Rapport de Projet - Première Soutenance

tAItris

Projet du quatrième semestre de l'EPITA

Groupe S4MasterRace :
Thomas Michelot (michel_k)
Julien Papini (papini_j)
Nicolas Loyau (loyau_n)
Sevan Murriguian–Watrin (murrig_s)

25 février 2018

Table des matières

1 Introduction

Nous sommes S4MasterRace, et ce rapport de soutenance aura pour but de vous montrer l'avancement de notre projet depuis la validation de notre cahier des charges.

Notre projet, tAItris, consiste en la réalisation d'une intelligence artificielle réalisé a l'aide du langage C et capable de faire le plus haut score possible de manière autonome lors d'une partie de Tetris.

Afin de faire fonctionner notre projet et de permettre à notre intelligence artificielle de faire correctement son travail, nous avons dû commencer par refaire une implémentation de Tetris en s'inspirant du jeu de base.

Le jeu est actuellement jouable mais peut toujours être cible à améliorations ou à ajustements pour correspondre au mieux aux besoins de notre intelligence artificielle.

Notre intelligence artificielle permet pour l'instant uniquement de trouver la meilleure position avec la meilleure rotation possible pour placer la pièce qui vient d'apparaître dans la grille de jeu en cours afin de marquer un maximum de points et de faire durer la partie au maximum.

Nous avons également un site web dédié à notre projet de disponible afin de partager des informations sur le projet en général ainsi que sur son état d'avancement actuel.

2 Tetris

FIGURE 1 – Ecran Principal

L'objectif de notre projet étant de réaliser une intelligence artificielle capable de jouer d'elle même au jeu Tetris tout en essayant de faire le score le plus haut, il nous faut donc évidemment un jeu Tetris de base afin qu'elle puisse exercer ses fonctions normalement.

C'est pour cela que nous avons d'abord dû recréer une implémentation du jeu Tetris basique afin de pouvoir faire fonctionner et tester notre intelligence artificielle.

Il s'agit d'un Tetris classique fonctionnant comme l'original sorti en 1984, dans lequel le but est de réaliser des lignes complètes en déplaçant des pièces de formes différentes appelées tetrominos qui apparaissent en haut de l'écran puis défilent sur celui-ci jusqu'à rencontrer un obstacle à sa chute.

Le joueur peut, pour arriver à ses fins avoir un certain contrôle sur les tetrominos en les faisant pivoter vers la droite et la gauche ainsi que de la translater dans les mêmes directions afin de positionner la figure à l'endroit souhaité.

Pour obtenir une meilleure fluidité de jeu, le jouer à également la possibilité de faire descendre instantanément le tetromino jusqu'au prochain obstacle en dessous d'elle afin d'accélérer le jeu.

Les lignes complétées vont ensuite disparaissent tout en rapportant des points et le joueur pourra ainsi de nouveau remplir les cases qui ont été libérées.

Le joueur peut réaliser un "Tetris" en complétant quatre lignes en un seul coup grâce au tetromino I placé dans un espace de 4 trous alignés verticalement, c'est le mouvement qui rapporte le plus de points.

La partie se déroule sur une grille de 10 carrés de large pour 22 carres de haut et la partie se termine lorsqu'un tetromino dépasse du haut de l'écran.

L'objectif pour le joueur est donc de réaliser le plus haut score possible en assemblant un maximum de lignes pour éviter la montée des tetrominos entraînant donc la fin de la partie. Plus le joueur fait de lignes en même temps, plus il gagnera de points.

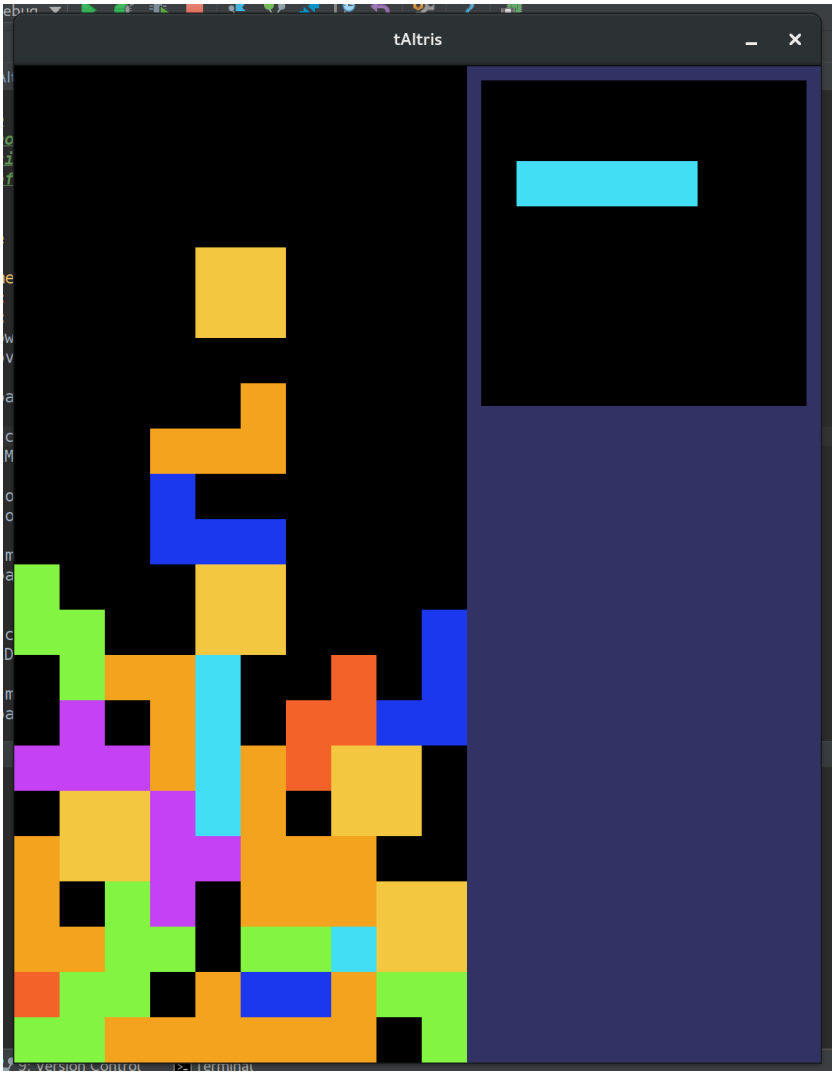

FIGURE 2 – Implémentation de Tetris

Les seules attentes de notre version de Tetris pour cette première soutenance est qu'elle puisse être utilisée par notre intelligence artificielle afin de pouvoir la mettre en marche et effectuer moultes tests.

Le jeu pourra ensuite être change et amélioré lors de l'avancement du projet et au cours des futures soutenance afin d'avoir un rendu final mieux finalise et qui corresponde au mieux aux besoins du projet et de notre IA.

2.1 SDL

Nous avons choisit la bibliothèque logicielle libre SDL (Simple Direct-Media Layer) pour l'affichage de notre jeu. Cette bibliothèque permet de réaliser de l'affichage simplement.

2.1.1 SDL_Surface

Une SDL_Surface est un élément principal dans la programmation en utilisant la SDL. C'est une structure qui contient une image sous forme de tableau, ainsi que sa taille et les différents paramètres associe a cette surface. C'est en manipulant cette structure que nous parvenons a créer l'image qui sera ensuite affichée au joueur. Il est aussi possible de travailler avec plusieurs SDL_Surface. Ces SDL_Surface seront ensuite ajoutées a une SDL_Surface principale qui représente en général l'écran a afficher.

```
typedef struct SDL_Surface {
    Uint32 flags;                           /* Read-only */
    SDL_PixelFormat *format;                /* Read-only */
    int w, h;                               /* Read-only */
    Uint16 pitch;                           /* Read-only */
    void *pixels;                           /* Read-write */

    /* clipping information */
    SDL_Rect clip_rect;                     /* Read-only */

    /* Reference count -- used when freeing surface */
    int refcount;                           /* Read-mostly */
} SDL_Surface;
```

2.1.2 SDL_Rect

Une structure de type SDL_Rect stocke des coordonnées. C'est avec cela que l'ont peut "blit" les différentes SDL_Surface dans la SDL_Surface principale.

```
typedef struct{
    Sint16 x, y;
    Uint16 w, h;
} SDL_Rect;
```

2.1.3 SDL_TTF

C'est un module de la SDL qui permet d'afficher du texte. Elle permet de d'écrire du texte sur des SDL_Surface, qui seront ensuite appliquées dans la SDL_Surface principale. C'est avec cela que nous affichons les différents éléments variable de l'interface, comme le score par exemple.

2.1.4 SDL_Image

C'est un module de la SDL qui permet de charger des images au format PNG, par exemple. C'est grâce a cette élément que nous avons pu charger le fond de notre Tetris ou bien les blocs qui compose les Tetromino. C'est en effet plus simple de charger une image que de la recréer depuis une SDL_Surface vide, en notant que cela est parfois non réalisable.

2.1.5 Gestion des événements

La gestion des événements est aussi une fonction de la SDL. En effet, il est possible "d'écouter" un ensemble définit d'input. La boucle principale du jeu est basée sur cela, on "écoute" les différentes touches de déplacement et lorsque qu'une touche est pressée, on réalise l'action souhaitée. Cela fonctionne similairement pour le redimensionnement de la fenêtre ou bien la fenêtre de celle-ci.

2.2 L'implémentation

2.2.1 Les éléments du Tetris

Les éléments principaux du Tetris sont tous représentés par des structures dans le code. Il y a donc une structure pour les pièces, ainsi que pour le tableau de jeu. Le jeu est organisé autour de tableaux, un principal, et un pour chaque pièce.

2.2.2 Le jeu

Le jeu repose, comme cela est précisé au dessus, sur une boucle principale. A chaque exécution, nous "écoutons" les différentes touches. Chaque action entraîne des événements. Bien entendu, en dehors des différents événements, il y a aussi des actions prédéfinies qui ont un rapport avec le temps. Il est nécessaire de faire descendre la pièce à chaque tour, et ce en fonction d'une valeur prédéfinie. Cette valeur sera probablement amenée à évoluer. Dans le Tetris, en fonction du nombre de lignes qui ont été complétées par l'utilisateur, la vitesse des Tetrominos augmente. Cela rend ainsi le jeu plus complexe. Le temps laissé à l'utilisateur pour placer le Tetromino est donc réduit au fur et à mesure.

Le but du joueur est de ne pas laisser le tableau se remplir de Tetrominos. Pour cette vérification, nous effectuons une simple contrôle sur la structure "board" du jeu, qui contient le tableau représenté à l'écran. Les tableaux sont donc nécessaires aux différentes vérifications que nous effectuons.

En effet, avant l'affichage, tout les éléments du jeu sont stocké sous forme de tableau. Lorsque que le jeu détecte que la pièce ne peut plus descendre, il fixe la pièce dans le tableau principal, et génère une nouvelle pièce. Cette nouvelle pièce est sélectionné de façon aléatoire parmi un set prédéfinie de pièce.

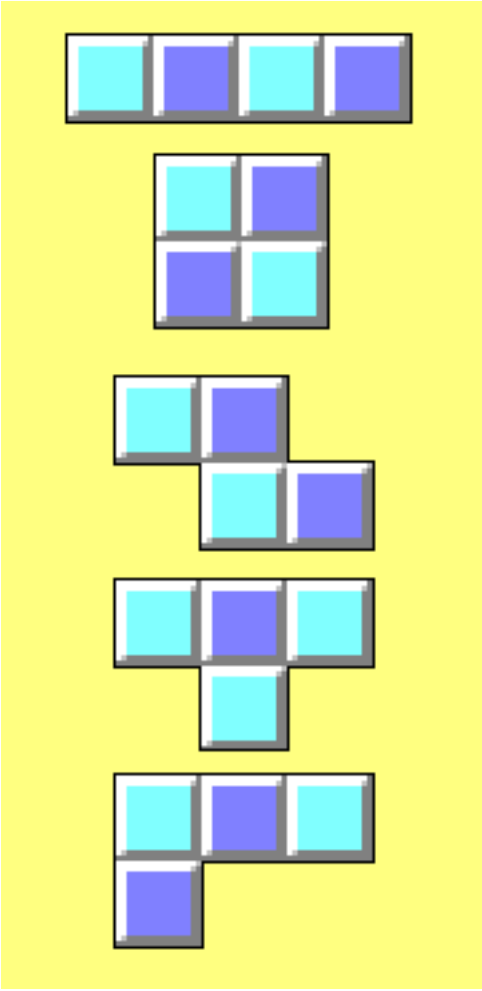

FIGURE 3 – Les Différents Tetramino

2.3 Les prochaines soutenances

Pour la suite du projet, nous allons continuer à améliorer le Tetris, comme cela est précisés au dessus, cette version est une version intermédiaire afin de de tester l'ia.

Lorsque que l'on aura aboutit à un Tetris final et fonctionnel, on peut penser à ajouter des fonctionna litées à notre projet. Nous pourrions, par exemple, ajouter des options afin de changer les différentes touches utilisées pour jouer. D'autres modes de jeu peuvent aussi intervenir, en effet, depuis le premier Tetris, les développeurs ont ajouté divers mode intéressant. Il existe par exemple des modes IA contre IA. Cela reste bien évidement des suggestions.

3 Intelligence Artificielle

Pour le projet tAItris, l'intelligence artificielle est le point clé puisque c'est sur celle-ci que repose l'entièreté du projet. Il est donc nécessaire d'en concevoir une la plus efficace possible et avec un comportement au plus proche de ce qui est attendu d'elle.

Pour générer notre intelligence artificielle nous utilisons un algorithme génétique pour toujours trouver les meilleurs emplacements dans lesquels jouer. Ces positions sont trouvées par une multitude de facteurs calculé réalisés en fonction de l'état de la grille au moment ou la nouvelle pièce apparaît.

L'intelligence artificielle va donc calculer un rang qu'elle va attribuer pour chaque position et chaque rotation que la pièce pourrait faire puis choisi la position et rotation qui a obtenu le meilleur rang. Ce range se calcule de la manière suivante :

$$a*(AggregateHeight)+b*(CompleLines)+c*(Holes)+d*(Bumpiness)$$

Où a, b, c et d sont des paramètres constants et AggregateHeight, CompleteLines, Holes, Bumpiness sont des features qui sont calculées par rapport à l'état de la grille à un certain moment.

Aggregate Height permet de calculer la "hauteur totale" de la grille en addition la hauteur de chaque colonne de la grille. On veut cette valeur la plus petite possible pour assurer plus de place pour continuer à jouer.

CompleteLines calcule simplement le nombre total de lignes complétées sur la grille. On veut cette valeur la plus haute possible car le but de l'intelligence artificielle est de compléter un plus grand nombre de lignes possible pour un plus haut score ainsi que pour libérer le plus de place possible.

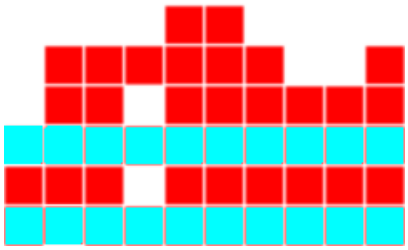

FIGURE 4 – Complete Lines

Holes va compter le nombre de trous ayant été recouverts avec une autre pièce par dessus et donc inaccessibles. Ces trous sont beaucoup plus dur à supprimer car on va devoir compléter toutes les lignes au dessus du trou avant de pouvoir y avoir accès. On veut donc cette valeur la plus petite possible.

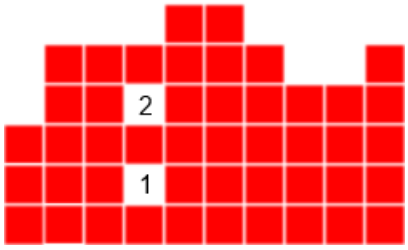

FIGURE 5 – Holes

Enfin, Bumpiness va permettre de repérer les "puits" présents sur la grille quand une colonne est beaucoup plus basse que ses colonnes voisines. On les detecte en additionnant la valeur absolue de la différence entre deux colonnes adjacentes.

FIGURE 6 – Exemple de puit

L'intelligence artificielle doit donc calculer toutes les features présentées ci-avant sur la grille actuelle puis calculer le rang à partir de la formule présentée plus tôt afin de trouver la position optimale où positionner le tetromino arrivant.

Pour trouver les meilleures valeurs possibles pour les constantes a, b, c et d, il faut faire fonctionner un algorithme génétique en appliquant des valeurs aléatoires à ces variables pour en ressortir les valeurs de a, b, c et d avec lesquels on trouve les meilleurs rangs.

Pour la prochaine soutenance il serait aussi intéressant de mettre une place une feature afin de savoir si il est possible de réaliser un Tetris en réalisant 4 lignes d'un coup à l'aide d'un tetromino I dans un espace de 4 trous alignés verticalement car c'est le mouvement rapportant le plus de points et permettant de libérer le plus de place en le moins de temps.

Ce type de mouvement est une amélioration du "niveau" de notre IA. Le but est ici de ne pas placer bêtement les pièces comme le ferai un joueur débutant. En effet avec cette façon de jouer, il est très difficile, voire impossible de finir une partie.

4 Site Web

Afin de faire des comptes rendus de l'état de notre projet et délivrer un certain nombre d'informations a son sujet, nous avons décide de mettre en place un site web pour permettre de suivre l'avancement du projet ainsi que trouver toutes sortes de renseignements sur celui-ci.

FIGURE 7 – Le site Web

Une présentation en forme de notre projet est également disponible sur le site internet afin de présenter l'objectif du projet ainsi que ses fonctionnalités actuelles et celles qui sont prévues a l'ajout. Cette présentation est susceptible d'être modifiée plus tard en cas d'ajout de fonctions supplémentaires par exemple.

Notre site web servira également a accueillir de la documentation nécessaire en rapport avec le projet comme de l'aide sur la façon de compiler et d'installer le programme.

Le site internet est construit à partir du logiciel Jekyll qui est un moteur en Ruby permettant de créer des pages internet et des articles assez facilement et avec un visuel correct à partir de fichiers MarkDown. C'est le même moteur utilisé par le célèbre site github et est donc un environnement assez familier aux membres du groupe.

L'objectif pour les prochaines soutenances a venir est d'avoir un site qui restera a jour au fur et a mesure de notre avancée dans notre projet en y ajoutant toutes les informations pouvant être importantes en rapport avec le projet comme par exemple notre avancée ou encore une liste des fonctionnalités disponibles.

5 Conclusion

Pour cette première soutenance, nous avons bien amorcé la conception de notre projet en avançant sur un peu toutes les taches que nous avions définies précédemment dans le cahier des charges.

Au terme de cette soutenance, nous avons donc un projet qui avance à un rythme relativement convenable et en restant assez proche des délais que nous nous étions fixés.

En effet nous avons pour la première soutenance notre propre implémentation du jeu Tetris qui nous servira de base et qui est un élément principal de ce projet.

Nous avons aussi un site internet fonctionnel pour présenter notre projet et y répertorier un maximum d'informations à propos de celui-ci.

Nous avons cependant un léger retard sur l'implémentation de notre intelligence artificielle qui sera rattrapable assez facilement pour la prochaine soutenance.
Nous n'avons pour l'instant qu'un début d'intelligence artificielle qui nous servira par la suite de base pour notre programme final.

Pour la suite et la prochaine soutenance, nous allons donc avoir à peaufiner le Tetris que nous avons actuellement pour avoir une version la plus fidèle à nos attentes possible et qui fonctionne le mieux possible avec notre modèle d'IA.

Il nous faudra également avancer sur l'intelligence artificielle afin d'en obtenir une qui devra être plus poussée et avec un réseau de neuronnes.

Nous allons également devoir mettre à jour le site internet de notre projet avec toutes les nouvelles informations pouvant être ajoutées entre temps.

Table des figures