using FadeTransition.

PLAYABILITY FROM 2 TO 8 PLAYERS

User End: Playing with only two players becoming quickly boring, the game needed to be fully playable from 2 to 8 players. All that has to be done in order the increase/decrease this number is two change the very first argument when launching the ServerMain with the wanted value (the rest being the names).

Java Implementation: Now, whenever a there's a need to iterate over all the players, it is done on a sublist of PlayerId.ALL, and its size is obtained by looking at the size of the player Names' map given to Game.play. The GUI was modified so as to be suitable with any number of players. The ServerMain opens as many threads as needed. We also adapted the game info received so the longest(s) trail(s) and the amount of points at the end of the game are well formulated.

When we thought we were done with extensions, one thing still problematic : the tCHu map was way too small to be played with entertainment and enjoyment up to more than 3 players. This unpleasant remark led us to cross oceans, and build a new version of tCHu from scratches, based believe it or not in the United States of America.

AMERICAN VERSION OF TCHU

User End: The incredible opportunity to travel throughout the American continent.

Java Implementation: To build the map, we found great inspiration in the game Ticket To Ride - USA ®. We wrote the class UsaMap similarly to ChMap, changing the routes, stations and tickets. The background map was generated from the OpenStreetMap data on the site stamen (http://maps.stamen.com), and reshaped it on a vector drawing software to place and add all the city names. Then we drafted a little JavaFX program to place the 305 rectangles for the routes with the mouse, that generates the style sheet. All what remains to be done is to enjoy thrilling games on this new map. DROPBOX LINK FOR RESOURCES : https://www.dropbox.com/s/hmy9fq316ybf1dj/resources.zip?dl=0.