

|  |
| --- |
| **SPEAPING**  APICHESS |

|  |
| --- |
| **RAPPORT DE PROJET** |

|  |  |
| --- | --- |
| **N° de version** : 1.1 | **Auteur** : Meliani Keyvan |
| **Date de diffusion** : 09/01/2016 |  |

**SOMMAIRE**

Sommaire

[1. Cahier des charges 4](#_Toc483905842)

[1.1 PRESENTATION GENERALE 4](#_Toc483905843)

[1.1.1 Introduction 4](#_Toc483905844)

[1.1.2 Descriptif de l’opération 4](#_Toc483905845)

[1.2 Restrictions 5](#_Toc483905846)

[1.2.1 Site Web 5](#_Toc483905847)

[1.2.2 Projet 5](#_Toc483905848)

[1.2.3 Code 5](#_Toc483905849)

[1.3 Technologies 6](#_Toc483905850)

[1.3.1 Site Web 6](#_Toc483905851)

[1.3.2 Langage de développement 6](#_Toc483905852)

[1.4 Charte graphique 7](#_Toc483905853)

[1.4.1 Charte graphique 7](#_Toc483905854)

[1.5 Repartition des charges 8](#_Toc483905855)

[1.5.1 Membres 8](#_Toc483905856)

[1.5.2 Répartition 8](#_Toc483905857)

[1.5.3 Diagramme de PERT 9](#_Toc483905858)

[1.5.4 Diagramme de GANTT 10](#_Toc483905859)

[2. Repartition des tâches 13](#_Toc483905860)

[3. Récit de la réalisation 15](#_Toc483905861)

[3.1 Joies 15](#_Toc483905862)

[3.2 Peines 15](#_Toc483905863)

[3.3 Problèmes rencontrés 15](#_Toc483905864)

[4. Annexes 17](#_Toc483905865)

[4.1 Impression d’écran 17](#_Toc483905866)

[4.2 Dessin d’origine 19](#_Toc483905867)

[4.2.1 Cases de l’échiquier 19](#_Toc483905868)

[4.3 Structures 20](#_Toc483905869)

[4.4 Algorithmes 21](#_Toc483905870)

[4.4.1 Intelligence artificielle 21](#_Toc483905871)

# Cahier des charges

## PRESENTATION GENERALE

### Introduction

Souvent nommé comme le plus noble des jeux, le roi des jeux, depuis le 6eme siècle il est considéré comme le défi intellectuel le plus abouti. Menant les meilleurs joueurs à avoir des capacités intellectuelles et plus précisément de mémorisation bien au-dessus de la moyenne mondiale.

Et menant même certains grands joueurs ayant consacré leur vie aux échecs dans la folie, dépression ou paranoïa tel que le grand joueur polonais Akiba Rubinstein 3 fois champion de Russie, Paul Morphy grand joueur américain spécialiste des parties à l’aveugle et pourtant affirmant qu’il n’était pas un joueur d’échec, l’autrichien Wilhelm Steinitz et même Robert James Fischer.

Pour le premier coup chaque joueur a le choix entre 20 coups possibles. Ce qui veut dire qu’il y a déjà 400 possibilité après le premier coup de chaque joueur. Après 4 coup il y a déjà plus de 300 milliards de possibilités. Le nombre de positions théoriquement possible dans une partie s’élève à plus de 10 puissance 45, le nombre d’atome dans le système solaire.

Ce qui différencie un grand joueur se trouve surtout dans sa capacité à anticiper. Les meilleurs anticipent presque tous les déplacements et cela demande une concentration extrême.

Ce ne sont tout de même que des exceptions et jouer aux échecs apporte de nombreux bienfaits. Amélioration de l’attention, élaboration de stratégies avec un objectif à atteindre et résolution de problèmes complexes.

### Descriptif de l’opération

Ce projet a pour objet de réaliser un jeu d’échec possédant une intelligence artificielle permettant de jouer à un bon niveau. A l’instar des champions de jeux d’échecs et en essayant donc d’imiter les capacités cognitives de l’être humain, il faudra que l’intelligence artificielle puisse analyser et évaluer l’état du terrain, de rechercher le meilleur coup par rapport à une succession d’évaluations des différentes possibilités que la partie peut engendrer. Le coup le plus profitable sera ainsi choisi permettant une probabilité maximale de victoire.

La durée de ce projet est de 4 mois. Et la date de fin, étant incompressible, devra être respectée. Il sera quand même laissé libre aux différents développeurs du projet de le poursuivre, de l’améliorer et de l’utiliser à toutes fin considérée comme utile.

Ce projet comprendra 3 phases scindée par 3 soutenances (dates ci-dessous).

Pour chaque soutenance :

* Un site web devra être utilisé
* Un rapport devra être rendu

## Restrictions

### Site Web

Le site web devra posséder un accueil permettant d’accéder aux éléments suivants :

Présentation du projet (historique, membres, chronologie de réalisation, problèmes rencontrés, solutions envisagées)

Les liens sur les sites (des membres, des logiciels, des images, sons, librairies, …)

Des liens de téléchargements (rapport, projet version complète et lite)

### Projet

Le projet devra être continuellement mis à jour à l’aide d’un logiciel de gestion de version. Ceci permettant entre autres la conservation d’un historique de chaque modification des fichiers.

Le gestionnaire de versions git devra être utilisé.

### Code

Le développement de l’application devra respecter les règles suivantes :

* Le langage C sous sa version standard C99
* Etre correctement indenté
* Les noms des identifiants, variables, fonctions … seront en anglais
* Le code ne devra pas dépasser 80 caractères par lignes et ne pas avoir d’espaces en fin de ligne
* Le code devra ne pas disposer de warning à la compilation et devra être compilé avec les options :
  + -Wall
  + -Wextra
  + -std=c99

## Technologies

### Site Web

Les technologies pour développer le site web sont laissées libre.

L’équipe a donc envisagé plusieurs solutions :

Utilisation d’un Content Management System

* WordPress
* Drupal

L’utilisation d’un CMS permet de gérer facilement, rapidement, l’aspect esthétique et le contenu d’un site web.

Utilisation de langages web

* HTML et CSS afin d’en développer la maquette
* PHP/JS afin de prodiguer de l’interactivité

Afin de pouvoir faciliter et développer celui-ci rapidement l’utilisation et la modification de Template Open Source a été envisagé.

Pour pouvoir fournir des liens permettant le téléchargement l’utilisation d’une base de données l’utilisation de MySQL et Oracle a été envisagé.

* Oracle permet de générer de manière performante un très grand lot de données. Les licences sont payantes selon les type de serveurs.
* MySQL permet de gérer facilement et de manière performante une base de données de manière souple, légère et gratuitement.

Il a donc été souhaité d’utiliser les technologies web HTML, CSS, PHP et JavaScript dans un intérêt pédagogique et d’utiliser une base de données MySQL afin de faciliter le développement de celle-ci.

### Langage de développement

Une première analyse rapide montre plusieurs répétitions dans le principe logique d’un jeu d’échec. L’ensemble des pions, l’ensemble des cases du terrain et la partie peuvent être structurés et modélisé par un langage Objet.

Les langages de programmation C++, C#, Java, JavaScript ou même PHP étant des langages partiellement vu ou connu de l’équipe de développement en charge du projet peuvent permettre de résoudre le problème.

En effet pouvoir modéliser une grande partie du projet par des Classes régies par de l’héritage où chacune possèderait ses attributs et méthodes qui lui sont propres simplifierais énormément le développement.

Mais l’utilisation d’un langage orienté Objet présente plusieurs limites dont en premier lieu l’aspect pédagogique.

Le langage de développement choisi imposé par la restriction sera donc le C standard 99.

Afin de pouvoir faciliter le développement de l’interface graphique et de la représentation de la partie, les librairies suivantes seront utilisées :

* GTK
* SDL et peut être certains de ses dérivées

Il est aussi envisagé d’utiliser la librairie :

* Math

## Charte graphique

### Charte graphique

Il est laissé libre de choisir l’esthétisme et l’interface de son choix du moment qu’elle soit claire et explicite. Celle-ci devra être une représentation parfaite de la partie en cours et d’en faciliter l’ergonomie du jeu.

## Repartition des charges

### Membres

**Meliani Keyvan**

Chef de projet. Etudiant en API.

**Hautier Julien**

Membre du projet. Etudiant en API.

**Gravaillac Jean-Charles**

Membre du projet. Etudiant en API.

### Répartition

Le travail sera réparti en 2 partie.

Une première partie, algorithmique, de modélisation et d’anticipation du projet.

Une deuxième parie, de développement et d’avancement sur le projet.

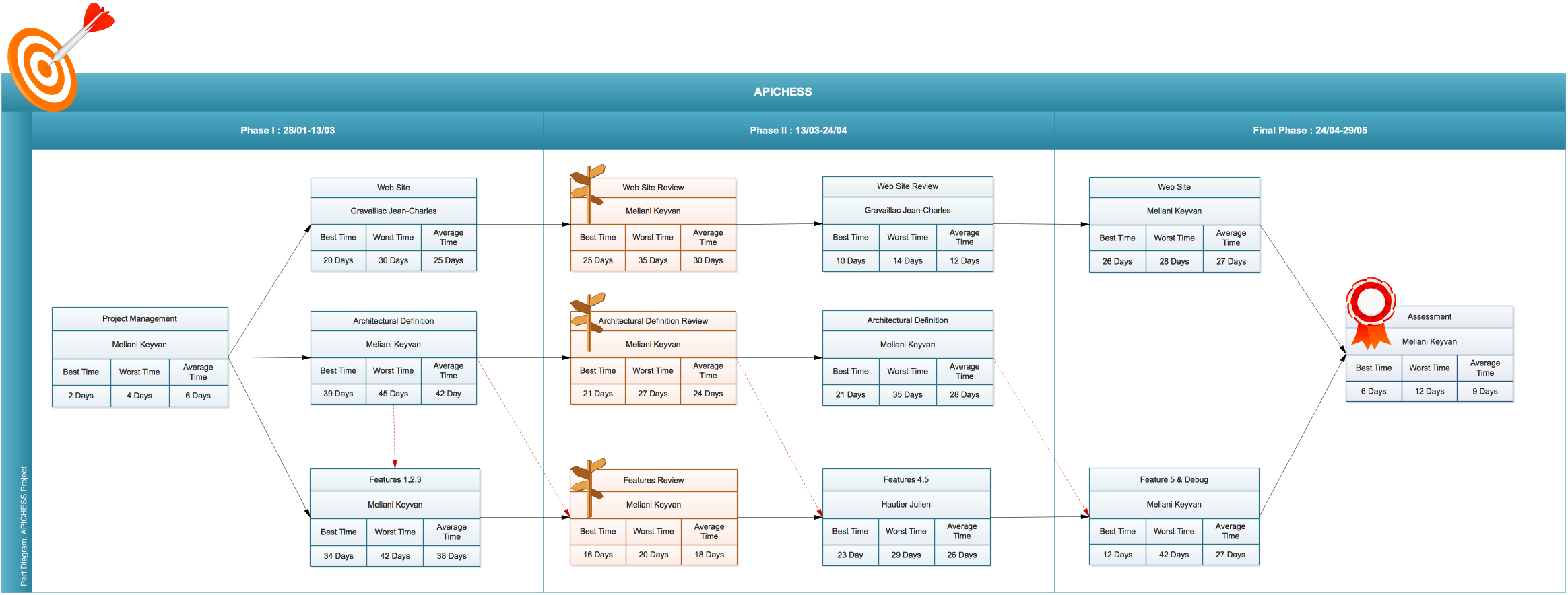
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Meliani Keyvan | | | | | |
| Soutenance I | | Soutenance II | | Soutenance III | |
| Modélisation | Développement | Modélisation | Développement | Modélisation | Développement |
| Interface Console | Mode 1 Joueur | Mode 2 Joueur | Site Web | IA - Analyse | Review |
| Moteur du Jeu | Moteur Console | IA - Analyse | Feature Review |  |  |
| Mode 1 Joueur |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Hautier Julien | | | | | |
| Soutenance I | | Soutenance II | | Soutenance III | |
| Modélisation | Développement | Modélisation | Développement | Modélisation | Développement |
| Echiquier | Echiquier | IA - Recherche | Inf° Console | IA - Recherche | Site Web |
| Pièces | Pièces |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Gravaillac Jean-Charles | | | | | |
| Soutenance I | | Soutenance II | | Soutenance III | |
| Modélisation | Développement | Modélisation | Développement | Modélisation | Développement |
| Site Web | Règles | IA - Evaluation | Mode 2 Joueurs | IA - Evaluation | IHM |
| Règles | Déplacement |  |  |  |  |
| Déplacements |  |  |  |  |  |

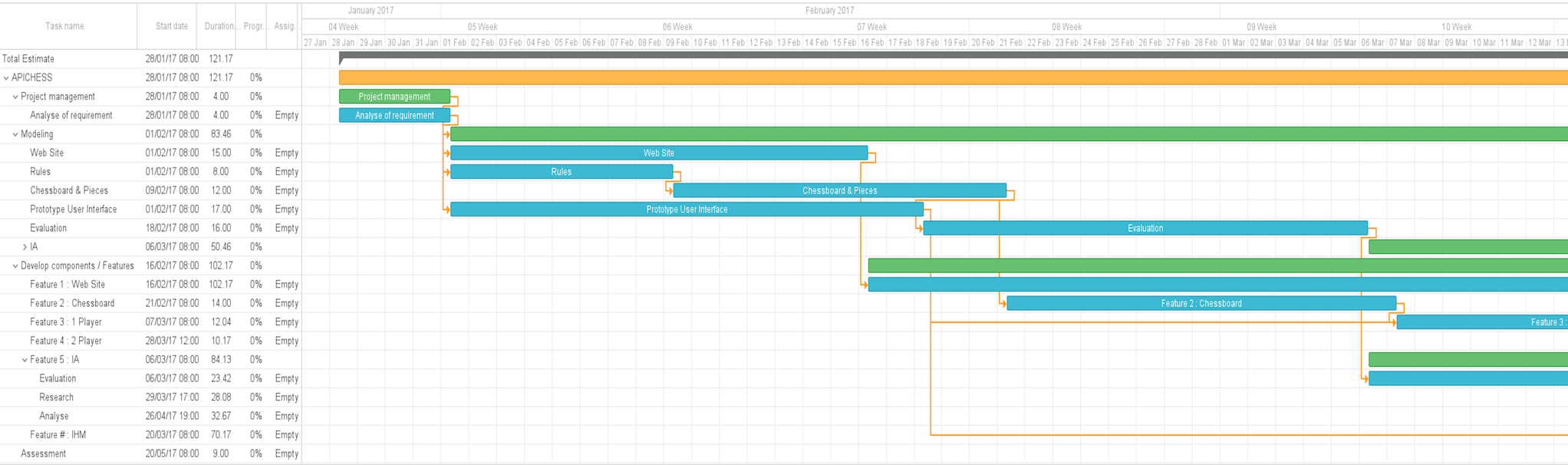
Conjointement à cette répartition le groupe travaillera ensemble à la résolution de l’évaluation du plateau, de la modélisation et du développement de la recherche et de l’analyse du meilleur coup.

### Diagramme de PERT

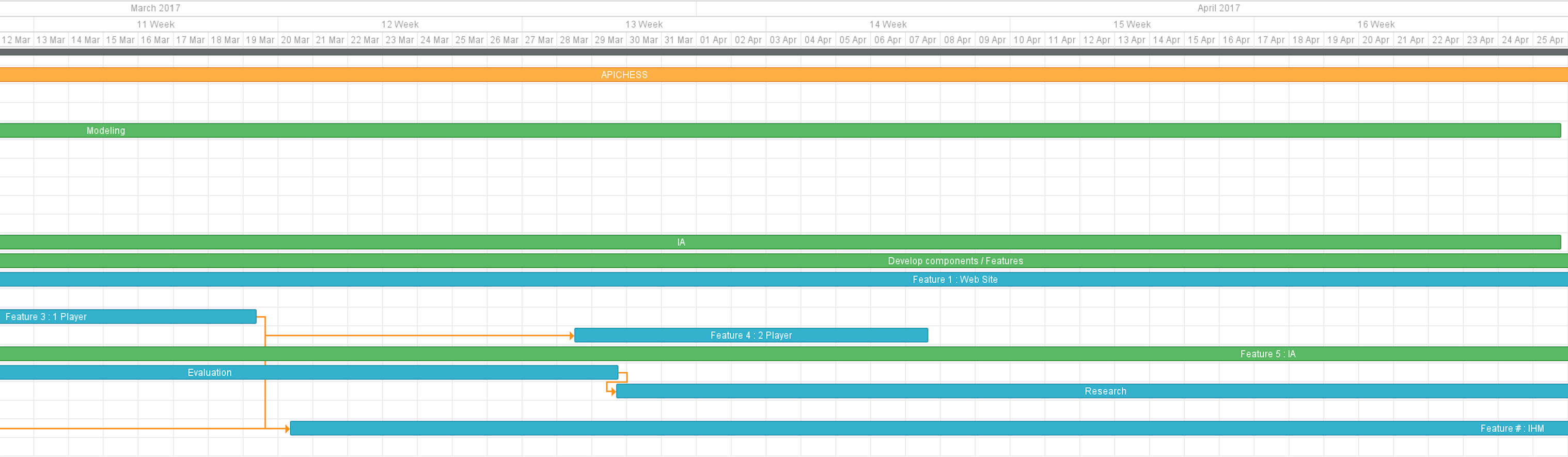


### Diagramme de GANTT

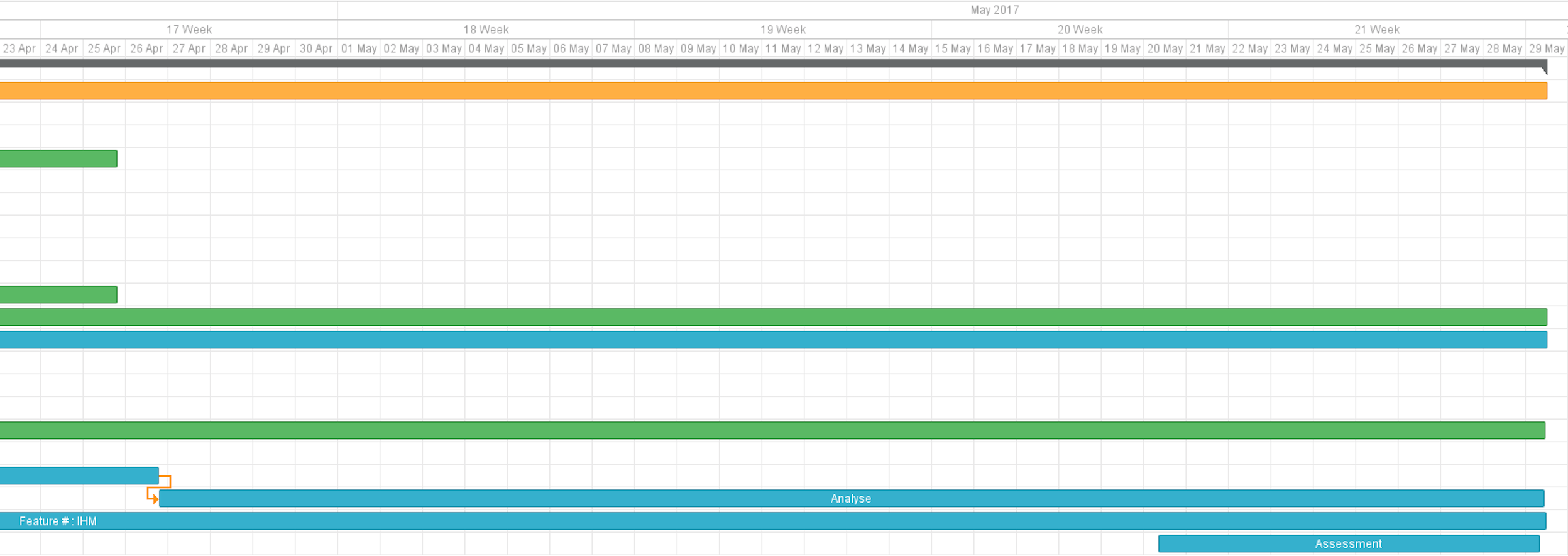
1. Phase 1 : 28/01 – 13/03



1. Phase 2 : 13/03 – 24/04



1. Phase Finale : 24/04 – 29/05



# Repartition des tâches

Pour cette soutenance finale, nous avions décidé d’optimiser le projet délivré à la soutenance 2 et de rejoindre globalement la répartition des tâches faites dans le cahier des charges.

La répartition des tâches qui a été décidée pour cette soutenance finale est celle-ci :

Gravaillac Jean-Charles

Mission :

De par son investissement depuis la première soutenance d’optimiser et de terminer l’interface graphique.

Pour ce faire, les missions qui lui ont été confiées étaient :

* Refacto son code. Celui-ci étant présent dans le fichier main.c doit être dans un fichier séparé ui.c avec son header attitré ui.h.
* Corriger les erreurs dont celle-ci disposait
* Erreur de segmentation en cas de touche ‘esc’ en lieu de quitter le jeu ou de déplacement de la souris
* Allongement du délai de l’AI d’environ 5 secondes alors que celle-ci choisis son coup en moins d’une seconde
* Affichage à l’écran des pièces perdues pour chaque joueur

Hautier Julien

Mission

Refactoriser le code tout en le nettoyant et corrigeant ses erreurs. Puis pour mission seconde de développer les fonctions permettant de créer tous les nœuds fils ainsi que celle nécessaires au fonctionnement de l’algorithme de recherche de l’intelligence artificielle.

Meliani Keyvan

Mission

Réfléchir sur la structure de donnée établie à la deuxième soutenance et de travailler l’Algorithme Min-Max, Alpha-Beta ainsi que leur variante respective et de les faire comprendre au groupe. Conjointement à cela, en tant que chef de projet, il a dû choisir l’implémentation de l’algorithme de recherche et d’évaluation du meilleur coup sur lequel le groupe allait se diriger et de le développer.

# Récit de la réalisation

## Joies

Gravaillac Jean-Charles :

Premier développement d’interface graphique couronné d’une multitude de segfault. Ma plus grande joie a été la fois où le jeu est apparu devant moi. J'ai pris du plaisir à choisir les images, à faire le menu et de devoir réfléchir en amont.

Meliani Keyvan :

J’ai adoré travailler sur ce projet. Il fut très intéressant et enrichissant et m’a permis de confirmer le domaine dans lequel je voudrais travailler à savoir, évidemment, l’intelligence artificielle. Honnêtement je n’ai éprouvé que de la joie en travaillant sur APICHESS. Essayer en permanence d’anticiper les éventuelles erreurs qui pourraient survenir a été un véritable challenge. De même pour le management et la direction du groupe qui fut d’une part laborieux mais d’autre part a permis à certains moments des résultats qui fut au-dessus de mes attentes. Ma première grande joie a été de voir l’IA jouer pour la première fois.

Hautier Julien :

J’ai surtout aimé le coté algorithmique du projet. De sans cesse devoir « Divide and Conquer » le projet, de réfléchir sans cesse à optimiser le code afin de pouvoir avoir le jeu le plus fluide possible.

## Peines

Gravaillac Jean-Charles :

Ma plus grande peine a été la multitude de segfault qui a suivi, la conversion des coordonnées graphiques, et l’optimisation de l’interface graphique du au ralentissement de SDL\_Event qui considérait en permanence un mouvement de souris comme un évènement.

Meliani Keyvan :

Il fut très difficile de concilier le lancement et la direction de mon entreprise ainsi que ceux de APICHESS.

Hautier Julien :

Il fut très difficile de devoir assembler et refactoriser l’entièreté du code le dernier jour. De plus le développement massif des règles et des pièces fut long et fastidieux. La rencontre d’un segfault sur la mise en place finale de l’intelligence artificielle.

## Problèmes rencontrés

Nous avons tout d’abord été surpris par la durée de la période avant cette soutenance finale.

Et il fut difficile de concilier la courte période de cette soutenance avec les révisions des partiels.

Nous aurions aimé mieux gérer notre temps afin de mettre en place une meilleure fonction d’évaluation. Nous avons pensé à pondéré d’avantage les pièces telles que tour, fou, dame lorsque la partie est avancée et que le nombre de pièces ennemies restantes est basse.

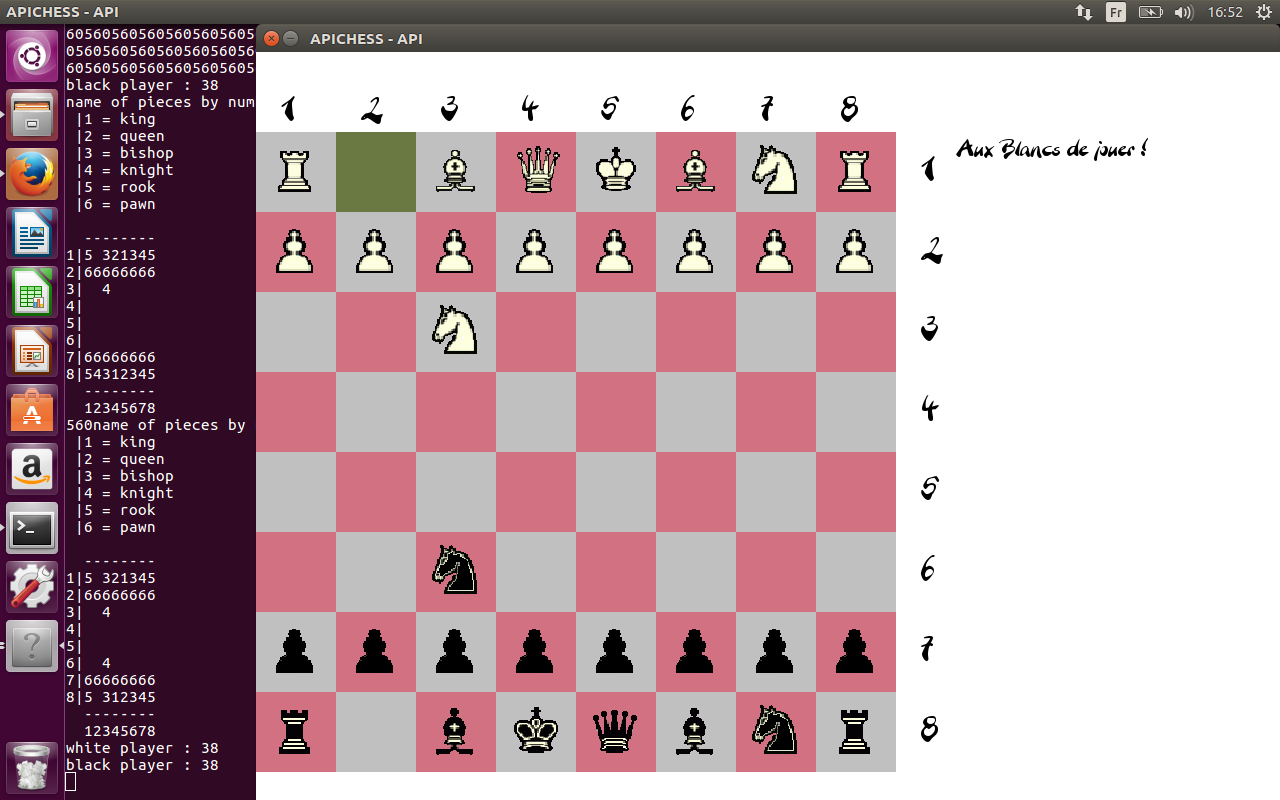
Segfault. Ce produisait lors d’un event lorsqu’on essayait de quitter le jeu ou le menu par le biais de la touche échappe ou lors du clic sur la croix rouge. Problème dû à la tentative du programme d’accéder à une SDL\_Surface qui était toujours à nulle. L’ajout d’un exit avant le Break tout en veillant à bien quitter la SDL nous a permis de résoudre ce problème.

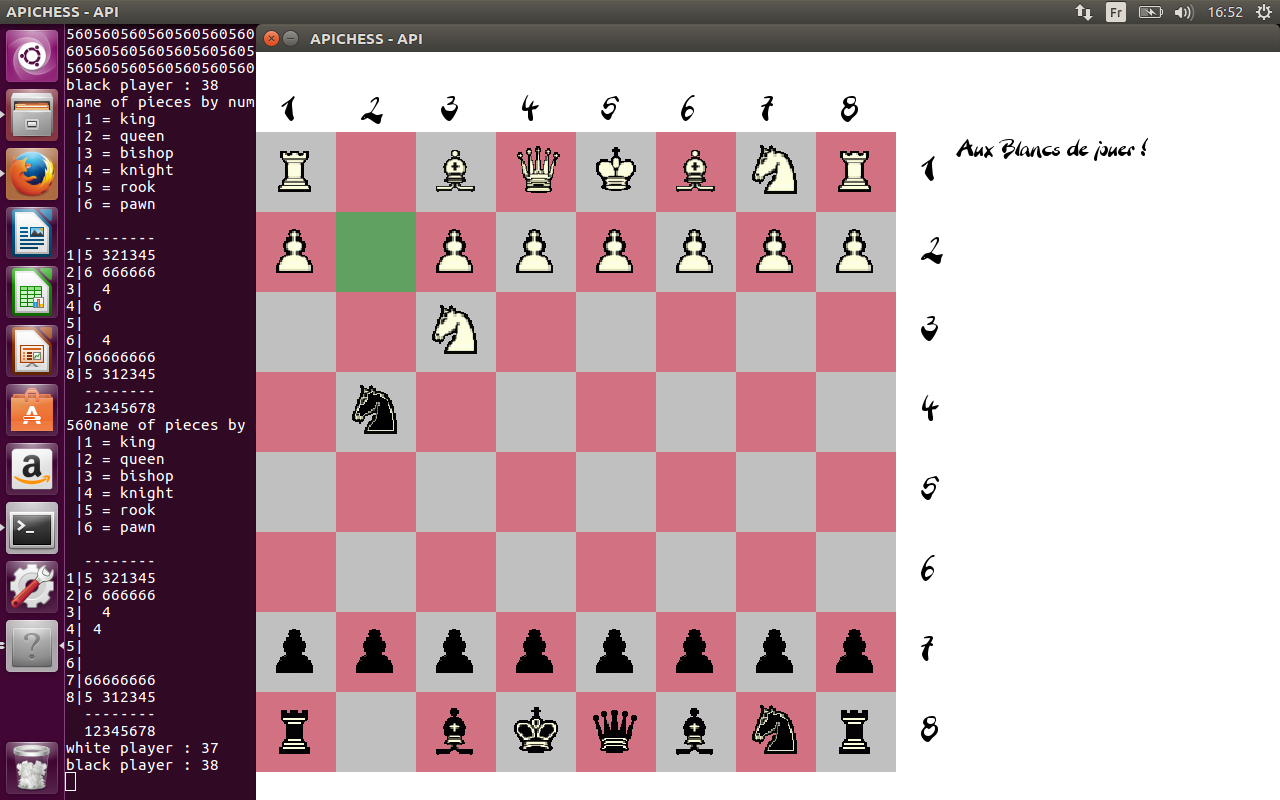
Vitesse d’exécution côté Joueur. La vitesse d’exécution côté joueur semble dépendant du système sur lequel on fait tourner le programme. Nous avons essayé de résoudre ce problème en en portant notre intérêt sur les fonctions de la librairie SDL en particulier SDL\_BlitSurface car nous pensions envoyer les mauvais paramètres ce qui aurait entrainé un ralentissement du traitement des données dû à un envoie de données excessives. Apres recherche approfondi le problème semblerai venir sur l’acquisition et la conversion des coordonnées géographique pour le mouvement. Ce problème se situerait donc au niveau de SDL\_WaitEvent et du switch (event.type). Nous avons donc décider de remplacer SDL\_WaitEvent par SDL\_PoolEvent afin de savoir si un event est en attente. Le tout mis dans une boucle while afin de pouvoir dépiler de tous les events pour une itération de la boucle du jeu au lieu dépiler un à un par itération du jeu. Nous avons aussi pris soin d’éviter les events fantômes grâce au type NoEvent affecter à la variable SDL\_Event.

# Annexes

## Impression d’écran







## Dessin d’origine

### Cases de l’échiquier



Case blanche



Case noire



Case sélectionnée



Pièces blanches



Pièces noires

## Structures

Nous avons redéfini les structures nécessaires à l’intelligence artificielle afin que celle-ci soit le plus optimiser possible. Un allégement maximum et une anticipation de fonction d’évaluation plus perfectionnée prennant en compte le tour actuel et le nombre de pièces restante.

struct AIDatas {

int turn;

int pLeftW;

int pLeftB;

int eval;

AISquare \*\*AIMat;

};

struct AISquare {

int color;

int type;

int value;

}

## Algorithmes

### Intelligence artificielle

Déclaration

INT\_MIN et INT\_MAX valent respectivement la valeur minimale d’un entier (représentant moins l’infini) et la valeur maximale d’un entier (représentant plus l’infini)

H\_MM\_MAX représente une définition d’entier statique fixée à la valeur 3.

#### Fonction min-max

Notre algorithme MinMax est le suivant :

Fonction min\_max (d : ref datas vecteur AIDatas, h entier, maxi booléen) : entier

Variables :

Entier best\_move

Entier value

Début

Si (h==0 ou \*datas ==null) alors

Retourne eval()

Fin Si

Si (maxi) alors

best\_move := INT\_MIN

Pour tous les coups possible faire

value := min\_max(coup, h-1, faux)

best\_move := maximum(best\_move, value)

Fin Pour

retourne best\_move

Sinon

best\_move := INT\_MAX

Pour tous les coups possible faire

value := min\_max(coup, h-1, vrai)

best\_move := maximum(best\_move, value)

Fin Pour

retourne best\_move

Fin Si

Fin algorithme min\_max

Fonction min\_max\_call(d : ref datas vecteur AIDatas, maxi booléen) : entier

Début

retourne min\_max(\*datas, H\_MM\_MAX, maxi)

Fin algorithme min\_max\_call

#### Fonction alpha\_beta

Notre algorithme AlphaBeta est le suivant :

Fonction alpha\_beta\_max (d : ref datas vecteur AIDatas, h entier, alpha entier, beta entier) : entier

Variables :

Entier best\_score

Début

Si (h==0 ou \*datas ==null) alors

Retourne eval()

Fin Si

Pour tous les coups possible faire

best\_score = alpha\_beta\_min(coup, h-1, alpha, beta)

Si (best\_score >= beta) alors

Retourne beta;

Fin Si

Si (best\_score > alpha) alors

alpha = best\_score

Fin si

Fin Pour

Fin algorithme alpha\_beta\_max

Fonction alpha\_beta\_min (d : ref datas vecteur AIDatas, h entier, alpha entier, beta entier) : entier

Variables :

Entier best\_score

Début

Si (h==0 ou \*datas ==null) alors

Retourne eval()

Fin Si

Pour tous les coups possible faire

best\_score = alpha\_beta\_max(coup, h-1, alpha, beta)

Si (best\_score <= beta) alors

Retourne alpha;

Fin Si

Si (best\_score > alpha) alors

beta = best\_score

Fin si

Fin Pour

Fin algorithme alpha\_beta\_min

Fonction alpha\_beta\_call(d : ref datas vecteur AIDatas, maxi booléen) : entier

Début

retourne alpha\_beta\_max(\*datas, H\_AB\_MAX, INT\_MIN, INT\_MAX)

Fin algorithme alpha\_beta\_call

Fonction eval(d : ref datas vecteur AIDatas, color entier) : entier

Variables :

Entier count

Début

count := 0

Pour i allant de 1 à 8 faire

Pour j allant de 1 à 8 faire

Si(datas->AIMat[i][j].color == color)

count += datas->AIMat[i][j]

Fin Si

Fin Pour

Fin pour

Retourne count

Fin algorithme eval