

Rapport des extensions de tCHu

Elyes Ben Chaabane (330441)

Rasan Younis (329880)

● Extensions primaires

- Fenêtre d'accueil:

Dès que l'utilisateur lance l'application, il aura tout d'abord une fenêtre ouverte qui lui proposera de choisir son pseudonyme (pour les fans de Zelda taper Link). Il pourra ensuite choisir d'être l'hôte de la partie (il indiquera son port d'accès) ou de rejoindre une partie (entrera l'adresse IP et le port du server).

Pour ce faire nous avons créé deux nouvelles classes:

- **LaunchMenu** créant et gérant les différentes fenêtres tout comme le lancement du server ou du client
- **Play** héritant de Application instanciant **LaunchMenu** pour lancer la fenêtre

Pour communiquer les noms des joueurs, nous avons ajouté une méthode dans l'interface **Player** (`getName()`) dans l'intérêt de demander au joueur son nom. La méthode **play() de la classe Game** appelle cette méthode pour chaque joueur avant de les initialiser. Une nouvelle méthode dans **GraphicalPlayer** consistera uniquement en un nouveau gestionnaire de d'action gérant la sélection de nom servant enfin dans **GraphicalPlayerAdapter** à ajouter et retirer d'une queue bloquante le nom de son joueur stocker dans une variable statique dans **LaunchMenu**. L'adaptateur relaie à travers les flux de communication (avec une nouvelle **Messaged**) le nom au server gérant le jeu.

- Carte "Bombe":

Un nouveau type de carte permettant à un joueur de détruire (non-définitivement) une route emparée par l'adversaire (carte assez rare pour pas trop déséquilibrer le jeu)

Nous avons défini une nouvelle **Color (Color.BOMB)** et un nouveau type énuméré dans **Card**. D'autant plus, il était nécessaire de créer dans **PlayerState** et **GameState** des méthodes retournant un nouvel état de joueur complet correspondant à la perte d'une route ou

la destruction d'une route (en fonction du point de vue et un nouvel état de jeu complet. Ceci implique un nouveau **TurnKind** dans *play()* (*Game*) et un nouveau gestionnaire d'action de destruction de routes (modification du comportement d'interaction avec les route sur la carte de la Suisse dans **MapViewCreator** lié à de nouvelles propriétés indiquant pour chaque route dans **ObservableGameState** si le joueur peut la détruire et ajout d'une nouvelle propriété stockant le nouveau gestionnaire d'action dans **GraphicalPlayer**). Bien sûr pour afficher la carte nous avons fait des modifications aux codes css de **colors.css** et **decks.css**.

- **Musique:**

La prochaine extension est un menu de paramètres de sons, laissant à l'utilisateur l'option de choisir à travers 3 chansons, de changer le volume, mettre en pause ou arrêter définitivement la musique actuelle.

Pour ce bonus nous avons uniquement modifié **GraphicalPlayer**, car il n'y a aucune modification au niveau du jeu, mais seulement pour l'interface graphique/sonore. Nous avons étendu la **BorderPane** du jeu avec une **MenuBar** constitué d'un **MenuItem** ouvrant sur un clique une fenêtre additionnelle avec des **RadioButtons** pour la sélection d'une musique à la fois, des **Buttons** pour gérer le déroulement de la musique et un **Slider** lié à la propriétés de sons des musiques. Les musiques sont jouées par des **MediaPlayer**.

- **Extensions secondaires:**

- **Élargissement des cartes faces visible:**

Lors des interactions avec les cartes faces visibles, ces dernières s'agrandissent légèrement et les bords changent de couleurs.

Création d'une nouvelle sous-classe css de la style classe **.card** (**.interact**) avec un nouvel attribut **:hover** (ajouté à la **styleClass** de chaque emplacement de carte)

- **Affichage des noms des routes interagissables:**

_____Étiquette portant le nom des villes constituant la route suivant la souris lorsqu'elle pointe sur la route.

Nous avons créé dans **MapViewCreator** un nouveau **Group** constitué d'un **Rectangle** et d'un **Text**. En plus de cela, nous redéfinissons une **ActionEvent** lorsque nous entrons et quittons le domaine du **Group** activant et désactivant respectivement la propriété de visibilité du **Group**.

- **Texte indiquant le tour d'un joueur:**

Espace de texte apparaissant dans un coin pendant 3 secondes lorsque c'est le tour de l'utilisateur.

Création d'un **Group** similaire à celui de l'extension précédente dont la propriété visibilité est activée à l'aide d'un **ActionListener** par une **ReadOnlyBooleanProperty** de **ObservableGameState** indiquant si l'utilisateur est le joueur actuel. Nous avons également créé un attribut **Timer** dans **MapViewCreator** et définissons une nouvelle tâche **TimerTask** à chaque appel du **ActionListener**.

- **Bouton affichant les points totaux et de tickets actuels:**

Bouton dans la qui, sur un clique, affiche un espace de texte indiquant les points actuels du joueur.

Nous avons ajouté deux nouvelles **ReadOnlyIntegerProperties** dans **ObservableGameState** stockant les **finalPoints()** et les **ticketPoints()** du joueur. Dans **InfoViewCreator**, le contenu de ces propriétés est lié à un **Text** dont la propriété de visibilité du dernier est lié à l'activation d'un **Button**.

Voici ci-joint un lien contenant les fichiers du répertoire de ressources avec les icônes et les musiques:

https://drive.google.com/drive/folders/1UsevIwI49MYLhbE-7wO_jPK7I3iyfT6J?usp=sharing