

**GEN**

## **Rapport Projet : Poker**



**Auteurs :** Balestrieri Hakim & Gomes Christian

**Professeur :** Lachaize Patrick

**Date :** 10.06.2020

# Table des matières

---

Introduction.....	3
Règles du jeu « Poker Texas Hold'em » .....	4
Organisation .....	4
Début d'une partie .....	4
Combinaison de cartes .....	6
Adaptation du jeu.....	7
Gestion de projet.....	8
Sprint 1 .....	8
Sprint 2 .....	8
Sprint 3 .....	9
Sprint 4 .....	9
Interface graphique.....	10
Connexion.....	10
Inscription.....	11
Table du joueur .....	12
Modèle de domaine .....	12
Diagrammes .....	13
Activité – Connexion / Inscriptions .....	13
Etat – Phase de la partie.....	14
Difficultés rencontrées.....	15
Conclusion .....	15

## Introduction

---

Dans le cadre de la deuxième année concernant le cours de GEN, nous devons réaliser un projet qui comporte différents éléments du cours afin de les mettre en pratique. Notre choix s'est porté sur le jeu d'un « Poker » à deux cartes.

Ce choix nous permet de mettre en pratique :

- La programmation réseau & multi-threads
- L'utilisation de GitHub
- La gestion de projet avec IceScrum
- Les tests unitaires ainsi que Travis
- Les différents diagrammes

Le projet a été divisé en 4 sprints avec à chaque fois un entretien avec le « Project Owner » pour la validation des étapes.

Les dates des sprints ont été les suivantes :

- Sprint 1 : Du 1<sup>er</sup> Avril au 23 Avril 2020
- Sprint 2 : Du 24 Avril au 7 Mai 2020
- Sprint 3 : Du 8 Mai au 21 Mai 2020
- Sprint 4 : Du 22 Mai au 11 Juin 2020

Le choix de notre projet sur ce laps de temps n'est pas l'aisée au niveau de sa complexité, niveau interaction et jouabilité. L'objectif ici, est de s'approcher au maximum de la jouabilité « classique » d'un jeu de Poker afin d'en obtenir une version finale à la fin du projet.

## Règles du jeu « Poker Texas Hold'em »

---

Pour jouer à un jeu de Poker entre amis ou officiellement, il nous faut le strict minimum qui est :

- Un jeu de 52 cartes
- 2 joueurs minimum
- Des jetons afin de miser (différentes valeurs)

### Organisation

*« Pour commencer, vous devez désigner un donneur, afin de le distinguer des autres, le bouton doit être placé sur la table à côté de ce joueur. Le donneur est en charge de distribuer les cartes. A chaque tour, le donneur change dans le sens des aiguilles d'une montre. »*

*Les deux joueurs situés à la gauche du donneur sont appelés petite blinde et grosse blinde. Ces deux joueurs sont obligés de miser une somme définie au départ entre l'ensemble des joueurs, cela avant même de distribuer les cartes.*

*Les joueurs qui sont petite blinde et grosse blinde change en même temps que le donneur, c'est-à-dire à chaque manche.*

*Les sommes mises pour la petite blinde et la grosse blinde augmentent au cours du jeu. Par exemple les joueurs peuvent se mettre d'accord pour augmenter les mises toutes les demi-heures.*

*Exemple : Selon la valeur que vous donnez pour chaque jeton, vous pouvez commencer par 5 et 10 pour la petite et grosse blinde, puis au bout d'une demi-heure, passer à 10 et 20. »*

### Début d'une partie

*« Le donneur distribue à chaque joueur deux cartes, une par une. »*

*Le premier tour de mises démarre par le joueur immédiatement à la gauche de la grosse blinde. Ce joueur, en fonction des cartes qu'il a en main, peut :*

- **Se coucher.** Il jette alors ses cartes et se retire définitivement de la partie.
- **Suivre la grosse blinde,** c'est-à-dire qu'il met dans le pot la même somme que celle déposée par la grosse blinde.
- **Relancer.** Il mise alors une somme supérieure à celle déposée par la grosse blinde.

*Une fois que le premier joueur a pris sa décision, c'est au tour du joueur qui se trouve à sa gauche et ainsi de suite. La petite blinde et la grosse blinde peuvent soit s'aligner sur les mises des autres joueurs, soit relancer. Si relance il y a, un deuxième tour de mises s'effectue alors.*

*Lorsque le premier tour de mises est terminé, trois cartes sont retournées au centre de la table par le donneur. Ces cartes sont appelées le Flop.*

*A savoir : le donneur doit « griller » une carte à chaque fois qu'il doit poser des cartes, c'est-à-dire qu'il ne conserve pas la première carte du talon.*

*Le deuxième tour de mises démarre avec le joueur à gauche du donneur. Dans ce nouveau tour chaque joueur a la possibilité de dire :*

- **Check**, le joueur peut checker c'est-à-dire qu'il ne mise pas de somme supplémentaire dans le pot. Il peut checker seulement si les joueurs d'avant n'ont pas misé eux non plus.
- **Miser**, il décide de mettre dans le pot la somme qu'il souhaite.
- **Relancer**, c'est-à-dire miser une somme plus importante si un joueur a déjà misé.
- **Suivre**, si un joueur a déjà misé, vous égalisé cette mise dans le pot.
- **Se coucher**. Il jette alors ses cartes et se retire définitivement de la partie.

*Lorsque le deuxième tour de mises est terminé, une quatrième carte commune est retournée au centre de la table. Cette carte est appelée le Tournant.*

*Le troisième tour de mises peut commencer comme le précédent.*

*Lorsque ce tour est terminé, une cinquième carte commune appelée la Rivière est retournée au centre de la table. Une fois que les joueurs ont fini de miser, ceux qui sont toujours dans la course (c'est-à-dire les joueurs qui ne se sont pas couchés) montrent leurs cartes. C'est le joueur qui a la combinaison de cartes la plus importante qui gagne la manche, il remporte alors l'ensemble du pot.*

*Une seconde manche peut alors débiter.*

Les règles ci-dessous ont été reprises du site : <https://www.regles-de-jeux.com/regle-du-poker/>. Les adaptations pour notre projet seront expliqués au chapitre « Adaptation des règles ».

## Combinaison de cartes

Ci-dessous une image représentant les différentes combinaisons possibles au jeu du poker. Dans notre cas, entre les deux cartes en main d'un joueur et les 5 cartes de la table.



### **Royal Quinte Flush** ( Suite couleur royale ) :

La Royal Quinte Flush est composée de 5 cartes consécutives de la même couleur, partant de l'As.

Ex: As♥ - K♥ - Q♥ - J♥ - 10♥

C'est la main la plus forte au poker. Vous êtes imbattable !



### **Straight Flush** ( Suite couleur ) :

est composée d'une suite de 5 cartes de la même couleur mais ne partant pas de l'As. Ex: 10♦ - 9♦ - 8♦ - 7♦ - 6♦

C'est la 2ème main la plus forte. Mais peut être battu par une suite couleur plus forte.



### **Four of a Kind** ( Carré ) :

Le Carré est composée de 4 cartes identiques.

Exemple : A-A-A-A-K

C'est la 3ème meilleure main au poker.



### **Full House** ( Full ) :

Le Full est composée de 3 cartes identiques accompagnées de 2 cartes identiques ( un brelan + une paire )

Exemple : K-K-K-10-10



### **Flush** ( Couleur ) :

est composée de 5 cartes de la même famille et couleur mais qui ne se suivent pas.

Exemple : 8♣, 5♣, J♣, 3♣, 6♣



### **Straight** ( Suite ) :

La suite est composée de 5 cartes consécutives, autrement dit qui se suivent, mais de couleurs différentes :

Exemple : Q - J - 10 - 9 - 8



### **Three of a Kind** ( Brelan ) :

Le brelan est une main composée de trois cartes de même valeur. Si deux joueurs possèdent un brelan, le gagnant est celui qui possède le brelan le plus élevé. Par exemple, Q-Q-Q-T-4 bat 5-5-5-Q-2 puisque la dame est supérieure au cinq.



### **Two Pair** ( Double paire ) :

Cette main est formée de deux paires différentes. Si deux joueurs possèdent deux paires, on considère la paire la plus haute pour déterminer le gagnant. Par exemple, K-K-10-10-5 bat 8-8-2-2-J puisque le Roi est plus fort que le huit.



### **One Pair** ( La paire ) :

Une paire est formée de deux cartes identiques, de même valeur. Dans notre exemple : 2 Valets. Si deux joueurs ont une même paire, on considérera alors une troisième carte. Par exemple, J-J-K-10 bat J-J-9-8-10 puisque le roi est plus fort que le neuf.



### **High Card** ( La hauteur ) :

Si vous n'avez aucun jeu, c'est la carte la plus forte de votre jeu qui compte. Vous avez une carte haute à l'As si votre jeu comporte un As.

Figure 1 <https://kingofpoker.org/wp-content/uploads/2014/10/mains-au-poker.png>

## Adaptation du jeu

---

La première approche au cours du projet a été de s'approcher le plus possible des règles de jeux classiques. Au fil du projet et à cause de la complexité ou encore du temps disponible pour toute l'implémentation et la sortie d'une release finale. Il a fallu adapter certaines règles qui génèrent une approche « à notre sauce » mais toujours dans le sens de la même jouabilité.

Les modifications suivantes ont été apportés :

### **Gestion des tours**

Dans un jeu de poker classique, chaque joueur peut accéder au tour suivant uniquement si celui-ci :

- 1) N'a eu aucun joueur qui le précède effectuer une mise.
- 2) Suit la mise du joueur qui le précède.

Ce qui fait que si un joueur a eu un autre joueur qui le précède miser et que le joueur actuel remise par-dessus, le tour n'est pas terminé bien que les deux joueurs aient joué.

Ceci n'est pas implémenté dans notre jeu, un joueur ne peut uniquement jouer qu'une seule fois par phase et n'est pas obligé de suivre la mise d'un joueur qui le précède.

### **AS : carte la plus faible**

Dans notre implémentation, l'As sera la carte la plus faible comme un « 1 » et non la carte la plus haute comme dans un jeu classique.

### **Quinte flush royale supprimée**

La quinte flush royale n'est pas implémentée dans notre version.

### **Gestion des blindes**

La gestion concernant les blindes <sup>1</sup> a été omise car cela correspondait à une charge de travail élevée par rapport au temps mis à disposition.

---

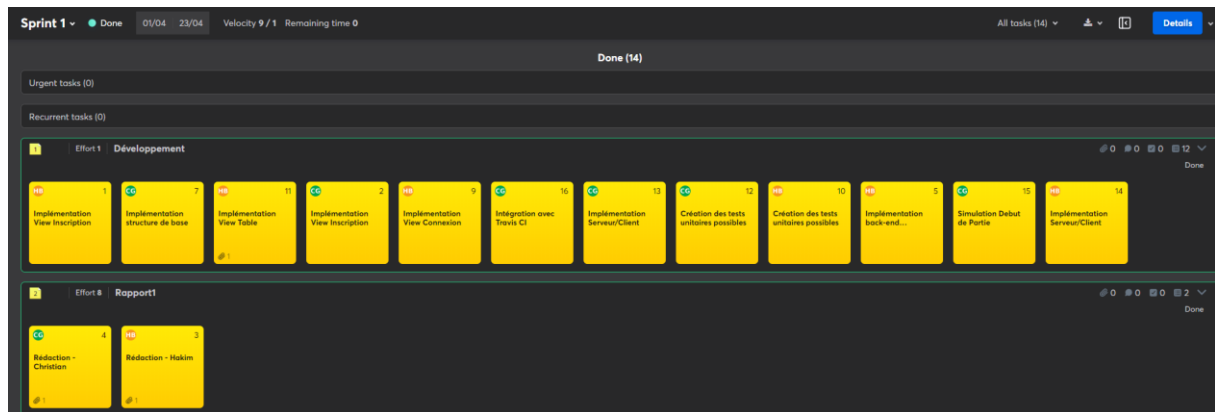
<sup>1</sup> Taxe prise à chaque joueur lors du début d'une partie. Une personne est la big blind (taxée à double par rapport à la small blind) et le reste des joueurs sont les small blind. Cela change à chaque partie.

## Gestion de projet

Dans le cadre de ce projet nous avons utilisé l'outil IceScrum. Celui-ci nous a permis de planifier diverses tâches réparties dans des « **Sprint** ».

Un « **Sprint** » est un ensemble de tâches découpées du projet. Celui-ci contient des « **Stories** » qui regroupent elles-mêmes un ensemble de tâches qui concernent une fonctionnalité du projet.

### Exemple d'un sprint



Les « post-it » jaunes sont des **tâches**. On peut voir que celles du haut sont rattachées à la **story** « Développement » et le tout est lié au **sprint** « Sprint 1 ».

## Sprint 1

Concernant le premier sprint, nous nous sommes principalement concentrés sur la base du projet. C'est-à-dire avoir un environnement d'application fonctionnel graphique qui tourne. Nous avons donc créé les deux GUI principales lors de cette phase (GUI de la table de jeu et GUI de connexion).

Travis CI a été implémenté afin de pouvoir profiter d'un outil d'intégration continue, permettant ainsi de facilement lancer les tests de notre application.

Nous avons notamment construit le squelette principal du rapport qui a été utilisé lors des différents rendus.

Le logo a lui aussi été créé durant cette phase.

## Sprint 2

Durant le deuxième sprint, nous avons établis un diagramme de communication ainsi qu'un diagramme de séquence. La GUI s'est vue modifiée afin de permettre une meilleure visualisation des joueurs et de leurs cartes respectives. On y implémenta également le fait de pouvoir miser / se coucher pour chaque joueur.



L'introduction de la communication Client- Serveur fût la tâche prenant le plus de temps. Celle-ci nous a permis de mettre en place une structure servant d'exemple afin de pouvoir ré—implémenter d'autre communications nécessaires au jeu (tel que le fait que les cartes du **Board**<sup>2</sup> doivent êtres similaires pour tout le monde).

Afin de permettre une communication synchronisée entre le client et le serveur (tout ceci pendant que le jeu se déroule) nous avons mis en place différents **Thread**<sup>3</sup> comme nous l'avons vu durant les différents cours de Génie logiciel.

Une première partie a pu être simulée (manuellement) afin de pouvoir visualiser les éventuels bogues / anomalies.

## Sprint 3

Lors du sprint 3, nous avons établis un diagramme d'activité.

Une gestion des différents états de la partie a notamment été introduit (reconnaissance de la phase actuelle : Flop<sup>4</sup>, River<sup>5</sup>, Turn<sup>6</sup>). Nous avons notamment revu l'ensemble des actions possibles joueurs afin de pouvoir correctement les interprétés entre le back-end et le front-end. Ainsi un joueur pourra uniquement miser lors d'une phase de mise (qui est affichée sur la fenêtre).

Un pot commun aux joueurs a été implémenté. Celui-ci est doit être remporté par le gagnant de la partie.

## Sprint 4

Durant ce dernier sprint, nous avons ajouté diverses fonctionnalités. Le calcul de la force d'une main a été une tâche demandant beaucoup de temps d'analyse et d'implémentation.

Afin d'observer le déroulement d'une multitude de parties, nous avons fait en sorte qu'à la fin d'une partie, une nouvelle en est lancée. Cela nous a été utile car certaines fois il y eut des soucis indécélables (quint flush royale qui n'est pas prise en compte) si nous nous étions contentés de tester des parties manuellement. Ceci aurait été indétectable dans le sens où, il y a très peu de chance d'en avoir une. Le fait d'avoir pu automatiser cela nous a permis de jouer plus de parties.

Un modèle de domaine a été créé afin de permettre une meilleure visualisation du projet.

Pour finir, une rédaction finale du rapport fût effectuée.

---

<sup>2</sup> Cartes communes à tous les joueurs. Elles se situent au milieu de la table. Il y en a cinq au total.

<sup>3</sup> Un thread est une séquence d'instructions qui s'exécutent parallèlement à d'autres thread. Ceci est utile afin de permettre plusieurs actions simultanées.

<sup>4</sup> Trois premières cartes du board

<sup>5</sup> Quatrième carte du board

<sup>6</sup> Cinquième carte du board

## Interface graphique

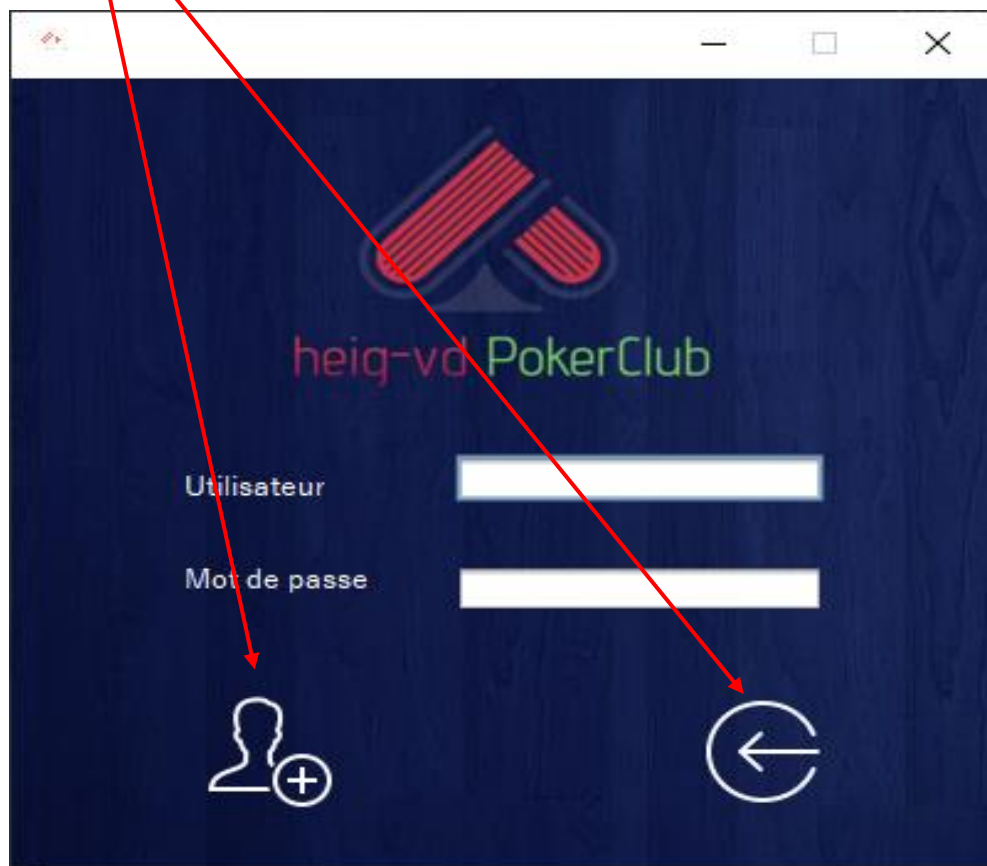
---

### Connexion

Notre interface de connexion dispose de deux emplacements afin d'indiquer son nom d'utilisateur ainsi que son mot de passe pour se connecter.

Deux boutons sont à disposition :

- Connexion
- Inscription



## Inscription

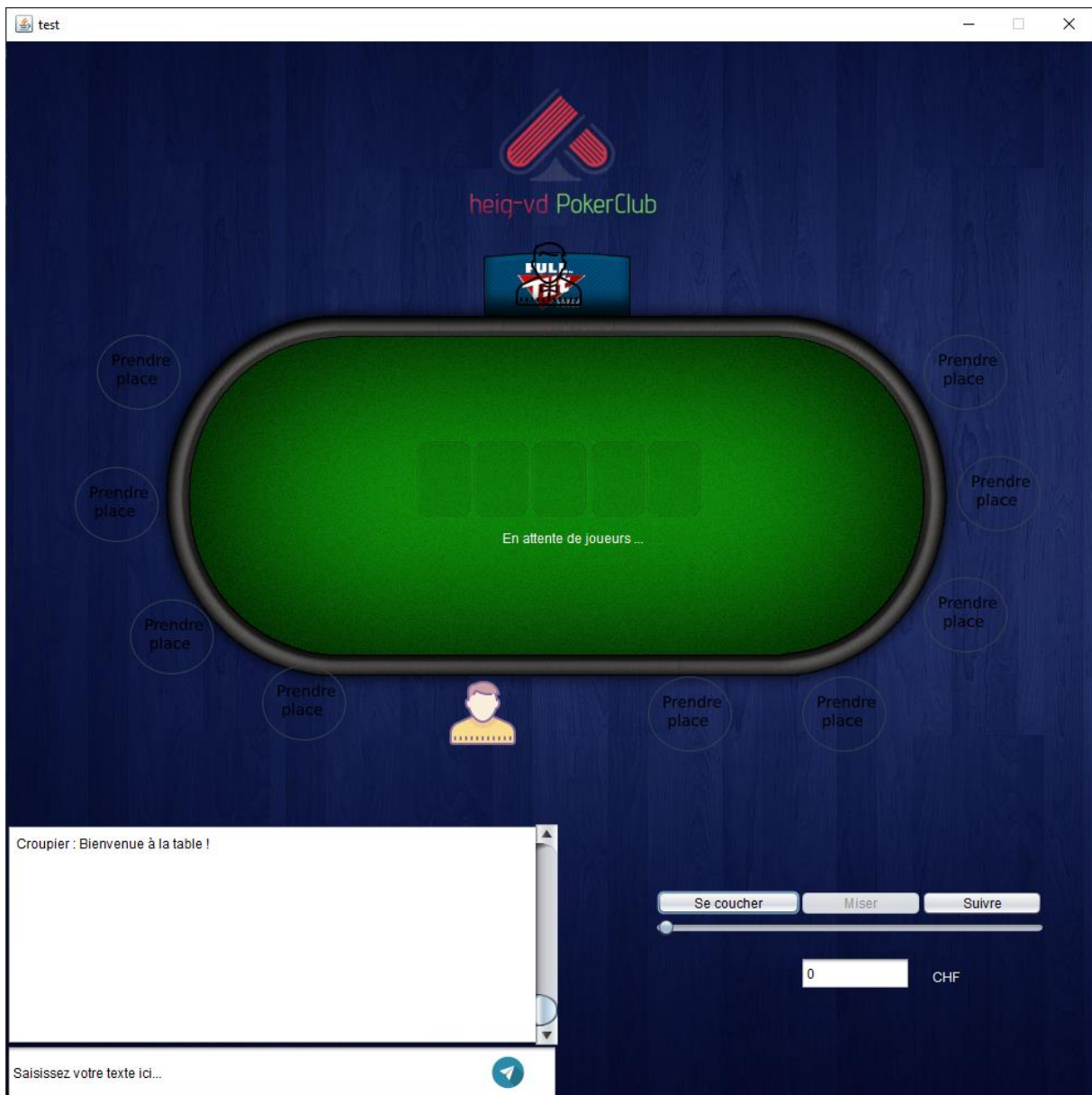
La fenêtre pour l'inscription comporte différents éléments utiles à l'inscription d'un joueur. S'il y a une erreur (âge, information obligatoire ou encore les conditions) un pop-up avertira l'utilisateur.



The image shows a screenshot of a web application window titled "Registration Form". The window has a dark blue background and contains the following elements:

- Form Title:** "Formulaire d'enregistrement" in white text.
- Fields:**
  - Nom d'utilisateur:** A text input field.
  - Mot de passe:** A text input field.
  - Sexe:** Two radio buttons labeled "Homme" (selected) and "Femme".
  - Naissance:** Three dropdown menus for day, month, and year, currently showing "01", "01", and "2020".
  - Adresse:** A large text area.
- Checkbox:** A checkbox labeled "Accepter les conditions générales d'utilisation".
- Buttons:** Two buttons at the bottom: "Envoyer" and "Effacer".

## Table du joueur



Voici l'interface principale de notre jeu de Poker.

Les différentes interactions possibles sont :

- Au début d'une partie, les utilisateurs peuvent s'asseoir à différentes places autour de la table. Une fois qu'il y a 3 personnes (configurable) prêtes à jouer, la partie se lance.
- Lors des phases de mise, chaque utilisateur peut se coucher ou alors miser / suivre.
- Chaque utilisateur peut notamment communiquer avec les autres personnes présentes dans la partie au moyen du chat présent en bas à gauche.

## Modèle de domaine

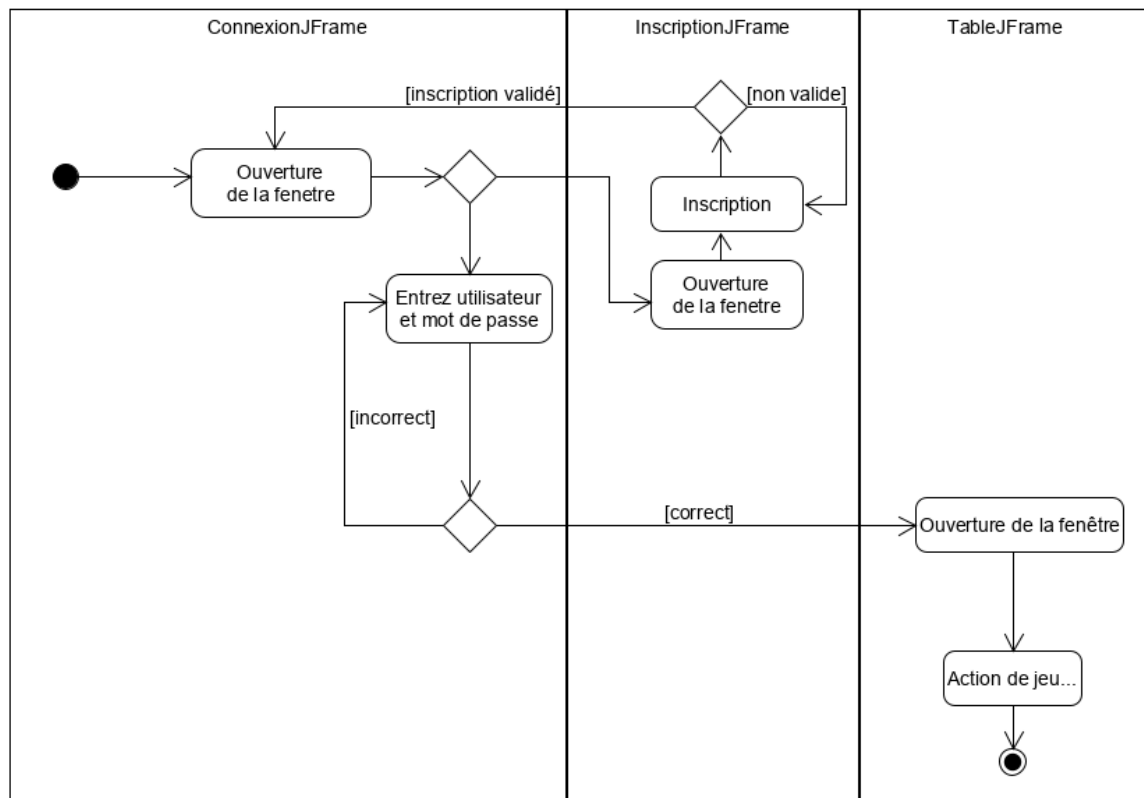
[Voir annexe](#)

## Diagrammes

Application de certains diagrammes vus en cours qui peuvent convenir par rapport à notre projet pour comprendre certaines parties.

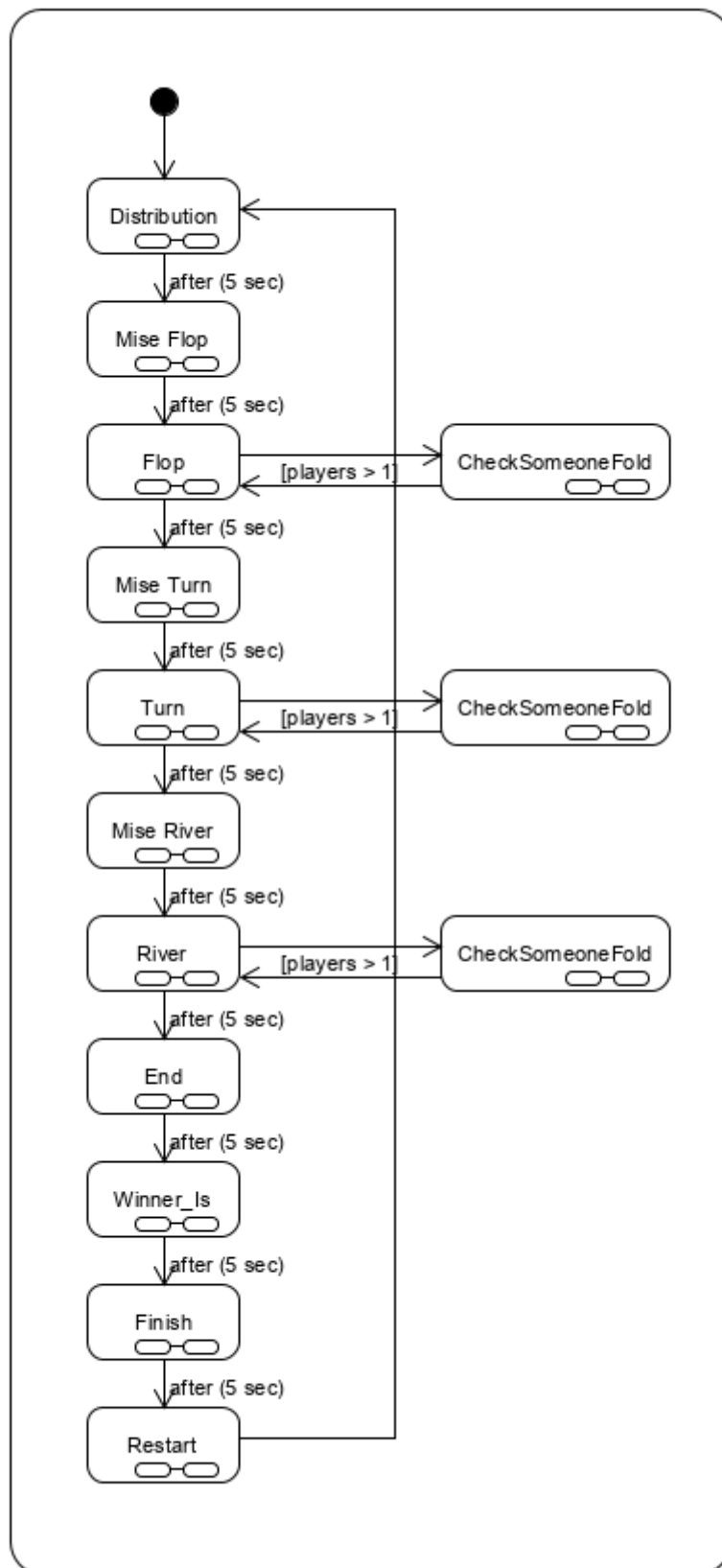
### Activité – Connexion / Inscriptions

Un diagramme d'activité qui démontre la connexion d'un joueur via sa fenêtre d'interaction.



**Remarque :** « Action de jeu... » représente la suite qui n'est pas démontré par ce diagramme.

## Etat – Phase de la partie



## Difficultés rencontrées

---

Les difficultés principales ont été la programmation couplée aux différents threads que devait comporter notre application. Ce fût pour nous une première expérience qui s'est avérée plus ardue qu'une simple application de communication réseau via un chat comme on l'a déjà lors des précédents laboratoires. Nous étions en difficulté durant pas mal de temps. Du coup nous avons eu du retard sur nos tâches. Le résultat de ceci donne une application qui marche avec moins de fonctionnalités.

## Conclusion

---

Cette fin de projet se conclue avec une release non fonctionnelle à 100%, ceci dû aux difficultés rencontrées tout du long. Ce projet fût le travail le plus long et complexe que l'on dû réaliser jusque-là au sein de notre parcours à la HEIG-VD. Le choix de celui-ci s'est porté sur un jeu auquel nous avons déjà joué et avons donc connaissance des règles. En phase de développement, nous nous sommes rendu compte de la complexité de celui-ci pour réussir à faire une version adaptée au jeu classique mais toujours dans le même thème. Il a été intéressant de pouvoir appliquer la matière vue en cours et ainsi mettre en pratique les différents diagrammes.

C'est de ce point de vue que nous remarquons qu'en phase pratique, tous les diagrammes ne sont pas forcément applicables aux différentes implémentations dans un cadre réel. Nous pensons que la partie la plus complexe pour nous a été, la programmation réseau et la gestion des threads. Au fil des ajouts de fonctionnalités, il faut pouvoir gérer différentes choses et avec le peu d'expérience sur la matière, il nous a été difficile de toujours pouvoir déboguer cela à temps.

La gestion de projet a également été une phase intéressante avec la découverte de IceScrum pour la planification de celui-ci.

Nous retenons de ce projet - même non abouti -, une grande satisfaction au résultat de celui-ci.