

Table De Matière

I/ INTRODUCTION.....	2
II/ PHASE D'ANALYSE :	3
1) <i>Diagramme cas d'utilisation :</i>	<i>3</i>
a) <i>Parties Rapides :</i>	<i>3</i>
b) <i>Règles Avancées :</i>	<i>5</i>
2) <i>Diagramme class final :</i>	<i>6</i>
a) <i>Bloc Game:.....</i>	<i>6</i>
b) <i>Bloc Carte:.....</i>	<i>8</i>
c) <i>Bloc Joueur:.....</i>	<i>10</i>
d) <i>L'association entre 3 blocs:</i>	<i>12</i>
e) <i>Diagramme class complet final:.....</i>	<i>13</i>
3) <i>Patron de conception "Strategy":</i>	<i>14</i>
4) <i>Diagramme de séquence :</i>	<i>15</i>
a) <i>Parties Rapides :</i>	<i>15</i>
b) <i>Règles Avancées :</i>	<i>17</i>
III/ ETAT ACTUEL DE L'APPLICATION :	19
1) <i>Interface de la console</i>	<i>19</i>
a) <i>Parties Rapides:.....</i>	<i>20</i>
b) <i>Règles Avancées:.....</i>	<i>22</i>
2) <i>Interface graphique :</i>	<i>24</i>
a) <i>Le pattern MVC.....</i>	<i>24</i>
b) <i>Interface graphique</i>	<i>24</i>
IV/ DIFFICULTES RENCONTRES :	29
V / CONCLUSION :	30

I/ INTRODUCTION

Dans le cadre de l'UV L002, nous devons créer une version électronique du jeu de cartes « Menhirs ».

Le projet est divisé en trois parties qui forment à terme un seul programme :

- La première partie consiste à la création de la modélisation UML initiale (les diagrammes de cas d'utilisation, de classes et de séquence) qui nous ont permis de centrer les difficultés et de mieux comprendre le jeu.
- Dans la seconde partie, nous devons créer un moteur du jeu "Menhirs" par le codage Java et il est fonctionné et utilisé en lignes de commandes.
- La troisième partie complète le jeu en développant une interface graphique minimaliste en utilisant un MVC pour un scénario dans lequel un joueur physique joue contre des joueurs virtuels.

Grâce aux informations fournies dans le cahier des charges, nous avons utilisé l'outil UML et le langage Java afin de produire une application qui est utilisable dans la réalité.

Je souhaite remercier mon professeur Monsieur Stéphane Lebonvallet, le responsable de notre projet qui m'a donné beaucoup d'enseignements très utiles pour la rédaction de ce cahier des charges ainsi que l'application.

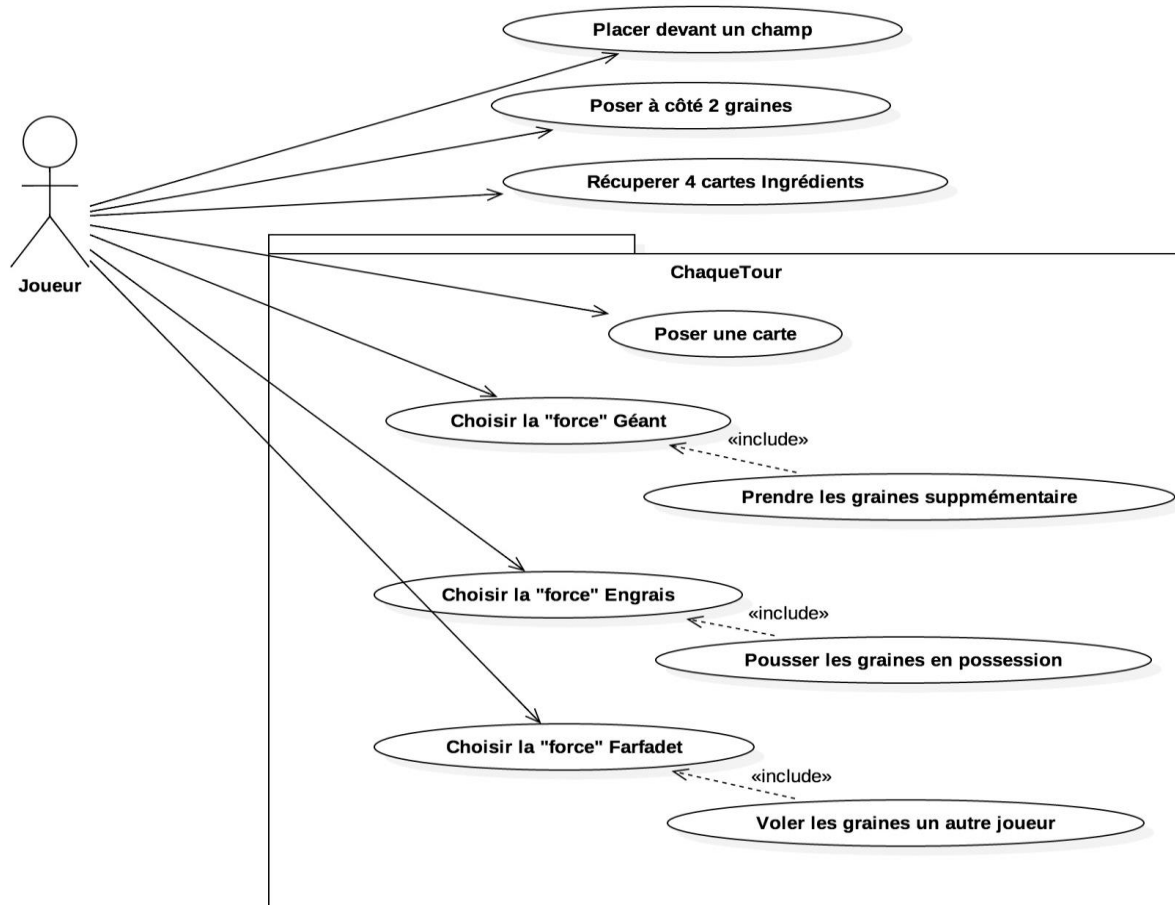
II/ PHASE D'ANALYSE :

1) Diagramme cas d'utilisation :

Cette partie est consacrée à définir quelles actions vont pouvoir effectuées par les joueurs de ce jeu. A partir des cas d'utilisation, nous avons choisi le joueur du jeu comme l'acteur principal.

Dans ce jeu, il y a 2 types différents : parties rapides et règles avancées. Donc, voici deux diagrammes de cas d'utilisation de chaque type de jouer de ce jeu.

a) Parties Rapides :

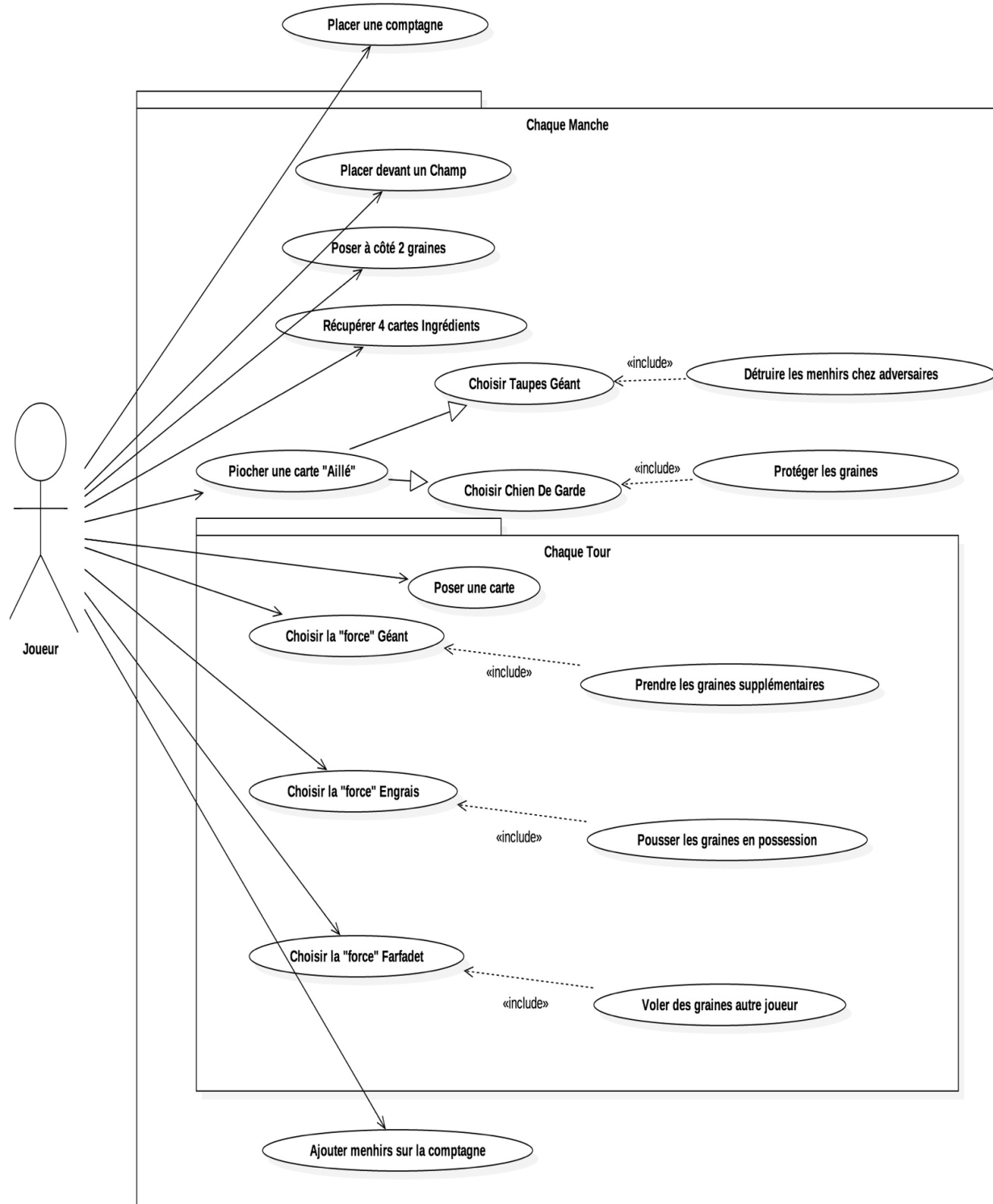


Tout d'abord, chaque joueur a besoin d'un champ et deux graines. Donc, le joueur doit placer devant lui un champ et deux graines à côté d'un champ. Une partie se déroule en quatre tours de jeu (printemps, été, automne, hiver). Avant commencer le premier tour, chaque joueur récupère quatre cartes Ingrédients (les cartes avec un Géant au verso) qui sont distribués par le système. À tour de rôle, les joueurs posent une carte et choisissent l'action à effectuer parmi les trois possibles (Géant, Engrais, Farfadet).

- Si on choisit la "force" Géant Gardien de la Montagne, on peut prendre les graines supplémentaires dans le pot commun (le nombre de graines est indiqué sur la carte).
- Si on choisit la "force" Farfadets Chapardeurs, on peut voler les graines d'un autre joueur. Comme la force Engrais, si le joueur que l'on chaparde a moins de graines que la valeur de la carte, on prend tous ses graines, l'excédent.
- Si on choisit la "force" Engrais Magique, on peut pousser les graines de menhir qui est indiqué ou moins si on n'en a pas suffisamment en son possession.

Donc, dans ce cas d'utilisation, nous utilisons <<include>>. Cela signifie que si l'on choisit la force Géant, on prendra les graines supplémentaires automatiquement. Ce sont les même avec la force Engrais et la force Farfadet.

b) Règles Avancées :



Ce type de jouer utilise toutes les cartes du jeu (les cartes avec Géant au verso et les cartes avec Farfadet au verso). Il est constitué de plusieurs manche, il y a autant de manches que de joueurs. Chaque manche se joue comme une partie rapide. Donc, nous prenons le cas d'utilisation de "parties rapides" et ajouter les informations supplémentaires comme les cartes Alliés ou la comptage des points.

Au début de chaque manche, les joueurs choisissent : soit de prendre 2 graines (comme "parties rapides") soit de piocher une carte Allié. Donc, nous ajoutons le cas d'utilisation "Piocher une carte Allié". S'ils choisissent une carte Allié, ils devront choisir soit "Taupés Géant" soit "Chien de Garde". Comme la partie rapide, nous utilisons aussi <<include>> avec les cartes Alliés. Cela signifie que:

- Si on choisit la carte "Taupés Géant", elle détruit les menhirs adultes chez un adversaire automatiquement.
- Si on choisit la carte "Chien de Garde", elle protège automatiquement les graines contre les Farfadet Chapardeur.

Dans le type "Règles Avancées", il y a une autre carte qui appelé "comptage de points".

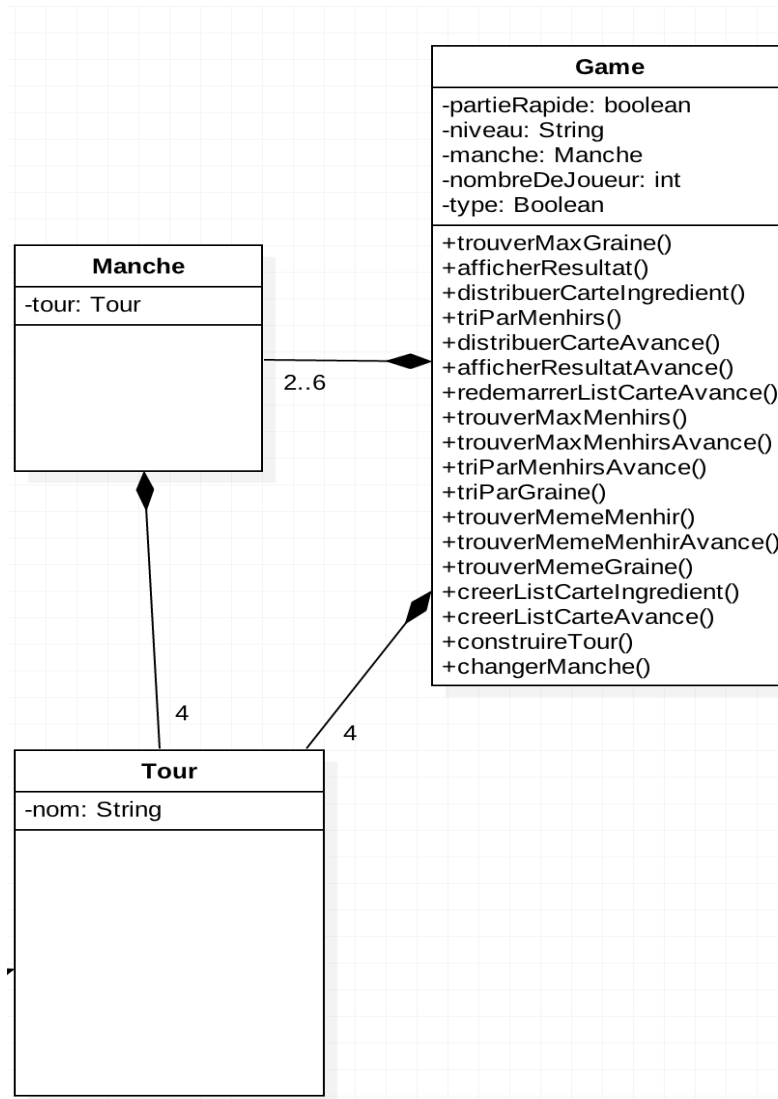
Donc, chaque joueur doit la placer près de son champ. Et à la fin de chaque manche, il doit aussi ajouter le nombre de menhirs sur la comptage de points. Donc, nous ajoutons "Placer une comptage" et "Ajouter menhirs sur la comptage".

2) Diagramme class final :

Ce diagramme regroupe toutes les classes du système ainsi que les relations qui existent entre elles. Mais pour faciliter de gérer le système, nous avons divisé en trois grands blocs.

a) Bloc Game:

Rapport de L002 : Jeu de Menhirs



- La classe Tour contient à un attribut correspondant au nom de la tour (printemps, été, automne, hiver).
- La classe Manche a un seul attribut "tour" de type Tour. Une manche est constitué quatre tours.
- La classe "Game" a cinq attributs. Le premier qui s'appelle "partieRapide" est une variable de booléen pour vérifier le "game" est "parties rapides" ou "règles avancées". Ensuite, le deuxième est niveau. Il est constitué à "Débutant" et "Avancé". Le troisième attribut corresponde aux nombres de joueurs. Le quatrième attribut correspondent à la classe Manche et nous juste utilisons cet attribut quand ce game est

Rapport de L002 : Jeu de Menhirs

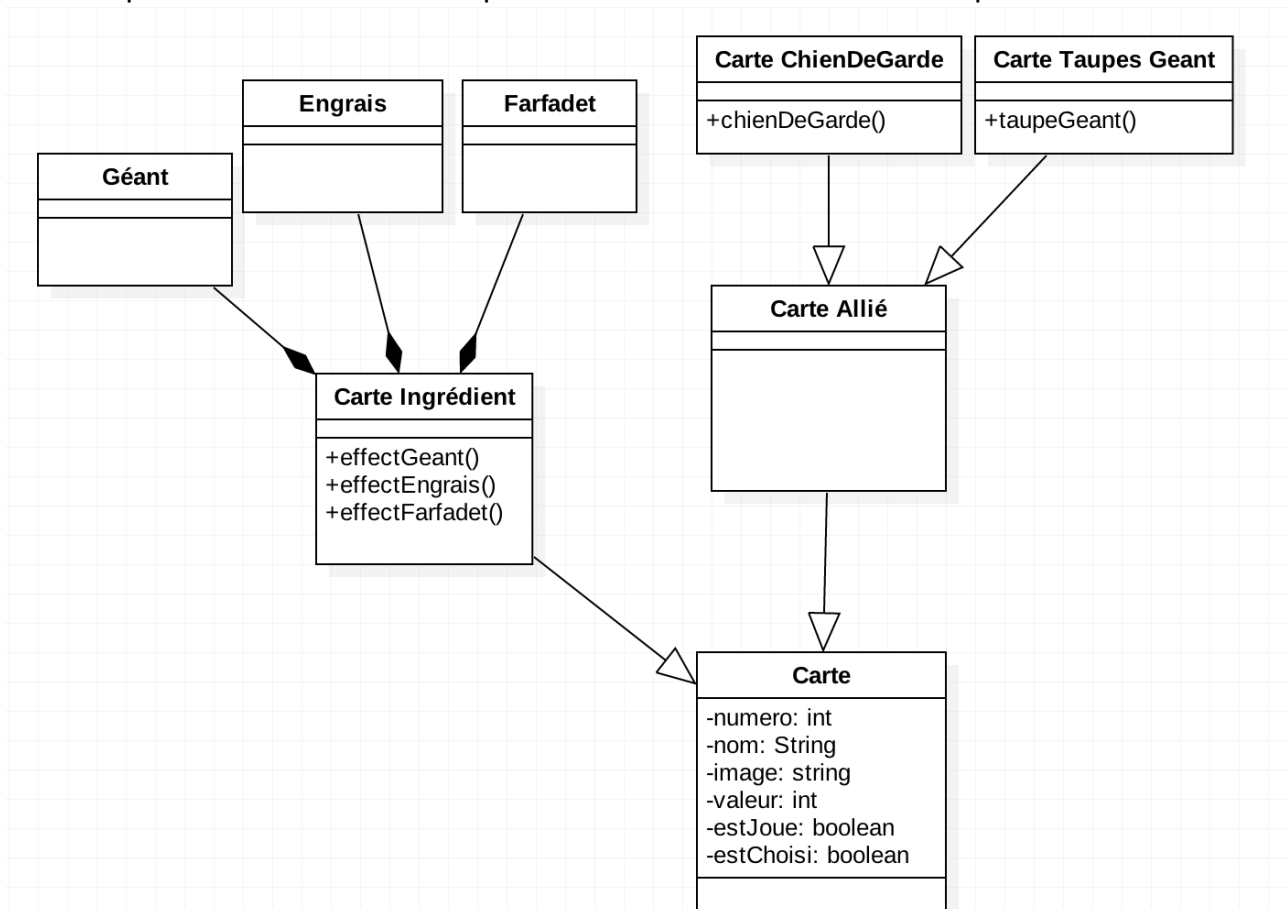
un partie rapide. Le dernier attribut correspondent à la class Tour (on trouve aussi une relation de composition). En outre, la classe Game a 18 méthodes avec ses objectifs qui sont indiqués avec les commentaires dans le code.

De plus, on peut voir que la “Tour” est composée dans la “Game” et aussi dans la “Manche”. C’est parce que on a deux types de jouer. Si on est sur “Parties Rapides” la “Tour” va composer directement dans la “Game” mais sinon la “Tour va composer dans la “Manche” et la “Manche” va composer dans la “Game”.

Ensuite, nous parlons sur la multiplicité. Comme l’énoncé, on peut concevoir que un game ou un manche de jeu va dérouler dans quatre tours qui correspondent à quatre saisons. En dehors, un “game” a de deux à six joueurs et le manche est égal au joueur donc on peut réaliser que un “game” a de deux à six manches.

b) Bloc Carte:

- La classe Carte a six attributs. Trois premiers attributs représentent le numéro, le nom et l’image de la Carte. Le quatrième attribut est le valeur de la Carte (il dépend de la force et du saison). L’attribut “estJoué” de type booléen vérifie la carte est jouée ou pas. Le dernier attribut représente une carte a été choisi ou pas.



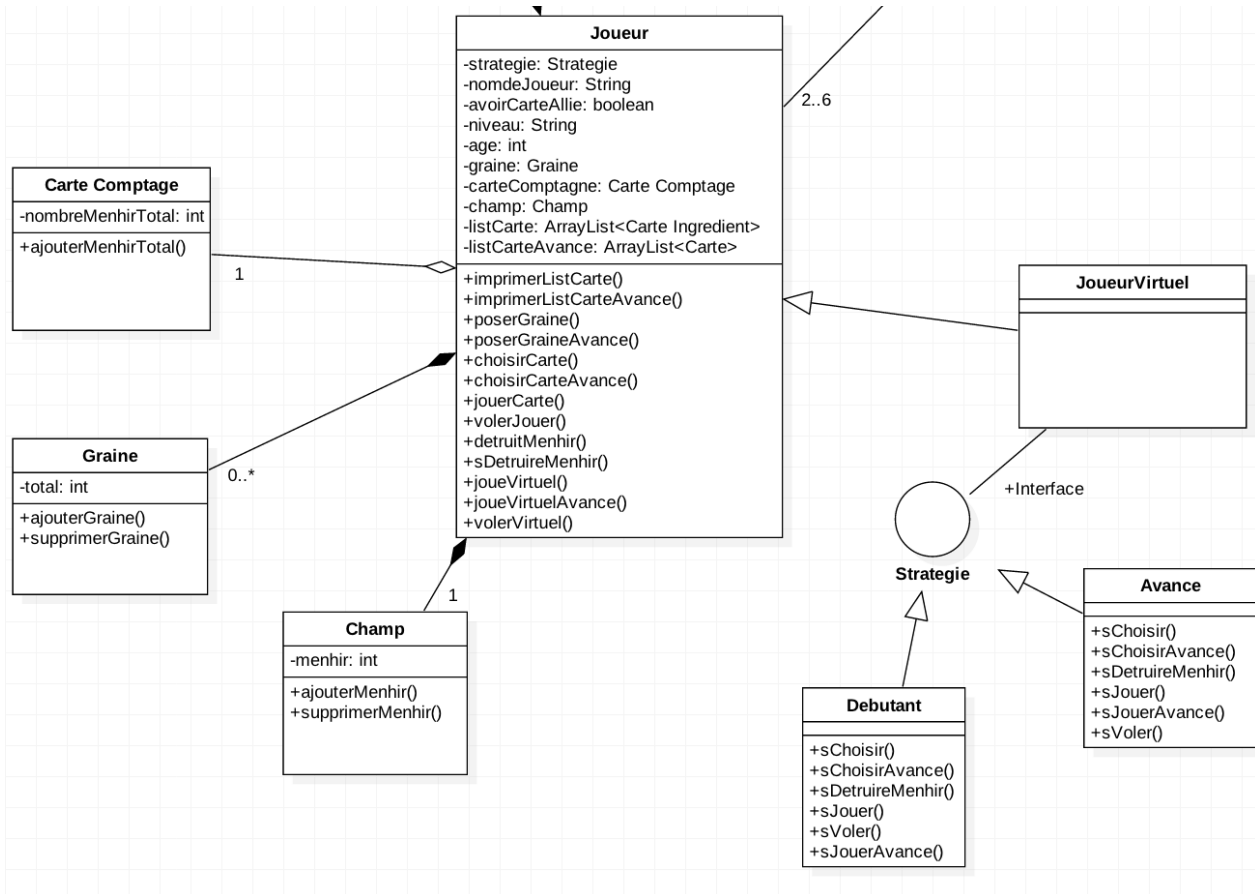
- Il y a deux types de Carte, donc, dans ce diagramme, il y a aussi deux classe qui représentent deux types. L'association entre la classe Carte et la classe Carte Ingrédient est la class Carte Alliés sont généralisations. Tout d'abord, nous allons présenter la classe Carte Ingrédient et la classe Carte Allié après.
- La classe Carte Ingrédient a tous les attributs et méthodes de class Carte comme l'attribut "nom", l'attribut "image", l'attribut "valeur", l'attribut "estJoué", l'attribut "estChoisir". En plus, il a aussi trois méthodes qui sont présenté les effets des trois "forces". Il y a trois sous-classes qui sont "Géant", "Engrais" et "Farfadet".
 - + effectGeant(): ajouter graines.
 - + effectEngrais(): ajouter menhirs et supprimer graines.
 - + effectFarfadet(): ajouter les graines chez joueur et supprimer chez adversaire.
- Comme la classe Carte Ingrédient, la class Carte Allié a aussi tous les attributs et méthodes de class Carte. En outre, elle a deux sous-classes : "Carte ChienDeGarde" et "Carte Taupes Géant"
La classe Carte ChienDeGarde a une méthode "chienDeGarde()" qui protège les graines contre le Farfadet.
La classe Carte Taupes Géant contient à méthode "taupeGeant()" qui détruit les Menhirs chez une adversaire.

Mais pourquoi ici il y a les différences entre la relation entre "CarteIngrédient" avec sa trois sous-classes et "CarteAillé" avec sa deux sous-classes? Nous avons fait selon les images de la carte que nous avons vu sur le site de ce jeu. Trois sous-classes sont composée directement dans une carte ingrédient. Contrairement, pour la carte aillé, on a deux cartes distincts pour "Taupés Géant" et "Chien De Garde" donc nous avons choisi les généralisations.

Rapport de L002 : Jeu de Menhirs

c) Bloc Joueur:

- La classe Carte Comptage a un seul attribut “nombreMenhirTotal” de type



entier. Elle a une méthode “ajouterMenhirTotal()” qui ajoute le nombre de menhirs sur la comptage de points à la fin de chaque Manche.

- La classe Graine contient un seul attribut qui correspond au nombre de graines d’un joueur. En outre, cette classe a deux méthodes pour ajouter les graines et supprimer les graines.
- La classe Champ a un attribut “menhir” qui concerne au nombre de menhirs sur le champ. Elle a aussi deux méthodes pour ajouter et supprimer les menhir.
- La classe Joueur contient dix attributs. Ce sont “nomDeJoueur”, “numeroDeJoueur”, “age”, listCarte, graine, champ, carteComptagne,... Cette classe a une méthode “choisirCarte()” qui permet au joueur de choisir la carte à

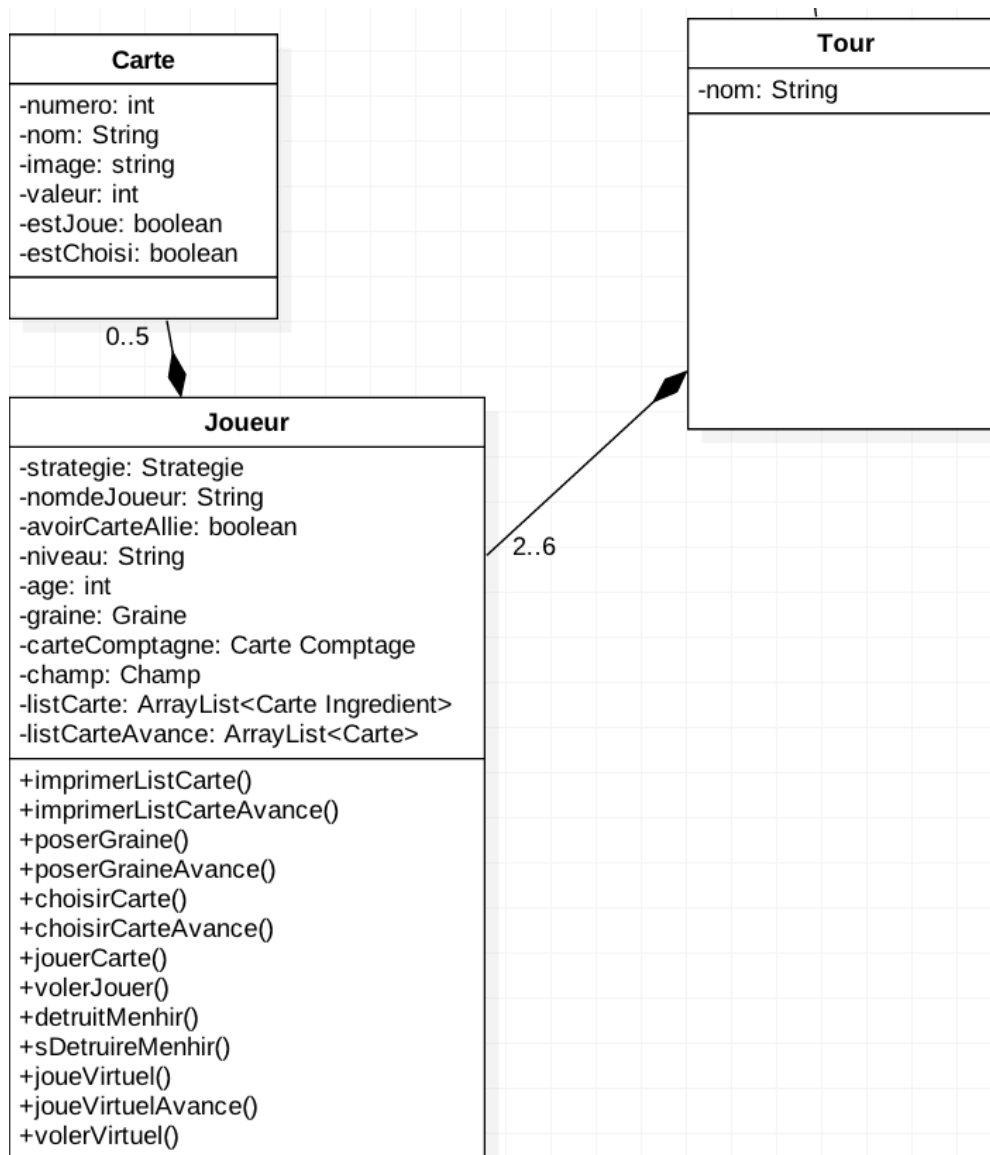
chaque tour (chaque saison). En outre, elle a méthode “poserGraine()”, “jouerCarte”, imprimerListcarte(),....

- La classe JoueurVirtuel est un cas spécifié de la classe Joueur. Mais nous avons remplacé les méthodes “choisirCarte()” par sChoisir(), “joueCarte()” par sJouer() et “volerJouer()” par sVoler() pour indiquer que ces trois méthodes vont être fait par l’interface du joueur virtuel. Nous allons détailler dans le patron de conception Stratégie après.

Pour les relations entre les classes, nous avons choisi les compositions pour le Champ et le Graine. C’est parce que ce sont deux parties indispensables d’un joueur pour tous les types de jouer ce jeu. À l’inverse, nous avons choisi l’agrégation pour la carte Comptagne à cause de son absence dans le type “Parties Rapides” de ce jeu.

Concernant à la multiplicité, c’est trop simple ici. Un joueur peut avoir zero ou plusieurs graines pendant un “game” mais il a une seule Champ et une seule carte Comptagne.

d) L'association entre 3 blocs:

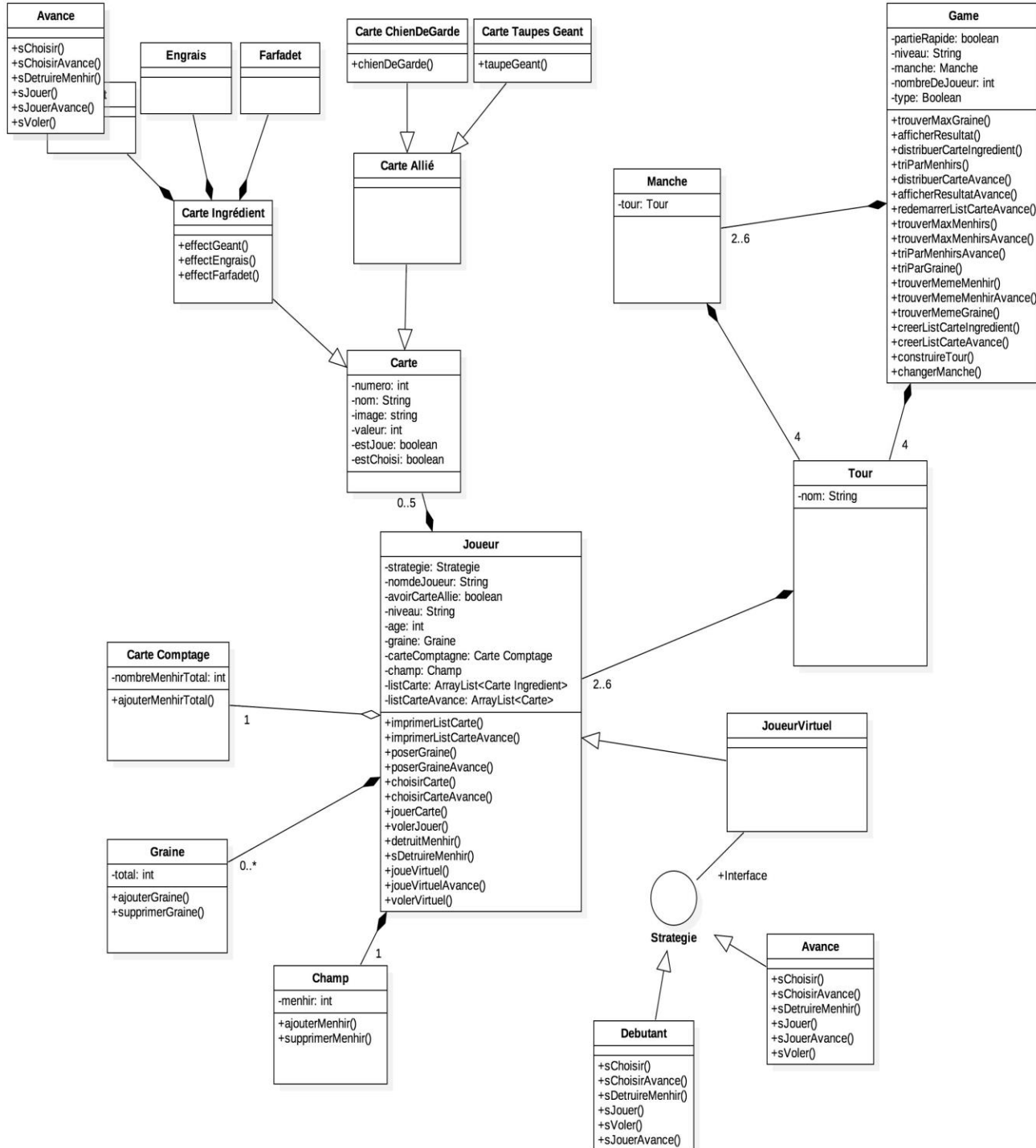


Chaque Joueur peut avoir de zero à cinq cartes parce que dans “parties rapides”, chaque joueur a maximum quatre cartes. Après chaque tour, si une carte est jouée, on la rend dans le pot commun et le numéro de cartes est diminué. Dans “règles avancées”, chaque joueur peut avoir plus une carte. C’est la carte Allié. Donc, chaque joueur a maximum cinq cartes dans “Règles Avancées”.

On peut trouver la relation de composition entre class Joueur et class Tour. D’après les règles de jeu Menhir, dans chaque “game”, il y a de deux à six joueurs qui le participent. À côté, chaque attribut dans chaque class qui a la visibilité “private” a aussi deux méthodes `get()` et `set()`.

Rapport de L002 : Jeu de Menhirs

e) Diagramme class complet final:



3) Patron de conception “Strategy”:

Pour les joueurs virtuels, ils ont besoin de suivre l'interface “Strategie” qu'on a créé les méthodes qui leur permettent de déterminer quelle “force” ils jouent sur la carte qu'ils ont choisi.

- Pour le niveau débutant, le joueur virtuel va choisir au hasard parmi trois “forces” Géant, Engrais ou Farfadets sur la carte Ingrédient.
- Pour le niveau avancé, le joueur virtuel va choisir la force selon ses conditions (le nombre de menhirs, le nombre de graines, les valeurs de chaque force “Géant”, “Engrais” ou “Farfadets” dans cette “saison”,....).

Grâce aux règles du jeu, son but est de faire pousser plus de menhirs que les adversaires. En cas d'égalité, le nombre de graines restantes sert à départager les joueurs. Donc selon nous, tout d'abord, si la valeur de “Engrais” est suffisante, on va choisir l'Engrais pour pousser les graines d'avoir plusieurs menhirs. Sinon, on va comparer deux valeurs de graines sur deux autres forces dans “saison” et puis choisir le plus grand pour obtenir plusieurs graines.

De plus, pour le type “Règles Avancées”, on a deux autres cartes “Taupés Géant” et “Chien De Garde” donc ensuite, nous expliquons quand on peut utiliser ces cartes. Pour la carte “Chien De Garde”, c'est plus simple, on va utiliser pour protéger les graines si les autres joueurs qui utilisent le “Farfadets” pour voler nos graines. À côté, pour la carte “Taupés Géant”, c'est plus compliqué. On va utiliser cette carte quand on veut détruire les menhirs chez les adversaires mais le problème est quand on va l'utiliser? Selon nous, on compare toujours les menhirs avec les autres joueurs. Si la différence est plus grand, on va utiliser cette carte sinon on va utiliser dans la dernière saison avec l'adversaire qui a plus menhirs dans le jeu.

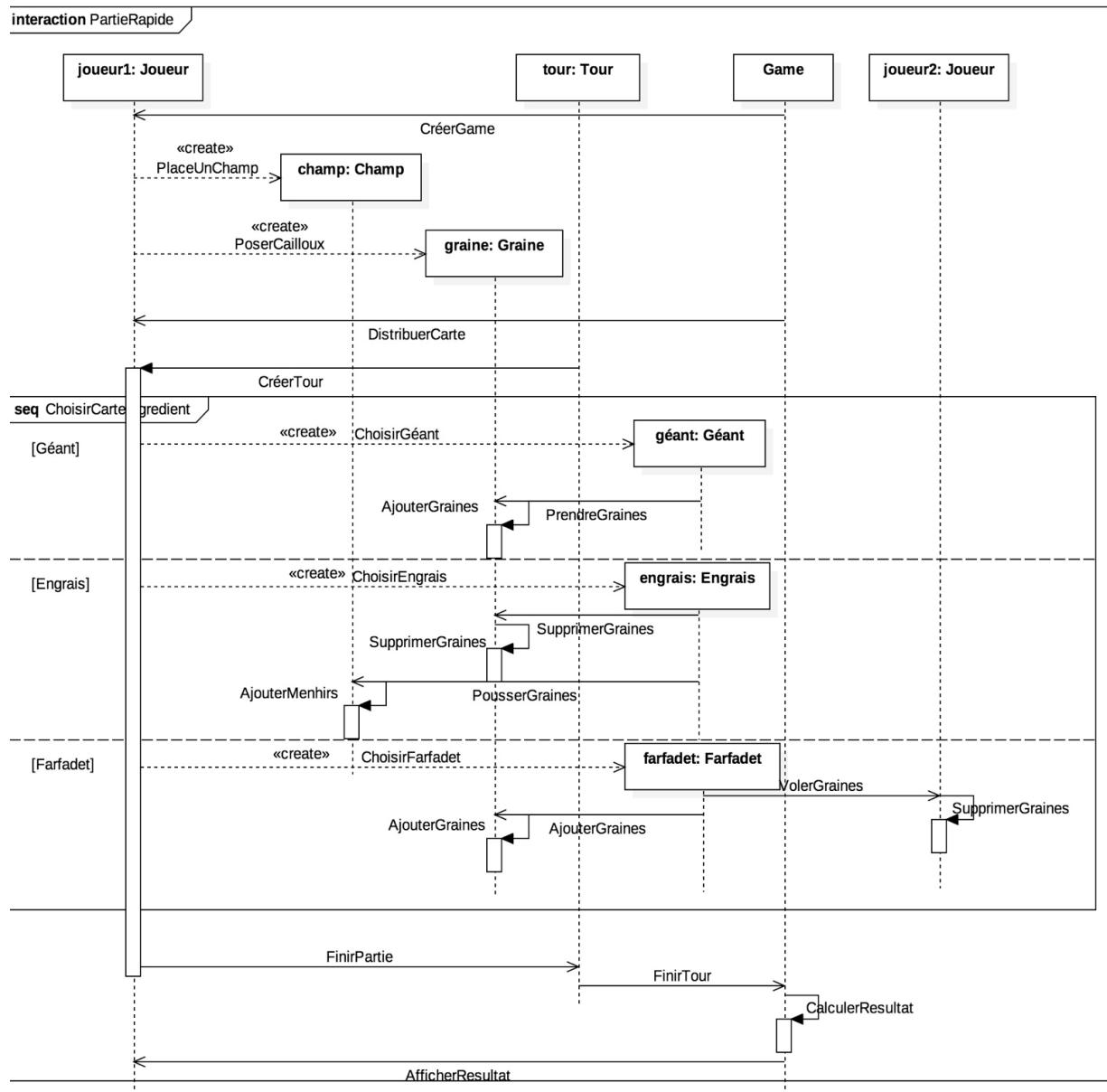
4) Diagramme de séquence :

Le diagramme de séquence permet de représenter les interactions entre les classes qu'on a définies dans le diagramme précédent de façon chronologique.

De plus, ce diagramme va nous donner une vue générale sur le déroulement d'un Game de ce jeu.

À cause de deux types de jouer ce jeu, on a créé deux diagrammes de séquences pour les deux types (Parties Rapides et Règles Avancées).

a) Parties Rapides :



En effet, après le Game est créer, le jeu va dérouler étapes par étapes.

- Le joueur place un champ qui est pour pousser et stocker les menhirs.
- Le joueur pose deux cailloux à côté le champ (qui sont appelés graines).
- Le Game va distribuer quatre cartes ingrédients pour chaque joueur.
- Le tour va indiquer quel joueur est dans la partie de jouer.

Ensuite, on a créé un autre sous-diagramme de séquence appelé “ChoisirCarteIngrédient” qui est composé dans ce diagramme cas le joueur peut choisir soit le Géant soit le Engrais soit le Farfadets pour jouer.

+ “Géant”: la carte va aider de prendre les graines supplémentaires selon le valeur sur la carte et va ajouter sur les graines de ce joueur.

+ “Engrais”: D’abord selon le valeur sur la carte, il va supprimer les graines et après il va pousser les graines sur le champ pour transformer aux menhirs.

+ “Farfadets”: Selon la valeur sur la carte, on va voler les graines de une autre joueur2 (joueur2 va supprimer ses graines) et après on va ajouter les graines dans les graines de joueur1.

- Le joueur va déclarer de finir sa partie.

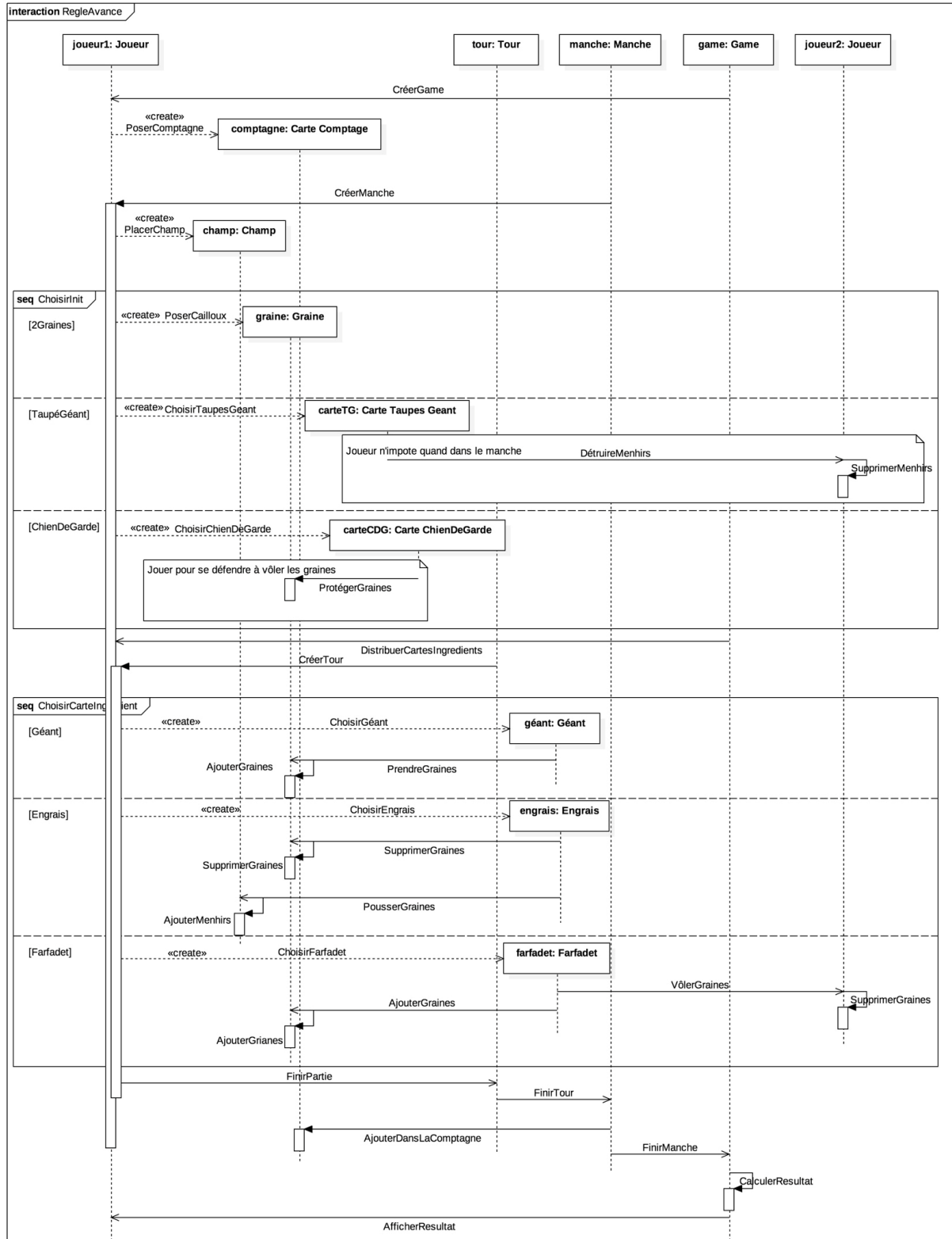
Ce diagramme va répéter 4 fois (selon 4 saisons que on va jouer) et après on va utiliser le message “FinirTour” pour indiquer que le Game est fini.

- Le Game (le système) va calculer le résultat.
- Finalement, il va afficher le résultat pour tous les joueurs.

Et puis, on va aller sur le diagramme de séquence de “Règles Avancées” qui est plus compliqué que de “Parties Rapides”.

Rapport de L002 : Jeu de Menhirs

b) Règles Avancées :



Le principe de ce diagramme est le même que le diagramme de séquence de “Parties Rapides” mais on a ajouté quelques choses selon les différences entre les règles du jeu de deux type de jouer.

D’abord, on va introduire la classe “Manche” qui est seulement dans le “Règles Avancées”. Le Game va dérouler dans plusieurs manches (les manches est égal aux les joueurs) et dans chaque manche on va dérouler comme une jeu sous le type “Parties Rapides”.

Ensuite, après le Game est créé, le joueur va poser une carte de comptage qui est utilisé pour stocker le nombre de menhirs après chaque manche de jouer et après tous les manches, il va utiliser pour le résultat final.

De plus, on a créé un autre sous-diagramme de séquence appelé “ChoisirInit” qui est composé dans ce diagramme de séquence car tout d’abord, après placer un champ, le joueur peut choisir soit poser deux cailloux comme la “Parties Rapides” soit la carte Alliés “Taupés Géant” soit la carte Alliés “Chien De Garde pour jouer. Nous allons détailler le but et le déroulement de deux type de carte Alliés.

- + “Taupés Géant” pourra jouer n’importe quand dans le manche si on veut détruire les menhirs chez l’adversaire. Selon la valeur de la carte, on va supprimer les menhirs chez l’adversaire que on veut détruire.

- + “Chien De Garde” peut jouer pour se défendre à éviter de voler les graines. Selon le valeur de la carte, on peut protéger les mêmes graines.

Comme le diagramme de séquence de “Parties Rapides” mais après chaque manche est fini, on va ajouter les menhirs sur la carte de comptage. De plus, le manche va répéter selon le nombre de joueur et si tous les manches sont finis, on va calculer le résultat fini et afficher pour tous les joueurs.

III/ ETAT ACTUEL DE L'APPLICATION :

1) Interface de la console

Dans le livrable 2, on a réussi de faire le moteur du jeu fonctionné. En exécutant l'application, on va démarrer l'interface en console du jeu.

L'écran de console va vous demander de saisir votre nom et après votre âge parce que selon l'énoncé de Menhir, on va organiser l'ordre de joueurs selon les âges des joueurs (le joueur le plus jeune va commencer le jeu). Et puis, le système va vous demander le nombre de joueurs (de 2 à 6 joueurs). Ensuite, on a besoin de taper quel mode de jeu on va jouer (Parties rapides ou Règles avancées) et quel niveau du jeu (Débutant ou Avancé).

```
-----Menhir-----
```

```
Saisir votre nom
```

```
Hoang
```

```
Saisir votre age
```

```
18
```

```
Saisir le nombre de joueurs =
```

```
3
```

```
Choisir le niveau de joueur Virtuel : Debutant ou Avance
```

```
Debutant
```

```
Choisir la mode de jouer Rapide ou Avance:
```

```
Rapide
```

Rapport de L002 : Jeu de Menhirs

Après, le système va construire l'ordre de la liste de joueurs et distribuer la liste carte pour chaque joueur puis va afficher la liste de joueurs après organiser et les informations pour chaque joueur (le nombre de graines initiales ainsi que la liste carte qui est composé les cartes Ingrédients et les cartes Alliés).

```
Voici la liste de joueur
JoueurVirtuel2 18 Debutant
Hoang 23 Physique
JoueurVirtuel1 21 Debutant
Donc JoueurVirtuel2 va commencer d'abord
Voici les cartes pour chaque joueur
List carte pour JoueurVirtuel2
4 CHANT DE SIRENE
18 ESPRIT DE DOLMEN
5 CHANT DE SIRENE
19 ESPRIT DE DOLMEN

JoueurVirtuel2 a 2 graines
List carte pour Hoang
9 FONTAINE D'EAU PURE
11 FONTAINE D'EAU PURE
22 RIRES DE FEES
10 FONTAINE D'EAU PURE

Hoang a 2 graines
List carte pour JoueurVirtuel1
23 RIRES DE FEES
21 RIRES DE FEES
8 LARMES DE DRYADE
3 CHANT DE SIRENE

JoueurVirtuel1 a 2 graines
```

a) Parties Rapides:

Dans ce mode de jeu, on va commencer tour par tour (printemps, été, automne et hiver). Dans chaque tour, chaque joueur dans la liste que on a déjà organiser va jouer (choisir la carte dans la liste carte et choisir la force pour effectuer pour le joueur "physique" et pour les joueurs virtuels le système va jouer selon la stratégie qu'on a défini et après afficher quelle carte et quel force ils ont choisi).

Rapport de L002 : Jeu de Menhirs

Ici la saison Printemps
JoueurVirtuel2 a choisi carte numero 5
JoueurVirtuel2 a choisi Engrais
Choisir une carte que vous n'avez pas encore joué entre:
Carte 9 avec le numero 0
Carte 11 avec le numero 1
Carte 22 avec le numero 2
Carte 10 avec le numero 3
Quelle carte vous voulez jouer? Entrer le numero de carte:
1
Ceci les valeurs de chaque force
Geant: 2
Engrais: 2
Farfadet: 1
Vous choisissez quel force?
geant
Donc vous avez 4 graines
JoueurVirtuel1 a choisi carte numero 23
JoueurVirtuel1 a choisi Geant

Après 4 tours de jeu, le système va afficher les menhirs et les graines de chaque joueur et puis déterminer qui gagne du jeu en comparant les menhirs d'abord et après les graines (si les menhirs sont le meme).

Hoang a 4 menhirs et 3 graines.
JoueurVirtuel2 a 2 menhirs et 0 graines.
JoueurVirtuel1 a 0 menhirs et 5 graines.
Le gagnant est
Hoang a 4 menhirs et 3 graines.

Le jeux est fini!!!

b) Règles Avancées:

Pour ce mode de jeu, on va jouer le nombre de manches selon le nombre de joueurs et chaque manche va avoir lieu comme une partie rapide mais au début de manche, on a besoin de choisir 2 graines ou une carte allié et le système va réorganiser l'ordre de joueurs en changeant le premier jouer et après construire la liste. (construireTour(joueur) pour le premier manche et changerManche(joueur) pour les autres manches).

```
Vous voulez prendre une carte Allie ou prendre 2 graines?  
Saisir "true" si vous voulez prendre une carte Allie  
Saisir "false" si vous voulez prendre 2 graines
```

Ensuite on va jouer comme la partie rapide mais après chaque tour de manche, le système va vous demander d'utiliser la carte allié (si vous avez une).

```
Vous avez une carte Allie CHIEN DE GARDE  
Vous voulez la jouer? oui ou non ?
```

Et si quelqu'un veut voler vos graines, si vous avez une carte chien de garde, le système affichera que vous avez une carte allié et il vous demande d'utiliser ou pas.

```
JoueurVirtuel 1 veut prendre 1 graines de Hoang  
Vous avez une carte Allie CHIEN DE GARDE avec la valeur = 2  
Vous voulez la jouer ? oui ou non?  
oui
```

Rapport de L002 : Jeu de Menhirs

À côté de la, si vous restez la carte Taupé Géant dans deux dernières tours, le système va remarquer et afficher que vous voulez utiliser ou pas. Si vous jouer cette carte, le système va afficher les informations.

JoueurVirtuel 2 utilise carte TAUPE GEANT

Voici la liste d'autre joueurs

Entrer numero 1 pour: Hoang avec 2 Menhirs Aldults

Entrer numero 2 pour: JoueurVirtuel 1 avec 0 Menhirs Aldults

JoueurVirtuel 2 veut detruire 1 menhirs de JoueurVirtuel 1

Et finalement, après tous les manches, le système va afficher les informations de menhirs des joueurs et déterminer qui gagne cette partie.

La partie est teminee

afficher les resultats

Hoang a 7 menhirs et 0 graines (les graines dans la derniere manche).

JoueurVirtuel 1 a 3 menhirs et 0 graines (les graines dans la derniere manche).

JoueurVirtuel 2 a 2 menhirs et 0 graines (les graines dans la derniere manche).

Hoang a gagne

Le jeux est fini!!!

2) Interface graphique :

a) Le pattern MVC

Le pattern MVC signifie Modèle – Vue – Contrôleur, et ce pattern est une architecture qui sépare les données et la logique de l'application (le modèle), l'interface homme – machine (la vue), et la logique de traitement (contrôleur).

Le modèle

C'est la classe **Partie**, le cœur du programme, c'est lui qui s'occupe du traitement des actions demandées par l'utilisateur. La classe **Partie** est héritée la classe **Observable**

La vue

Ce qu'on nomme la vue est en fait une IHM. Celle-ci est en fait ce qu'a l'utilisateur sous les yeux. Nous avons créé la classe **MainFenetre** qui est implémenté l'interface **Observer**.

À côté, on a une autre classe qui a nommé **Main_Menu** pour demander l'utilisateur de configurer les informations pour le jeu (le nom, l'âge, le mode ainsi que le niveau du jeu).

Le contrôleur

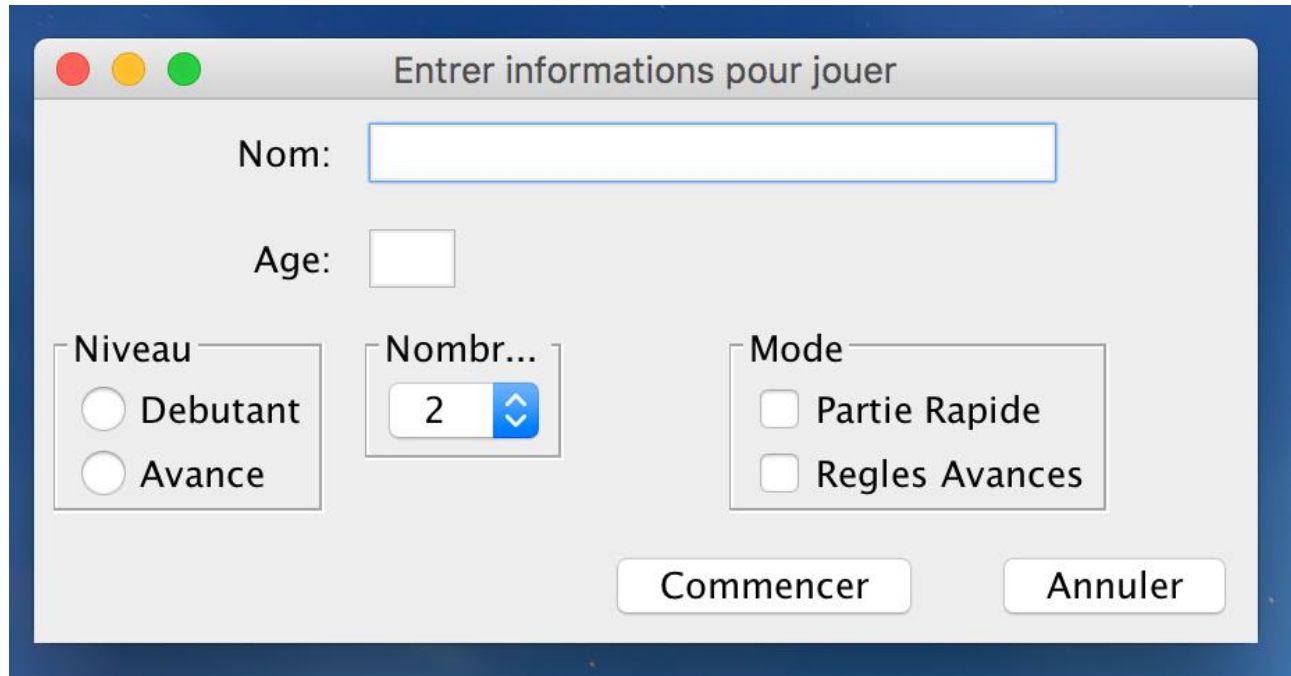
Il permet de faire le lien entre la vue et le modèle lorsqu'une interaction utilisateur est survenue sur la vue ! Nous avons donc créé la classe **Contrôleur** qui aura pour rôle de contrôler les données et, le cas échéant, ira dire au modèle qu'il peut utiliser les données envoyées par l'utilisateur en toute tranquillité !

En appliquant le principe du pattern MVC sur le projet, nous avons créé un modèle de base pour les deux interfaces graphique et ligne de commande. En particulier, dans l'interface graphique, nous avons utilisé le MVC dans Swing : **JPanel**, **JTable**, **JOptionPane**, **JScrollPane**, **JComboBox**, **JCheckBox**, **JRadioButton**,... pour pouvoir créer des vues pour la configuration des informations de jeu et pour l'écran principal de jouer ainsi que leurs modèles dynamiques et leurs contrôleurs.

b) Interface graphique

Rapport de L002 : Jeu de Menhirs

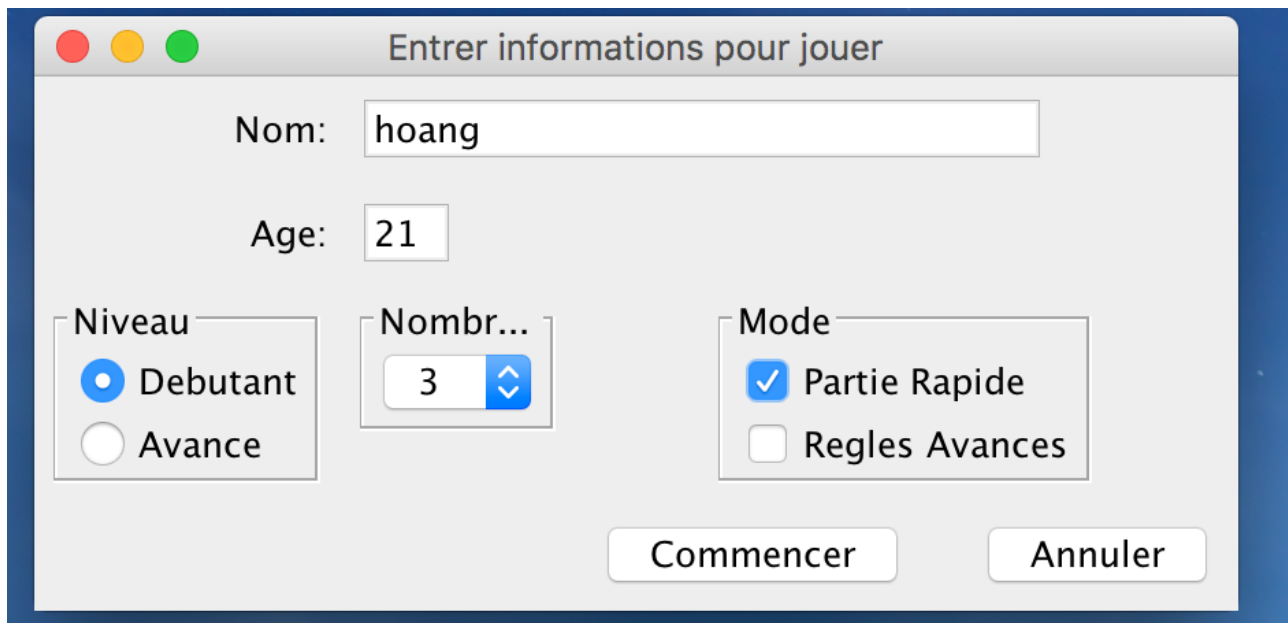
Lors d'un lancement du programme, une fenêtre va afficher et vous demander de saisir votre nom, age et ainsi de configurer la partie (nombre de joueurs totaux, niveau de difficulté, mode de jouer)



The screenshot shows a window titled "Entrer informations pour jouer". It contains the following elements:

- Nom:** A text input field.
- Age:** A text input field.
- Niveau:** Two radio buttons labeled "Debutant" and "Avance".
- Nombr...:** A spinner box showing the value "2".
- Mode:** Two checkboxes labeled "Partie Rapide" and "Regles Avances".
- Buttons:** "Commencer" and "Annuler" at the bottom right.

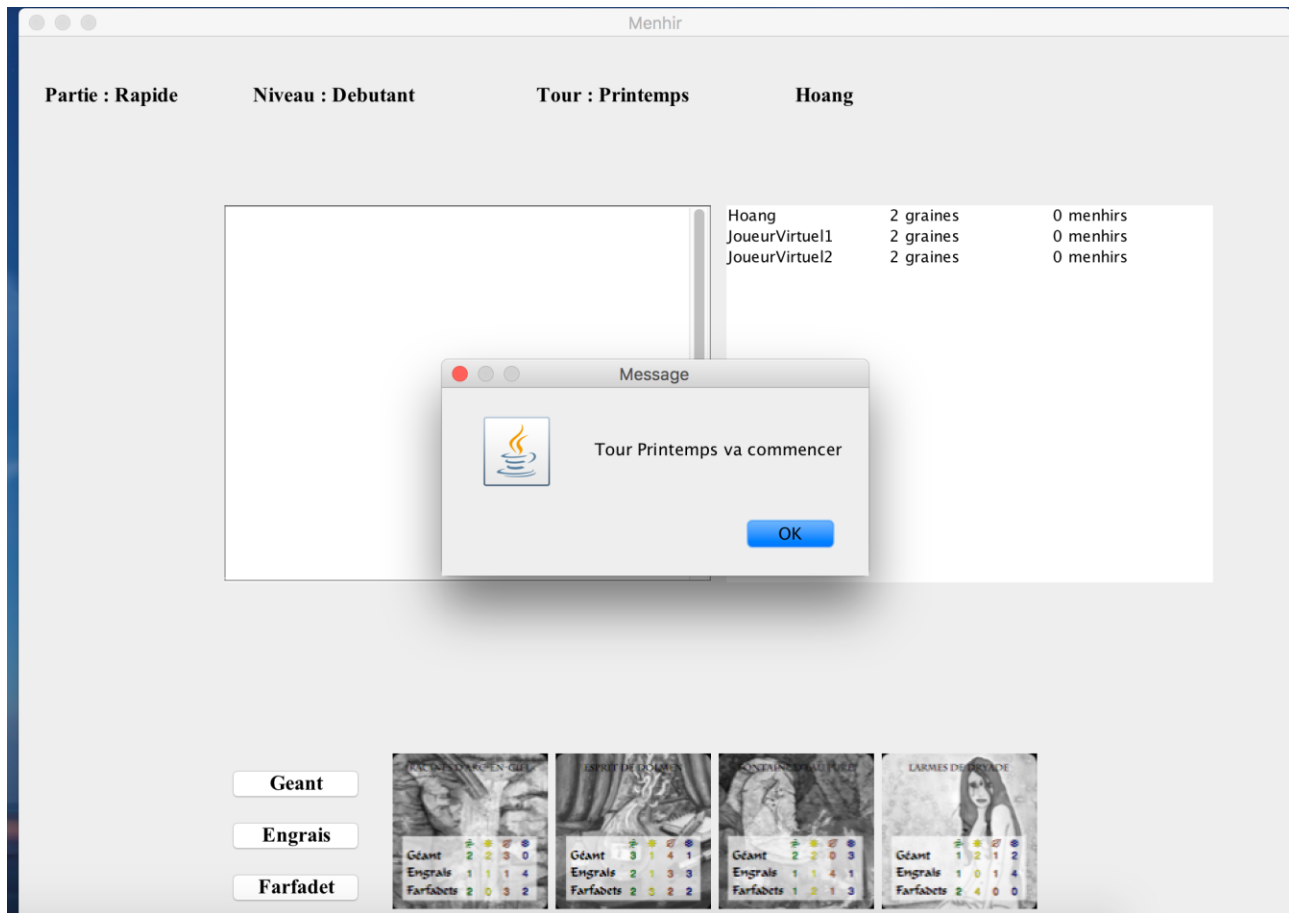
Lorsque l'on finit de saisir et cliquer sur le bouton **Commencer**.



The screenshot shows the same window with the following data entered:

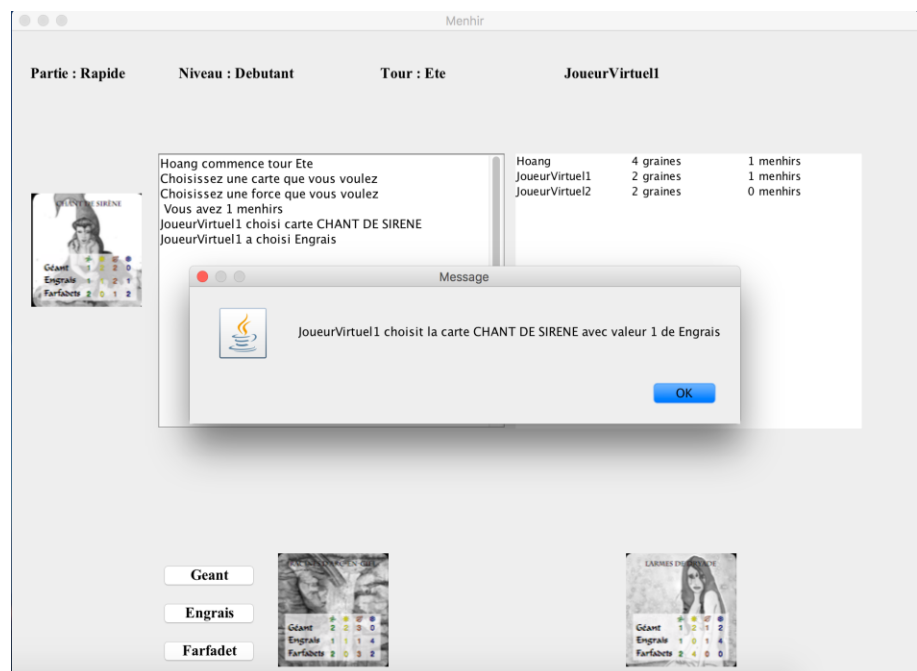
- Nom:** "hoang"
- Age:** "21"
- Niveau:** "Debutant" is selected.
- Nombr...:** "3"
- Mode:** "Partie Rapide" is checked.
- Buttons:** "Commencer" and "Annuler" are still present.

Rapport de L002 : Jeu de Menhirs



Le contenu de la fenêtre actuel va être modifié, la partie du jeu va afficher.

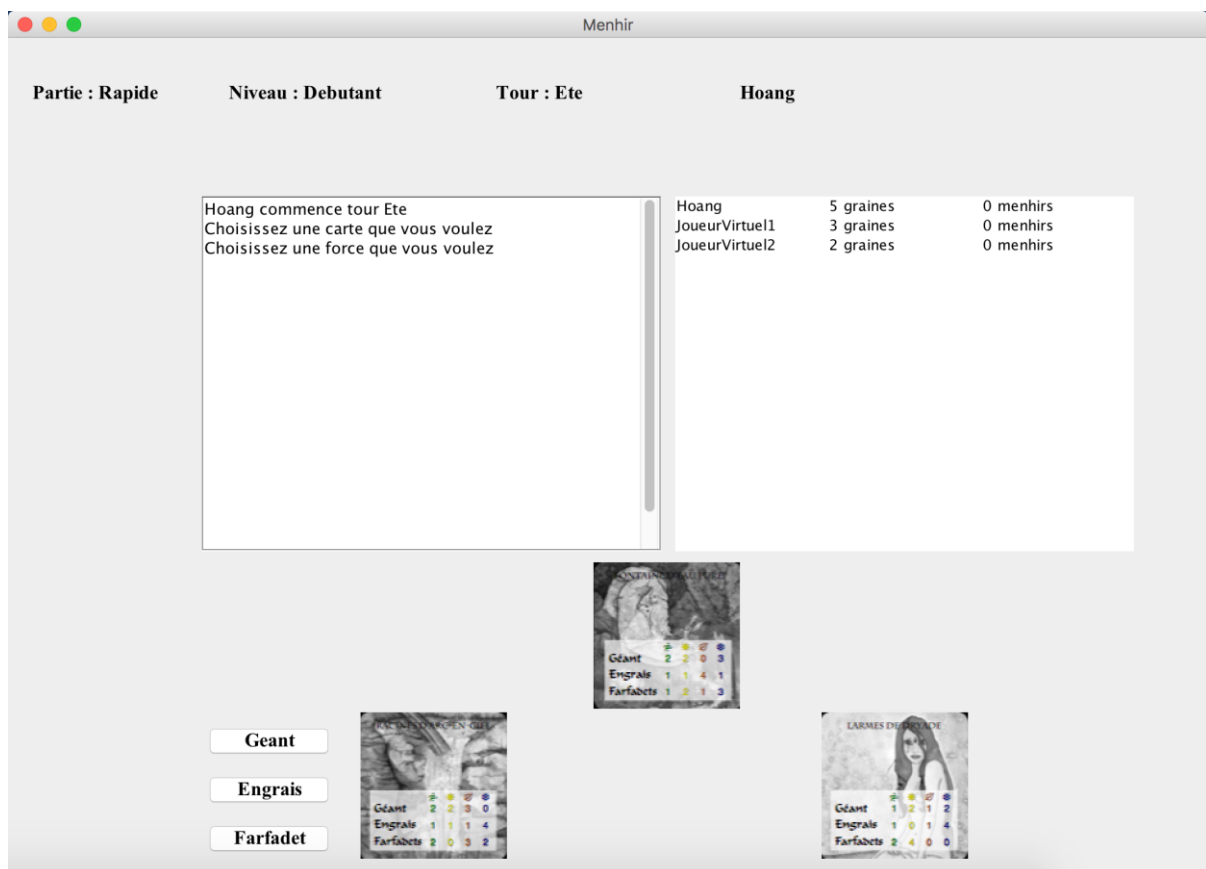
Dans la fenêtre de jouer, on a déjà désigné un “mainframe” avec tous les composants de jeu dans les images à côté.



Rapport de L002 : Jeu de Menhirs

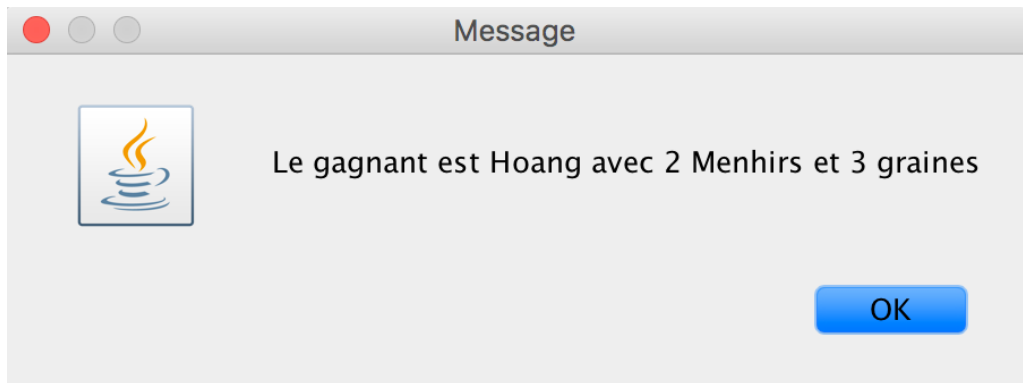
Dans l'image, on peut voir :

- Il y a 4 Labels pour indiquer les informations de cette partie (mode jouer, niveau, tour actuel, joueur en train de jouer).
- Au centre de la frame, il y a une JTextArea pour afficher les informations de cette partie (les joueurs virtuels ont choisi quelle carte, quel force,...) et une JTable pour afficher les graines et les menhirs de la liste de joueurs.
- En plus, à droite de frame, il y a un label qui est représenté la carte que le joueur virtuel a choisie. Et ce label est visible seulement si le joueur virtuel a choisi une carte.
- Dans le panel de joueur physique, il y a trois buttons pour trois force d'une carte (géant, engrais et farfadet) et 4 buttons représentant 4 cartes dans la liste carte de joueur "physique". Quand on a choisi une carte, sa position dans la liste va être invisible et cette carte va afficher dans un espace entre JTextArea et panel de joueur physique.



Rapport de L002 : Jeu de Menhirs

- Et finalement, quand la partie a fini, le système va afficher un JOptionPane pour présenter le résultat de cette partie.



- Pour les règles avancés, c'est quasiment le même interface que les Parties Rapides donc on peut voir directement dans le démo du projet.

IV/ DIFFICULTES RENCONTRES :

Comme les débutants en Java surtout le Swing, nous avons eu des difficultés à organiser le code de mon application et à nous familiariser avec des différents langages à la fois.

Les principaux problèmes rencontrés étaient principalement dus à des erreurs d'inattention. À cause des erreurs de syntaxe, nous avons perdu beaucoup de temps à les corriger et vérifier si tout fonctionne bien.

Concernant le design, nous nous sommes fallu parcourir plusieurs de tutoriels sur le Swing pour arriver aux objectifs attendus. Un problème qui méritait d'être cité concernait l'adaptation des composants sur le mainFrame (JTable, JLabel, ImageIcon,...).

.De plus, nous avons plusieurs projets dans les autres UVs donc nous n'avons pas plusieurs temps pour concentrer sur un projet.

Finalement, nous sommes étudiants étrangères donc ce sont aussi les difficultés de la communication ainsi que les fautes d'orthographe dans ce rapport-là.

V / CONCLUSION :

L'objectif de ce projet était de concevoir une version électronique du jeu carte "Menhirs". Pour ce faire, nous avons au préalable construit un modèle qui contient les diagrammes cas d'utilisation, le diagramme classe et les diagrammes séquences. De plus, ce modèle est capable de répondre aux règles du jeu "Menhirs" par exemple le cas d'utilisation pour identifier les ensembles des activités de joueurs et système, le diagramme classe pour indiquer quelles classes on va utiliser et le lien entre les classes, les diagrammes séquences pour illustrer un tour de jeu.

Nous savons que tout ce que nous avons fait est seulement la première conception donc le développement peut induire plusieurs modifications sur la conception surtout à nos avis, c'est sur le diagramme de classe (le lien, l'identification,..) ainsi que les attributs et les méthodes des classes.

Bien qu'il y a beaucoup de changements pendant la réalisation du code, la partie UML nous a donné une vue générale sur notre future application. Malgré la difficulté de Français, nous avons fait plusieurs efforts pour comprendre l'énoncé et écrire ce rapport.

En effet aucun des membres du groupe n'avez utilisé Java auparavant et donc ce projet a été très enrichissant. Grâce à ce projet, nous avons pu mesurer les possibilités offertes par UML et Java. Il aurait été intéressant de développer l'application à partir de nos spécifications afin de mieux mesurer l'efficacité du langage de programmation orientée objet.