

Projet LO21 – Schotten-Totten

Objectif

Implementer en C++ une application ‘Desktop’ qui permet de jouer au jeu Schotten-Totten aussi bien dans sa version 1 que 2. Il s’agira en premier lieu d’implémenter la version 1 tout en préparant l’implémentation de la version 2.

Membres

- Justine POUGET
- Axel CAMPAÑA
- Gabriel Santamaria
- Pierre ROUSSEL
- Yicheng WAMG

Lien utiles

- Diagramme de fonctionnalité collaboratif
- Diagramme de classe collaboratif – Premier jet à retravailler

Etude du jeu, de ses versions et de ses concepts

Schotten-Totten_v1

Dans sa version initiale, le Schotten-Totten nous propose une version 1v1 se présentant sous la forme d’un plateau de jeu séparé par une frontière. L’objectif principal pour gagner la partie est de remporter un nombre de manche prédéfini en prenant le contrôle de la frontière. Pour se faire les joueurs vont devoir s’affronter de part et d’autre de la frontière en posant des **combinaisons de 3 cartes** le long de la frontière. L’endroit où les combinaisons s’affrontent est plus communément appelé **Bornes (9 Bornes composent la frontière)**. Le joueur ayant la combinaison la plus forte (sans notion d’ordre) remporte la **Bornes**. La frontière est considérée comme contrôlée dès lors qu’un joueur remporte **5 Bornes quelconques** ou **3 Bornes adjacentes**. Lorsqu’un joueur remporte une manche, il marque alors 5 points et son adversaire remporte autant de points qu’il contrôle de **Bornes**.

Dans cette version, chaque joueur de la partie est soumis aux mêmes règles. Tour à tour, ils vont tenter de former leurs combinaisons pour battre celles de l’adversaire – il est alors possible de considérer chaque joueur comme un attaquant.

Le déroulement du jeu varie en fonction du mode de jeu choisi:

Mode normal En mode normal, pour former ses combinaisons, un joueur dispose de **6 cartes en main**. Ces cartes sont appelées **Cartes Clan** (au nombre de 54). Elles possèdent une **couleur (parmi 6)** ainsi qu’une **valeur (comprise entre 1 et 9)**.

Pour jouer un tour, un joueur doit poser une **Cartes Clan** de son côté de la frontière et le long d'une **Bornes**. Il peut ensuite choisir de **revendiquer** la **Bornes** si sa combinaison est plus forte que celle de l'adversaire ou s'il peut prouver qu'elle sera plus forte – ayant connaissance de sa main et des cartes déjà engagées sur le plateau. Une **Bornes** revendiquée passe alors sous le contrôle du joueur et ne peut plus être jouée. Enfin, il doit piocher une carte dans la pioche pour clore son tour.

Mode tactique En mode tactique, la disposition du plateau de jeu diffère légèrement. En effet, le nombre de cartes en main passe au nombre de 7 (au lieu de 6) et une deuxième pioche ainsi qu'une défausse font leur apparition. Il s'agit de la pioche de **Cartes Tactique** (au nombre de 10). Ces cartes peuvent être regroupées sous forme de trois sous-groupes:

Troupes Elites Ces cartes se jouent comme les **Cartes Clans** et sont au nombre de 4: * Joker (x2) – Elle permet de choisir une carte de couleur et de valeur quelconque. Un joueur ne peut jouer deux fois cette carte durant une manche. Dans le cas où cette carte serait piochée une deuxième fois, il devra la conserver jusqu'à la fin de la manche. * Espion – Elle a pour valeur 7 et une couleur quelconque. * Porte-Bouclier – Elle a la capacité de prendre une valeur comprise entre 1 et 3 et une couleur quelconque.

Modes de Combat Ces cartes se jouent par-dessus une **Bornes** et change les règles de revendication localement à la **Bornes**. Elles sont au nombre de 2: * Collin-Maillard – Elle force la somme des cartes et ne prend plus en compte les combinaisons. * Combat de Boue – Elle force la pose de 4 cartes.

Ruses Ces cartes sont à l'origine de la création de la défausse. Elles se jouent en amont du tour pour en modifier le déroulement. Elles sont au nombre de 4: * Chasseur de Tête – Elle permet au joueur de piocher 3 cartes (de n'importe quelle pioche) et de n'en conserver qu'une en main; les autres cartes repartant au bas de leur pile respective. * Stratège – Elle permet de replacer une carte déjà engagé dans une de ses combinaisons pour la replacer dans une autre. La **Bornes** ne peut être revendiqué pour pouvoir déplacer la carte. * Banshee – Elle permet de supprimer une carte d'une combinaison adverse en la plaçant directement dans la défausse. La **Bornes** ne peut être revendiquée pour pouvoir supprimer la carte. * Traître – Elle permet de subtiliser une carte d'une combinaison adverse et de la placer dans une de ses combinaison. La **Bornes** ne peut être revendiquée pour pouvoir subtiliser la carte.

Note: Aucun des joueurs de la partie n'est autorisé à jouer **2 Cartes Tactique de plus** que l'autre joueur.

Mode expert Le mode expert est un mode de jeu visant à réorganiser le déroulement d'un tour. Dans ce mode le joueur doit revendiquer sa **Bornes**

avant de poser une carte. Ce mode de jeu peu être appliquer aussi bien en mode **normal** qu'en mode **tactique**.

Schotten-Totten_v2

Dans la deuxième version de Schotten-Totten, il s'agit toujours d'un jeu de plateau séparé par une frontière. Cependant dans cette version il ne s'agit plus d'une frontière composée de **Bornes** mais de **Murailles** (**7 Murailles composent la frontière**). Les **Murailles** délimite l'intérieur du château de l'extérieur. Dans cette version du jeu il ne s'agit plus de 2 attaquants qui se disputent la frontière mais d'un attaquant qui tente de prendre le contrôle du chateau et d'un défenseur qui cherche à le défendre. Les **Murailles** ont désormais des états (neuves et endommagés) des propriétés qui vont définir les règles locales à appliquer sur la dites **Murailles**. Les propriétés modifiées peuvent être le nombre de cartes comme les priorités sur les combinaisons gagnantes.

Mode normal En mode normal, pour faire ses combinaisons, un joueur dispose de **6 cartes en main**. Ces cartes sont appelées **Cartes Siège** (au nombre de 60). Elles possèdent une **couleur (parmi 5)** ainsi qu'une **force (comprise entre 0 et 11)**.

Côté attaquant L'objectif de l'attaquant est de revendiquer 4 **Murailles** endommagées ou de revendiquer 2 fois la même **Murailles**. Si cela arrive, la partie prend fin.

Lors de son tour, l'attaquant peut et/ou doit: * Battre en retraite – Cela donne la possibilité à l'attaquant de retirer le(s) carte(s) de son choix de(s) **Murailles** de son choix * Jouer une carte – Sur l'une des **Murailles** de son choix * Piocher une carte * Revendiquer une **Murailles** – Cela permet de prendre le contrôle d'une **Murailles**

Côté défenseur L'objectif du défenseur est de défendre la **Murailles** des attaques de son adversaire en formant de meilleures combinaisons. Le défenseur remporte la manche lorsque la pioche vient à être vide.

Lors de son tour, le défenseur peut et/ou doit: * Jeter un chaudron d'huile – Au nombre de 3, ils permettent de défausser la carte la plus proche de la **Murailles** de son choix * Jouer une carte – sur l'une des **Murailles** de son choix * Piocher une carte

Note: Comme pour la version initiale, lorsque l'adversaire veut revendiquer une **Murailles**, il doit avoir une combinaison plus forte que celle de son adversaire ou être en mesure de prouver à son adversaire que sa combinaison sera plus forte que celle de son adversaire.

Mode tactique Comme pour la version initiale, en mode tactique la disposition du plateau change légèrement. Comme pour la version initiale, les joueurs disposent de 7 cartes en main (au lieu de 6) et une deuxième pioche ainsi qu'une

défausse font leur apparition. Cette nouvelle pioche est elle aussi composée de **Cartes Tactique** (au nombre de 11).

Ces cartes peuvent être regroupées en deux catégories:

Troupes Elite Ces cartes se jouent comme des **Cartes Siège** et sont au nombre de 4: * Joker (x2) – Elle permet de choisir une carte de couleur et de valeur quelconque. * Espion – Elle a pour valeur 7 et une couleur quelconque. * Porte-Bouclier – Elle a la capacité de prendre une valeur comprise entre 1 et 3 et une couleur quelconque.

Actions Ces cartes sont à l’origine de la création de la défausse. Elles se jouent en amont du tour pour en modifier sont déroulement. Elles sont au nombre de 7: * Redéploiement – Cette carte permet de déplacer une carte engagée dans une de ses combinaison sur une autre **Murailles** ou dans la défausse * Recrutement – Cette carte permet de choisir une carte défaussée et de la ré-engager dans une de ses combinaisons * Disertion – Cette carte permet de défausser une carte engagée dans une combinaison adverse * Traîtrise – Cette carte permet de subtiliser une carte engagée dans une combinaison adverse et de la placer dans une de ses combinaisons * Explosion – Cette carte permet de défausser toutes les cartes engagées dans une combinaison le long d’une **Murailles** * Soutient – Cette carte permet de remettre 2 cartes défaussées dans la pioche. Cette carte ne peut être jouer tant que la pioche n’a pas au moins 2 cartes. * Echange – Cette carte permet de choisir 3 cartes dans sa main respective et de les échanger avec l’adversaire

Note: Aucun des joueurs des la partie n’est autorisé à jouer **2 Cartes Tactique de plus** que l’autre joueur.

Combinaisons

Dans ce jeu, la liste des combinaison possible sont les suivantes: 1. Suite couleur – Trois cartes de la même couleur et de trois valeurs successives 2. Brelan – Trois cartes de la même valeur 3. Couleur – Trois cartes de la même couleur 4. Suite – Trois cartes de valeurs successives 5. Somme – Trois cartes quelconques

Les concepts

Que ce soit dans la verion initiale ou revisitée, le jeu propose un concèpte de stratégie où les deux adversaires cherchent à prendre le contrôle d’une frontière. Afin de prendre le contrôle de celles-ci, il s’agit de poser de son côté de la frontière des combinaisons de cartes qui seront meilleures que celles de l’adversaire. Dans la version initiale, les deux deux joueurs peuvent être considés comme des attaquants. Ils sont tous deux soumis aux mêmes règles de déroulement d’un tour.

Dans la deuxième version du jeu les joueurs sont plutôt à considérer comme attaquant/défenseur. Les règles de déroulement d’un tour diffère en fonction du

rôle occupé. Notamment la première phase du tour et la dernière phase.

Il est possible de noter que dans la première version et la deuxième, les tuiles diffèrent dans leur représentation. En effet, les tuiles de la deuxième version possèdent des propriétés pouvant imposer certaines formes de combinaison. Cependant, cette représentation peut se rejoindre car grâce aux **Cartes Tactique** de catégorie **Mode de Combat**, il est possible de donner des propriétés spécifiques aux tuiles de la frontière. Il faudra donc être en mesure de donner une représentation suffisamment large pour gérer cela.

De même, la manière dont le joueur dont le joueur interagit avec le plateau de jeu reste commune, dans les deux versions le joueur doit être capable de poser des cartes sur une frontière, piocher et revendiquer une tuile de la frontière. En fonction du mode de jeu et/ou de la version du jeu, le joueur doit être en mesure de pouvoir déplacer des cartes sur le plateau. Par exemple, en **mode tactique** dans la version initiale les joueurs doivent être capable de déplacer une carte déjà engagée dans une combinaison. Cette carte peut être d'une combinaison du joueur comme de l'adversaire et peut se retrouver dans une combinaison du joueur ou dans la défausse. On retrouve aussi ce concept dans la deuxième version du jeu avec les actions de début de tour. L'attaquant est capable de **Battre en retraite** et le défenseur de **Jeter un chaudron d'huile**.

Pour se focaliser plus sur les cartes, il est nécessaire de prendre en compte que les cartes communes (**Cartes Clans** ou **Cartes Siège**) ont une valeur ainsi qu'une couleur. Pour les **Cartes Tactique**, il faut noter que ces dernières se jouent à des endroits différents et ont des propriétés différentes. Les **Troupes Elite** se jouent de la même manière et ont les mêmes propriétés que les cartes communes à la différence que leurs attributs (valeur et/ou couleur) peuvent être défini. Les **Modes de Combat** modifient les propriétés des tuiles localement. La tuile une fois modifiée peut être assimilé à une **Murailles** dans la deuxième version. Il faut donc prévoir des fonctionnalités qui permettent de modifier les propriétés en fonction des besoins. Enfin, les cartes **Ruses** tout comme les cartes **Actions** se jouent dans la défausse et modifie le déroulement d'un tour ainsi que les actions possible pour un joueur.

Dans chaque version, il faudra être en mesure de comparer deux combinaisons entre elles en fonction des propriétés de la tuile. Il faudra aussi prévoir les règles d'égalité de combinaison qui vont forcer à sommer les valeurs des cartes composant la combinaison. En cas de nouvelle égalité, il faudra pouvoir déterminer quel joueur à compléter en premier sa combinaison.

Structuration et organisation du projet

Le projet sera hébergé sur GitHub et sera suivi via la fonctionnalité **Project**. Le projet sera divisé en **Issues** qui seront assignées aux membres du groupe. Chaque issue suivra le workflow suivant avec ces différents statut: * **To do** – Ce statut défini la liste des tâches toujours à réaliser * **In Progress** – Ce statut défini la liste des tâches en cours de réalisation * **Staging** – Ce statut défini la liste des tâches qui sont réalisées et tester spécifiquement lors de l'implémentation * **Done** – Ce statut défini la liste des tâches finies qui sont

testés et considérés comme fiable Dans le projet, chaque tâche pourra être priorisé grâce à un attribut dans l'issue. Ils pourront aussi être prioriser par le ordre d'apparition dans les **Backlogs**.

Un suivi des issues qui sont assignées sera possible grâce à la view **My issue**. La branche **main** sera la branche principale du projet et servira pour le code considéré comme fiable. La branche **staging** sera celle où les tâche sont réalisées mais pas entièrement testées. Les autres branches seront les branches de developpement et porteront comme nom la clé du ticket (à voir si réalisable).

Code guidelines

- Les noms des variables et des fonctions doit être fait en **Anglais**. Ainsi, si une fonction effectue dans une recherche avec l'algorithme de tri rapide, on l'appellera par exemple **quickSort** au lieu de **triRapide**
- On utilisera la convention de nommage de **camelCase** (voir l'article Wikipédia pour en savoir plus: camelCase)
- On évitera de faire des pyramides de conditions. Par exemple:

```
/// On évite ce genre de structures
int main()
{
    int var = 0;
    if (var) {
        // code
    }
}

/// Qu'on privilégie à ce genre là
int main()
{
    int var = 0;
    if (!var) { return 0; }

    // code
}
```

Conception et architecture du projet

Choix du design pattern

Le design pattern que nous avons retenu pour le projet est le design "Observer".

Fonctionnement du Observer

Le design de l'Observer est particulièrement bien adapté pour le jeu vidéo, puisqu'il est basé sur le principe d'événements. Un objet *publie* un événement

auquel d'autres peuvent *souscrire*.

Lorsqu'un évènement est trigger, tous les objets ayant souscrit à cette évènement reçoivent un signal en plus de certains paramètres, leur permettant d'exécuter une fonction donnée.

Application de l'Observer au jeu de plateau

Pour notre jeu de plateau, chaque éléments dérive d'une classe initiale, la *Base Actor class*. Cette classe a des méthodes qui permettent à tous ses fils de publier, souscrire et déclancher des evènements.

Ainsi, par exemple, la classe **Player** peut publier un évènement **PlayCard** qui passe en paramètres la carte et l'endroit où le joueur veut la jouer. Une autre classe qui controle la logique du jeu peu alors souscrire à cet evènement pour vérifier si le coup est légal.

De même, cette même classe qui contrôle la légalité d'un coup peut très bien publier son propre évènement qui est trigger uniquement lorsque un coup est illégal, envoyant ainsi l'information à toutes les classes y ayant souscrit.

Diagramme de cas d'utilisation

-> Lien

Diagramme de classe

-> Lien