Université Sidi Mohamed Ben Abdallah
Faculté des Sciences Dhar Mahraz – Fès

Rapport de Projet Tutoré

Semestre S6

Filière Licence Fondamentale en Sciences Mathématiques et Informatique

Intitulé :

Conception et Développement d'un jeu vidéo sur Desktop : TETRIS

Réalisé par :
Mohammed EL KHOU
Mohammed KASMI

Sous la direction de :
Pr. Mohammed EL MOHAJIR

Présenté le 23/05/ 2017 Devant le Jury composé de :

Pr. Ahmed ZINEDINE
Pr. Mohamed CHAHHOU
Pr. Mohamed EL FAR
Pr. Mohammed EL MOHAJIR

Année universitaire 2017-2018

Remerciement

Le travail présenté dans cette mémoire a été effectué dans le cadre de la préparation de la licence fondamentale en filière Science Mathématique et informatique à la faculté des sciences Dhar El Mehraz.

Au terme de ce projet, Nous tenons à exprimer notre profonde gratitude et notre immense respect à notre encadrant Monsieur ***Mohammed el mouhajir****, Professeur en informatique à la faculté des sciences Dhar El Mehraz pour sa disponibilité, ses avis éclairés, et ses conseils judicieux. Nos vifs remerciements accompagnés de toute notre gratitude s'adressent également à tous nos professeurs qui ont gardé un œil attentif sur le déroulement et l'avancement de notre projet, donnant toujours des remarques très constructives ainsi pour la confiance qu'ils nous ont fait. Nous tenons à exprimer notre plaisir de travailler sous leurs directives.*

Dédicace

A Nos Très Chers Parents

Tous les mots du monde ne sauraient exprimer l'immense amour que nous portons pour vous, ni la profonde gratitude que nous vous témoignons pour vous pour tous les efforts et les sacrifices que vous avez fournis à fin de nous garantir une bonne instruction. C'est à travers vos encouragements et vos critiques que nous nous sommes réalisés. Nous espérons avoir répondu aux espoirs que vous avez fondés en nous. Nous vous rendons hommage par ce modeste travail en guise de notre reconnaissance éternelle et de notre amour infini. Que Dieu tout puissant vous garde et vous procure la santé, le bonheur et la longue vie pour que vous demeuriez le flambeau illuminant notre chemin. Nous dédions aussi ce travail à nos familles et à tous nos professeurs de la Faculté des Sciences Dhar El Mehraz.

A tous nos collègues, et nous leurs souhaitons le succès et l'atteinte de leurs ambitions pour leurs encouragements incessants…

Résumé

Le travail présenté dans cette mémoire s'inscrit dans le contexte d'un jeu vidéo : TETRIS.

Tetris est un type de jeu vidéo basé sur la réflexion. Son nom provient du puzzle, jeu consistant à mettre en ordre des pièces. Il se différencie des autres jeux de réflexion en contenant invariablement des pièces ou des objets à placer d'une façon ou dans un ordre précis.

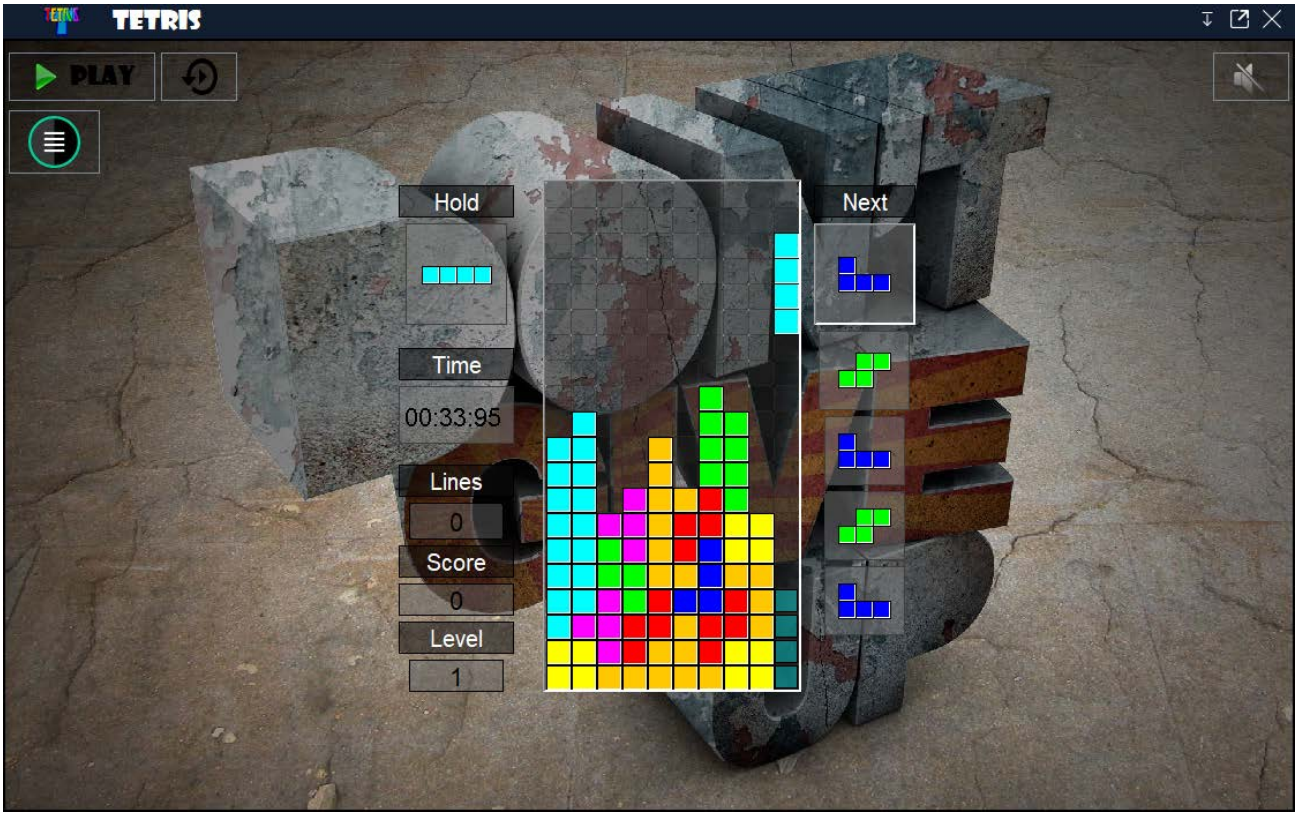
Notre objectif est de concevoir et de mettre en place une application Desktop présente un jeu vidéo Et de définir des méthodes et des interfaces graphiques simples à l'utilisation pour interagir avec notre jeu, s'amuser et en même temps développer l'esprit.

L'ensemble de ces informations ont été traité et analysé en utilisant :

- Un model conceptuel de données(UML).

-Développent des applications par des langages Java.

Enfin la mise en place des applications par les langages JAVA en se basant sur SWING et AWT.



Sommaire

Table des matières

Remerciement 2
Dédicace 3
Résumé 4
Sommaire 5
Table des figures 7
Cahier de Charge 8
1. Problématique 8
2. Les objectifs 9
Chapitre 1 : ETUDE DE L'EXISTANT 10
1. Les jeux et les nouvelles technologies 10
2. Les bienfaits des jeux de vidéos : 12
2.1. Vous deviendrez peut-être plus intelligent grâce à 'Starcraft' 12
2.2. Les jeux vidéo pourraient ralentir le vieillissement 13
2.3. Ils pourraient aider les enfants dyslexiques à mieux lire. 13
2.4. Les jeux vidéo peuvent soulager la douleur 14
2.5. Les jeux et le monde militaire 14
3. Les outils de création de jeux vidéo 19
4. Les langages utilisent dans le développement des jeux 21
4.1. *JAVA* 21
4.2. Le langage C 21
4.3. Le langage C++ 21
4.4. Le langage C# 22
4.5. PHP 22
4.6. PYTHON 22
5. Graphisme 22

5.1. 3D 23
5.2. Le pixel art 23
5.3. Le rétro vintage 23
6. Pourquoi besoin de développer un jeu ? 24
Chapitre 2: Phases d'analyse et de conception 25
1. Introduction 25
1.1. Modèle de cycle de vie d'un logiciel 25
1.2. Méthodologie de conception 26
2. Diagramme de GANTT 26
3. Diagramme de cas d'utilisation 28
4. Diagrammes de Séquence 29
5. Diagramme de classes 31
6. Diagramme Hiérarchique 34
7. Diagrammes d'activités 35
Rejouer 35
Chapitre 3 : Réalisation 36
1. Environnement de développement 36
1.1. Environnement Matériel 36
1.2. Environnement Logiciel 36
1.3. Environnement des langages de développements 41
Chapitre 4 : Présentation du Jeu 43
1. Introduction 43
2. Les interfaces graphiques 43
- Lancement du jeu 43
- L'interface d'accueil 45
- About-us 46
- Help 48
- Option 49
- Home 53
- Le plateau de jeu 54
- Le sous-plateau 55

- Les pièces 56
- Les mouvements latéraux 57
- Les rotations 58
- Commandes du jeu 59
- Suppression des lignes 59
- Les Niveaux 60
- Game Over 61
- High Scores 63

3. Conclusion 64

Conclusion générale 65

Webographie et bibliographie 66

Table des figures

Figure 1 : Capture d'application. 4

Figure 2 : Les Pieces. 8

Figure 3 : Java Icon. 13

Figure 4 : 3D. 15

Figure 5 : Pixel Art. 15

Figure 6 : Retro Vintage. 16

Figure 7 : Modèle en cascade. 18

Figure 8 : Méthodologie de conception adoptée. 18

Figure 9 : diagramme de Gantt. 19 - 20

Figure 10 : diagramme de cas d'utilisation. 21

Figure 11 : diagramme de Séquence. 22

Figure 12 : diagramme de classes. 24 - 25

Figure 13 : diagramme Hiérarchique. 26

Figure 14 : diagramme d'activités. 27

Figure 15 : Capture d'application. 35 - 55

Figure 16 : Les pièces. 48 - 49

Figure 16 : Les rotations des pièces. 50

Cahier de Charge

1. Problématique

Tetris est un jeu vidéo de puzzle conçu par Alekseï Pajitnov á partir de juin 1984 sur Elektronika 60. Tetris met le joueur au défi de réaliser des lignes complètes en déplaçant des pièces de formes différentes, les tétriminos, qui défilent depuis le haut jusqu'au bas de l'écran. Les lignes complétées disparaissent tout en rapportant des points et le joueur peut de nouveau remplir les cases libérées. Le jeu n'a pas de fin : le joueur perd la partie lorsqu'un tétrimino reste bloqué en haut. Il doit donc résister le plus longtemps à la chute continue des tétriminos, afin de réaliser le meilleur score.

Nous vous demandons d'implémenter une version de ce jeu, selon les règles suivantes.

Le jeu est principalement composé d'un champ de jeu à largeur fixe, où des pièces de formes différentes, descendent du haut de l'écran. Durant cette descente le joueur peut uniquement déplacer les pièces latéralement en utilisant les touches fléchées gauche et droite et leur faire effectuer une rotation de 90 degrés sur elles-mêmes à chaque fois qu'il appuie sur la touche fléchée haute, jusqu'á ce qu'elles touchent le bas du champ de jeu ou une pièce déjà placée. Le joueur ne peut ni ralentir la chute des pièces, ni l'empêcher, mais il peut l'accélérer en utilisant la touche fléchée vers le bas.

Le but pour le joueur est d'effacer le plus de lignes possibles, afin de garder de l'espace pour placer les futures pièces. Une fois une ligne complétée, elle disparaît, et les blocs placés au-dessus chutent d'un rang. Si le joueur ne parvient pas à faire disparaître les lignes suffisamment vite, l'écran peut alors se remplir jusqu'en haut. Lorsqu'un tétrimino dépasse du champ de jeu, et empêche l'arrivée de

tétriminos supplémentaire, la partie se termine. Le joueur obtient un score, qui dépend essentiellement du nombre de lignes effacées lors de la partie.

Le jeu ne se termine donc jamais par la victoire du joueur. Les pièces de Tetris, sur lesquelles repose entièrement le jeu et que vous devrez représenter, sont les suivants :

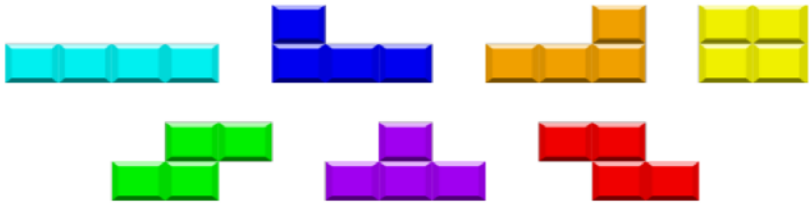


FIGURE 1 – Tetriminos.

Nous vous demandons donc d'implémenter à l'aide du module turtle, une simple version graphique du jeu. L'interface graphique, qui permettra à l'utilisateur d'interagir avec votre programme de manière plus conviviale, est totalement laissée à votre discrétion (vous pouvez par exemple choisir la couleur des pièces, comment représenter la grille et sa largeur).

Chaque fois qu'une partie se termine, votre interface graphique devra imprimer dans le champ de jeu combien de lignes l'utilisateur a effacées au total pendant le match et lui donner aussi la possibilité de démarrer une nouvelle partie ou de quitter la plateforme de jeu.

Vidéo de présentation du jeu sur YouTube

https://www.youtube.com/watch?v=YMwlsbpjgXE

2. Les objectifs

Tetris se joue sur un quadrillage, appelé la **matrice**, possédant une gravité (du haut vers le bas). Des formes géométriques, composées de quatre carrés, tombent depuis le haut de l'écran.

- Réalisation d'une interface graphique qui permettra à l'utilisateur d'interagir avec notre programme de manière plus conviviale :

- Le contrôle de chute des pièces.

- Changement des configurations du jeu (music, arrière-plan, niveau de difficulté, « high scores »).

Chapitre 1 : ETUDE DE L'EXISTANT

1. Les jeux et les nouvelles technologies

1-qu'est-ce qu'un jeu ?

Nous pouvons définir le « mot jeu » de différentes manières. En effet, nous pouvons être amenés à le définir par ce qu'il n'est pas. Un jeu ne doit pas être perçu comme un travail mais comme distraction ou le joueur ne perçoit pas de revenus pour ce qu'il fait. Cependant, il y a des situations ou les jeux et le travail ne peuvent se substituer l'un à l'autre comme dans les sports. Le jeu n'est pas un art en lui-même. De plus, malgré le fait que l'art est difficile à définir, il reste néanmoins un moyen d'exprimer ses idées et en ce sens le jeu peut être également perçu comme tel. Nous pouvons dire que les jeux peuvent former probablement le troisième pôle d'activité humaine en étant à la fois une activité structurée ou le plaisir est mélangé à l'instruction.

Jeu est un système ou les joueurs s'engagent dans un conflit artificiel, défini par des règles, avec des résultats faciles à évaluer

Chris Crawford, entrepreneur de jeu vidéo, définit les jeux selon quatre facteurs bien précis qui sont la représentation, l'interaction, le conflit et la sécurité. Ce qui signifie que les résultats des jeux sont moins durs que la situation du jeu.

Katie Salen et Eric de Zimmerman affirment qu'un jeu est un système ou les joueurs s'engagent dans un conflit artificiel, défini par des règles, avec des résultats faciles à évaluer.

→ les différents critères de synthétisation d'un jeu :

Jasper Jules, dans son livre Demi-Vrai, a créé une liste de critères qui tente de synthétiser les différentes définitions :

1. Règles : Les jeux sont basés sur des règles

2. Les variables ainsi que les résultats sont faciles à évaluer .

3. La mise en valeur des résultats : Les résultats du jeu sont différents et les valeurs assignées différent également, quelques positif et quelque négatif.

4. Effort de joueur : Le joueur exerce l'effort pour influencer les résultats. (Les jeux défient)

5. Le joueur est attaché au résultat : Le joueur est attaché avec émotion au résultat du jeu car sa victoire ou sa perte en dépend donc celui-ci sera content ou mécontent

6. Conséquences négociables : Le même jeu peut être joué avec ou sans les conséquences de la vraie vie.

En effet, en appliquent ces conditions, il est possible de déterminer s'il s'agit d'un jeu ou non. Par exemple, un jeu électronique appelé «SimCity» ne peut pas être considéré comme étant un jeu selon ces critères car il s'agit d'une stimulation ou il n'y a pas de vrais résultats ni de fin.

→ Types de jeux video:

Il existe plusieurs type de jeu,on peut citer quelques unes :

Les jeux de stratégie.

Les jeux coopératifs ou semi coopératifs.

Les jeux d'ambiance.

les jeux d'adresse.

les jeux de réflexion : « Teris » comme exemple qu'on a developpe dans ce projet.

2. Les bienfaits des jeux de vidéos :

2.1. Vous deviendrez peut-être plus intelligent grâce à 'Starcraft'

Des chercheurs britanniques ont constaté que certains jeux vidéo, plus particulièrement les jeux de stratégie comme "**Starcraft**" peuvent augmenter la "flexibilité du cerveau", que les scientifiques décrivent comme "la pierre angulaire de l'intelligence humaine."

L'étude, menée par la Queen Mary University of London et par l'University College London, se fonde sur des tests psychologiques conduits avant et après que 72 volontaires aient joué à "Starcraft" ou au jeu de simulation de vie "Sims" durant 40 heures, étalées sur six à huit semaines. Selon l'étude, lors des tests psychologiques, les participants ayant joué à "**Starcraft**" complétaient des tâches de flexibilité cognitive plus rapidement et avec plus de justesse.

"Désormais, nous devons comprendre qu'est-ce qui provoque exactement ces changements, et si ces coups de fouet cognitifs sont permanents ou s'ils diminuent à mesure du temps," a déclaré Brian Glass, chercheur de cette étude. "Une fois que nous aurons compris, il sera possible de développer

des interventions cliniques pour, par exemple, des symptômes reliés au trouble du déficit de l'attention avec hyperactivité ou aux traumatismes crâniens."

2.2. Les jeux vidéo pourraient ralentir le vieillissement

Selon une étude menée cette année par l'Université de l'Iowa, jouer à des jeux de stimulation deux heures par semaine suffirait à ralentir le degré du déclin mental associé au vieillissement naturel.

Une étude conduite sur 681 individus âgés de 50 ans et plus, a révélé que jouer 10 heures à certains jeux vidéo pouvait retarder jusqu'à sept ans le déclin naturel de différentes compétences cognitives.

Durant cinq à huit semaines, un groupe de seniors a pratiqué des mots croisés informatisés tandis que trois autres groupes ont joué à un jeu informatique intitulé " **Road Tour** " Ce jeu consistait à retrouver des photos de véhicules tout en se rappelant où était placé un panneau de signalisation particulier, avec de plus en plus de pièges à mesure de l'avancée du joueur. L'expérience devait refléter la difficulté des conducteurs plus âgés à gérer une multitude d'informations à un croisement.

"Qu'il s'agisse d'un jeu spécifique ou d'un jeu comme 'World of Warcraft,' ils sont complexes en termes cognitifs et demandent de l'énergie mentale et des capacités pour pouvoir y jouer" explique Jason Allaire, professeur au département de psychologie de la North Carolina State University à Raleigh, qui n'a pas participé à l'étude. "Chaque fois que vous faites quelque chose demandant de l'énergie mentale, vous exercez vos capacités – c'est exactement comme lorsque vous exercez vos muscles, vous devenez plus fort."

2.3. Ils pourraient aider les enfants dyslexiques à mieux lire.

Une étude de l'université de Padoue a jeté un froid sur l'idée que les jeux vidéo sont mauvais pour le cerveau des jeunes enfants. En février dernier, des chercheurs italiens ont démontré que jouer à des jeux vidéo d'action peut améliorer les capacités de lecture d'enfants atteints de dyslexie.

L'équipe a séparé en deux groupes des enfants âgés de 7 à 13 ans, l'un devant jouer à un jeu d'action appelé "Rayman Raving Rabids" tandis que l'autre jouait à un jeu au rythme moins soutenu. Quand les capacités de lecture des enfants ont été testées par la suite, ceux qui avaient eu le jeu d'action pouvaient lire plus rapidement et avec plus d'exactitude. Les auteurs de l'étude ont émis l'hypothèse que les jeux d'action aidaient les enfants à agrandir la portée de leur attention, un atout crucial pour la lecture.

2.4. Les jeux vidéo peuvent soulager la douleur

En 2010, lors de la réunion annuelle de l'American Pain Society (Association américaine contre la douleur), des chercheurs ont démontré que les jeux vidéo, surtout ceux représentant une emphase de la réalité virtuelle, étaient efficaces pour réduire l'angoisse ou la douleur causées par les procédures médicales ou des maladies chroniques. L'étude a constaté que lorsque les gens sous chimiothérapie ou d'autres traitements lourds étaient immergés dans un monde virtuel de jeu vidéo, ils témoignaient ressentir bien moins de stress et de peur. En plus, ceux qui étaient traités pour des brûlures connaissaient une diminution de leur peine allant de 30 à 50 %.

En se référant à la technologie de capture de mouvement de la Xbox Kinect ou de la Wii, Charles Friedman, rattaché aux Centres de soulagement de la douleur, explique que les jeux vidéo permettent au cerveau de rester actif en utilisant d'autres sens au lieu de rester focalisé sur la douleur. Jouer aux jeux vidéo permet aussi de relâcher des endorphines dans le cerveau, un composant chimique généralement associé au bonheur et capable d'endormir la sensation d'inconfort.

2.5. Les jeux et le monde militaire.

Le lien entre les jeux vidéo et le monde **militaire** n'est pas nouveau, et explique en partie pourquoi ils semblent être les produits d'une « culture de violence ».

Initialement publié par la revue Silence.

Ce divertissement est né au sein du complexe militaro-industriel américain en pleine guerre froide, dans le contexte de la course à l'espace avec l'URSS.

Une époque où le Pentagone dope de crédits le milieu de la recherche en informatique – dont font partie les hackers, inventeurs des jeux vidéo.

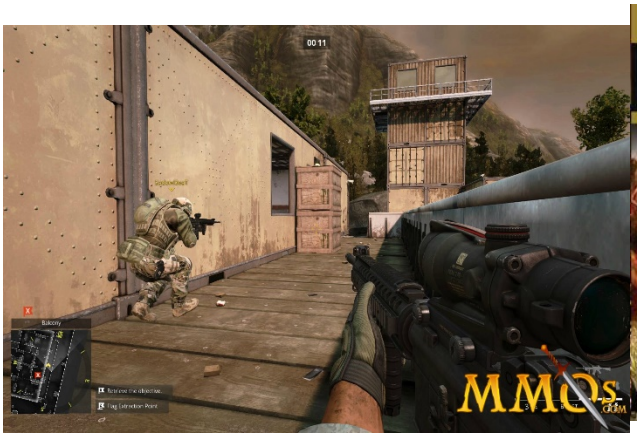
Apparu en 1962, **Spacewar**, le premier jeu vidéo simulant un combat spatial a servi avant tout à mesurer la puissance de calcul des ordinateurs. Dans les années 80, **Battlezone,** une simulation de tank vectorielle très célèbre a été utilisée par l'armée américaine comme support à l'entraînement.

➔ **Outil de recrutement :**

Mais c'est au cours des années 90 que cette collaboration, profitant d'une massification du marché, va déboucher sur une véritable synergie entre le Pentagone et l'industrie des jeux vidéo, le premier injectant des dépenses de recherche et investissement dans la conception de simulateurs militaires qui ont fini par être adaptés pour le commerce.

Des jeux comme **Harpoon**, Rainbow Six et Full Spectrum Warrior sont les fruits de cette collaboration.

Le recours au jeu vidéo comme outil de recrutement ne s'est amorcé qu'en 2002 avec **America's Army**, un jeu multijoueur en ligne qui a réussi à attirer jusqu'à présent près de 10 millions de joueurs.



En 2006, l'armée française s'est décidée à suivre le modèle américain à l'aide d'une version transformée de **Ghost Recon Advanced Warfighter** (GRAW) baptisée Instinct, jeu commercial dont la rhétorique est proche de la guerre contre le terrorisme.

Bande-annonce de « **Ghost Recon Advanced Warfighter** »

Une synergie s'établit alors entre l'armée et l'éditeur du jeu Ubisoft puisque la première a fourni au second une expertise militaire afin d'améliorer le réalisme de la simulation dans l'optique d'une future version commercialisée.

➔ **S'entraîner à la guerre :**

Les enjeux qui se profilent sont doubles. D'une part, les soldats peuvent s'entraîner à la guerre et se familiariser avec les armes et combinaisons futuristes dont les jeux militaires s'inspirent (les exosquelettes, muscles artificiels, senseurs etc. des soldats de GRAW) et qui ne sont pas encore introduits sur le terrain.

En France par exemple, pour répondre aux mutations de la guerre, l'armée française recourt au fantassin à équipement et liaisons intégrées (FELIN) pourvu des dernières nanotechnologies de pointe, un combattant proche du cyborg (mi- humain, mi- machine) anticipé par GRAW.

De l'autre, le public (jeune, masculin, issu de classe populaire) que l'armée cherche à recruter participe clairement à une « sensibilisation » militaire à partir du monde civil et se sensibilise à la rhétorique de la guerre propre et antiterroriste.

➔ **Développement des armées :**

Cependant, la collaboration entre l'armée française et l'industrie des jeux ne s'est pas vraiment poursuivie par la suite. Elle se limite aujourd'hui tout au plus à l'insertion de publicités de recrutement dans les jeux de sport.

D'évidence, les jeux de guerre ne pourraient voir le jour sans la conjonction de plusieurs forces. Comme la préparation des guerres réelles (par exemple, la recherche dans les nanotechnologies est

mise au service du monde militaire), la conception des simulateurs de guerre puise dans la technologie civile et les ressources des plus gros distributeurs de contenu culturel.

A un niveau marginal, nous participons donc au développement des armées « en jouant », comme c'est le cas, depuis longtemps, en « regardant » les nombreux films de guerre réalisés avec l'aide du Pentagone.

➔ Les jeux de Simulation

Il ne fait aucun doute que les jeux et simulations éducatifs peuvent être un moyen d'enseignement pertinent (Radoff 2011 cité par Cheng et al. 2014). De par leur aspect intrinsèquement motivant et engageant, ils apparaissent, pour plusieurs, comme un cadre idéal à l'apprentissage (Clark et al. 2011 et Lim 2008 cités par Cheng et al. 2014). Puisqu'un des rôles essentiels de l'école est la préparation des élèves au marché du travail, de plus en plus de projecteurs se braquent sur l'enseignement des sciences. En effet, de récentes recherches montrent que plus de 70 % des

meilleurs emplois au Canada nécessitent une formation en science et technologie (École branchée 2013).

3. Les outils de création de jeux vidéo

Depuis une trentaine d'années, les jeux vidéo se sont considérablement démocratisés au point de faire partie des loisirs préférés de très nombreuses personnes à travers le monde. On peut poser les questions suivantes : Quels sont-ils ? Est-il si facile de développer un jeu ? Peut-on trouver des outils de création pour débutants ?

➔ Quels sont les outils de création qu'utilise un programmeur de jeux vidéo ?

Tout d'abord, ce qu'il faut savoir c'est que pour programmer un jeu que ce soit sur PC ou sur consoles, le programmeur travaille principalement sur PC. Lorsqu'il ne travaille pas sur PC, il exécute son code sur un kit de développement qui est en quelque sorte une grosse console offrant de larges capacités de débogage. En général, le compilateur le plus utilisé est Visual C++ mais pour certaines plateformes Code Warrior peut également être utilisé. Il faut donc avoir de très bonnes connaissances en langage informatique pour devenir programmeur de jeux vidéo.

En plus de cela, des connaissances graphiques sont également nécessaires puisque certains programmeurs travaillent en collaboration avec des graphistes pour la 3D, les menus, les effets spéciaux… Il est donc très utile de savoir se servir des outils de graphistes afin de pouvoir être autonome dans son travail.

En règle générale, un jeu est programmé en C++ ou en JAVA et comme le nombre de ligne est particulièrement conséquent, chaque programmeur possède une spécialité (3D, IA, GamePlay, Création des outils, Son, Bas Niveau, Spécialiste sur une plateforme, Interfaces et menus, Effets Spéciaux, Recherche et Développement…).

Les outils de création pour les débutants

→ Les moteurs de création des jeux :

Contre toutes attentes, oui il est tout à fait possible de créer un jeu vidéo de chez soi sans forcément avoir des connaissances en programmation. Il vous suffit d'avoir une bonne idée de jeu, un scénario établi, un concept de gameplay ainsi que quelques logiciels dont certains sont même gratuits.

Scratch

Scratch est un logiciel permettant de créer des films d'animation et d'acquérir des bases en programmation. Ce n'est pas un logiciel spécialisé dans les jeux vidéo mais vous pourrez tout de même en créer quelques-uns avec puis, par la suite, vous pourrez publier vos créations sur le site de Scratch prévu à cet effet.

RPG Maker

RPG Maker est un logiciel payant permettant de créer des jeux d'aventure du même style que les tous premiers Zelda ou Final Fantasy. En d'autres termes, vous pourrez créer des RPG (ou jeux de rôle) dans leurs moindres détails (cartes, objets, scénario, énigmes, donjons…). Le logiciel est simple à prendre en main tout comme la partie programmation. C'est un logiciel idéal pour les débutants mais aussi pour les personnes un peu plus expérimentées.

Alice

Alice est un logiciel qui permet de programmer et créer des univers en 3D, des films d'animation et des jeux interactifs. L'interface est un peu difficile à prendre en main mais il existe de nombreux tutoriels. En revanche, Alice est un logiciel entièrement en anglais.

GameSalad

GameSalad est un logiciel très complet permettant de créer des jeux gratuitement. Il est également possible de créer des jeux pour iPhone et iPad mais il faudra payer des droits de développeur de 99 euros par an. C'est un logiciel très facile à utiliser mais qui possède un inconvénient : il ne fonctionne que sur MAC.

4. Les langages utilisent dans le développement des jeux

À une époque où Internet prend une place de plus en plus importante dans nos vies, il peut être très utile, voire indispensable dans certains cas, de savoir comment tout cela fonctionne. Ne soyez plus spectateur mais devenez acteur, sachez que vous êtes tout à fait capable de créer vos propres applications ou sites Internet. DGS vous présente 10 des langages de programmation les plus utilisés au monde des jeux.

4.1. JAVA

C'est quoi ? Le **Java** est un langage de programmation orienté objet qui a été développé par **Sun Microsystems** dans les années 90. C'est l'un des langages de programmation les plus populaires, devenu un standard chez les programmes d'entreprise, dans les contenus et jeux Web ou encore les applications mobiles. C'est également le langage de programmation utilisé pour le système d'exploitation mobile **Android**. Le **Java** est fait pour travailler sur de nombreuses plateformes : un programme codé sur **Mac OS X** pourrait par exemple fonctionner sur **Windows**.

4.2. Le langage C

C'est quoi ? Le langage C'est le langage de programmation le plus vieux et le plus utilisé au monde, créé dans les années 70. Il pose les bases de nombreux autres langages, comme le **C#**, le **Java**… il est donc conseillé de s'y intéresser de près. Le **C** est principalement utilisé pour intégrer des systèmes d'opérations et des applications incrustées dans des pages Web.

4.3. Le langage C++

Le **C++** est un langage de niveau intermédiaire qui possède des fonctions orientées objet et qui était à l'origine fait pour améliorer le langage **C**. Le **C++** est le moteur de nombreux programmes très connus

comme **Firefox**, **Winamp** ou encore ceux de la suite **Adobe**. Il est utilisé pour développer des applications, des systèmes d'exploitation, des serveurs et clients internet très performants ou encore des jeux vidéo.

4.4. Le langage C#

Le **C#**, prononcer « C sharp », est un langage de programmation à paradigmes multiples développé par **Microsoft** via la **.NET initiative**. Il combine des principes du **C** et du **C++** et est un langage généraliste utilisé pour développer des logiciels pour Microsoft et les plateformes **Windows**.

4.5. PHP

Le **PHP** (pour Hypertext Processor) est un langage de programmation libre utilisé pour le développement de sites Internet et d'applications dynamiques. Il peut être directement incrusté dans des documents **HTML** plutôt que par un fichier externe, ce qui l'a rendu très populaire chez les développeurs Web. Le **PHP** fait fonctionner plusieurs millions de sites Internet (environ 200 millions), notamment **WordPress** et **Facebook**.

4.6. PYTHON

Le **Python** est un langage de programmation de haut niveau utilisé pour les sites Internet et les applications mobiles. Il est considéré comme un langage plutôt facile à maîtriser pour les débutants grâce à sa syntaxe compacte et parce qu'il est très lisible, ce qui permet aux développeurs d'utiliser moins de lignes de code pour exprimer une fonction qui en aurait nécessité plus dans d'autres langages. Il fait fonctionner les applications Web d'**Instagram**, **Pinterest** et est utilisé par **Google**, **Yahoo!** et même la **NASA** !

5. Graphisme

Les graphismes de jeu vidéo représentent l'ensemble des formes sous lesquelles les jeux vidéo sont représentés à l'écran, en utilisant des techniques de plus en plus évoluées depuis le début de l'histoire du jeu vidéo. De nombreuses techniques ont été utilisées pour afficher le contenu de jeu vidéo, notamment sous forme textuelle, ou graphique de type vectoriel, en deux dimensions, isométrique, ou

en trois dimensions. L'évolution des graphismes de jeu vidéo est directement liée à la progression des capacités matérielles informatiques et celles des logiciels qui permettent aussi bien d'afficher que de fabriquer ces graphismes.

Il y a plusieurs types de graphismes utilises dans le monde de jeux vidéo, on cite quelques-unes :

5.1. 3D

3D

Très utilisée dans les jeux vidéo et les films. En communication normale elle doit être utilisée avec parcimonie car elle se démode extrêmement vite.

5.2. Le pixel art

Pixel art

Le pixel art est un graphisme provenant tout droit des jeux vidéo rétro.
Aujourd'hui il est à la mode pour les nostalgiques.

5.3. Le rétro vintage

Retro vintage

Ce style est vraiment de plus en plus utilisé.

Il se rapproche tout de même bien de tout ce qui peut être années 50 américaines. En sommes ce sont des graphismes avec des aspects vieillis, des graphismes souvent monochromes et des typographies à tout va.

Pour en conclure, chaque style présenté est fort. Ils sont tous très connotés et ne peuvent pas être utilisés pour tout et n'importe quoi.

6. Pourquoi besoin de développer un jeu ?

→ Tetris : plus qu'un jeu

Tetris a fait l'objet de nombreuses études dans le monde médical, et plus particulièrement en neurosciences. C'est ainsi que des recherches ont été conduites sur les effets du jeu Tetris sur le cerveau humain, par « The Mind Research Network », une fondation américaine spécialisée en neurosciences. L'étude a été prise en charge par un neurochirurgien américain, Richard Haier et conduite par l'institut neurologique de Montréal sur 26 jeunes filles, des adolescentes. Le choix s'est orienté vers les filles, car, à cet âge, le développement du cerveau est plus rapide que celui des garçons.

Ainsi, ces 26 jeunes filles devaient jouer au Tetris sur une période de 3 mois. Leur cerveau a été comparé à celui d'un groupe témoin de jeunes filles n'ayant jamais touché ce jeu. Au bout de 12 semaines, le docteur Richard Haier a procédé à une comparaison des cerveaux des jeunes filles ayant joué à Tetris et de celles n'ayant pas joué. Les résultats sont plus que surprenants : les joueuses ont développé un cortex plus épais que les autres : certaines zones du cerveau sont devenues très efficaces, et d'un autre côté, leur matière grise s'est multipliée. Pour faire plus simple, jouer à Tetris a permis à ces filles d'augmenter leur rendement cérébral.

Dans la vie pratique, les joueurs de Tetris peuvent par exemple lire une carte routière à l'envers, se garer facilement, repérer des lieux complexes plus facilement. À l'école, les enfants peuvent donc facilement assimiler les cours de géométrie ou de géographie et c'est notre objectif a laquelle on a développé ce jeu vidéo.

Chapitre 2: Phases d'analyse et de conception

1. Introduction

L'objectif de ce chapitre est de donner une vue sur notre Application, les différentes phases conçues et les différentes méthodes utilisées à travers le diagramme de class, et aussi de représenter visuellement l'état d'avancement des différentes activités.

1.1. Modèle de cycle de vie d'un logiciel

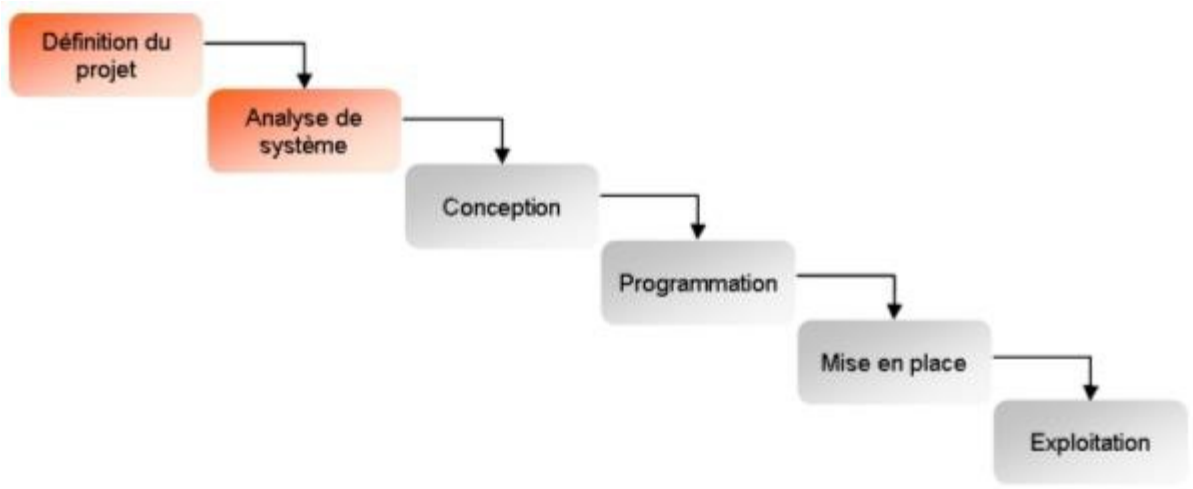


Figure : Modèle en cascade

1.2. Méthodologie de conception

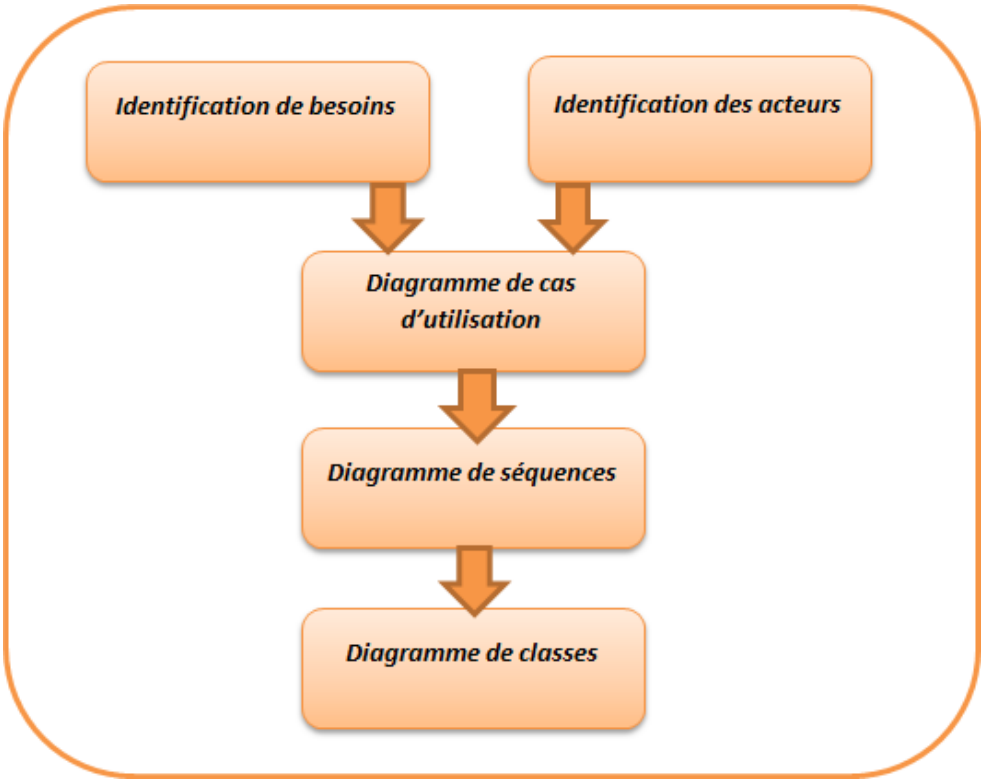


Figure : Méthodologie de conception adoptée

(Tâches) qui constituent notre projet en utilisant Le diagramme de Gantt.

2. Diagramme de GANTT

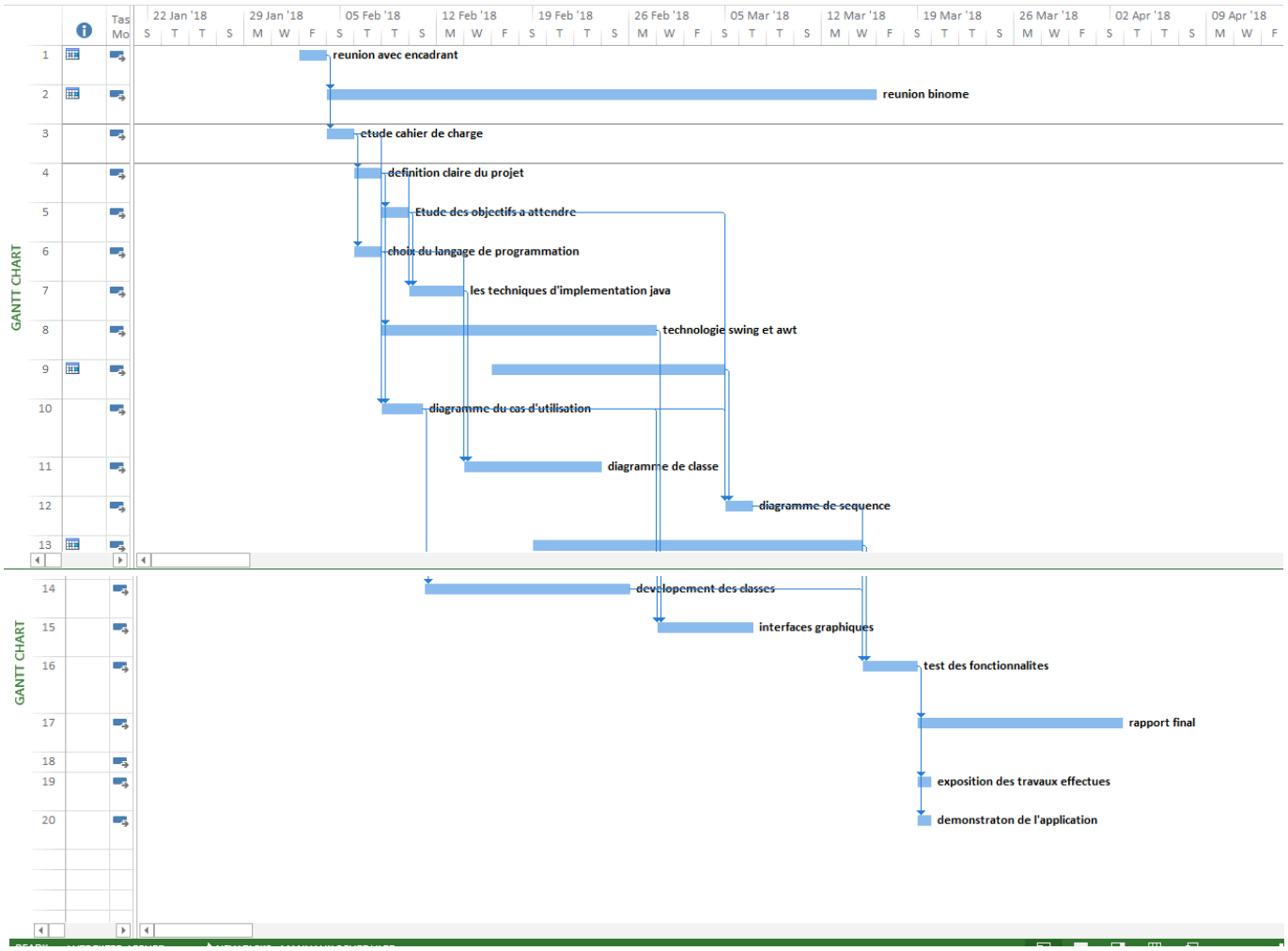
Un outil utilisé en ordonnancement et en gestion de projet et permettant de visualiser dans le temps les diverses tâches composant un projet. Il s'agit d'une représentation d'un graphe connexe et orienté, qui permet de représenter

graphiquement l'avancement du Project.

GANTT CHART

	Task Mode	Nom de la tâche	Duration	Start	Finish	Predecessors	Resource Names	Add New Column
1		Reunions avec encadrant	2 days	Fri 02/02/18	Sat 03/02/18		reunion avec encadrant	
2		reunions entre les binomes	70 days	Sun 04/02/18	Sat 14/04/18	1	reunion binome	
3		etudier le cahier de charge	2 days	Sun 04/02/18	Mon 05/02/18	1	etude cahier de charge	
4		definition claire du projet	2 days	Tue 06/02/18	Wed 07/02/18	3	definition claire du projet	
5		Etude des objectifs a attendre	2 days	Thu 08/02/18	Fri 09/02/18	4	Etude des objectifs a	
6		choix du langage de programmation	2 days	Tue 06/02/18	Wed 07/02/18	3	choix du langage de programmation	
7		connaitre les techniques d'implementation java	4 days	Sat 10/02/18	Tue 13/02/18	4;5	les techniques d'implementation	
8		Etude du technologie swing et awt	20 days	Thu 08/02/18	Tue 27/02/18	6	technologie swing et awt	
9		Conception Architecturale	17 days?	Fri 16/02/18	Sun 04/03/18			
10		Conception du diagramme du cas d'utilisation	3 days?	Thu 08/02/18	Sat 10/02/18	3;4	diagramme du cas d'utilisation	
11		conception du diagramme de classe	10 days?	Wed 14/02/18	Fri 23/02/18	6;7	diagramme de classe	
12		Conception du diagramme de sequence	2 days	Mon 05/03/18	Tue 06/03/18	5;9;10	diagramme de sequence	
13		Implementation	24 days	Mon 19/02/18	Wed 14/03/18			
14		developement des classes	15 days	Sun 11/02/18	Sun 25/02/18	10	developement des classes	
15		l'ajout des interfaces graphiques	7 days	Wed 28/02/18	Tue 06/03/18	8;10	interfaces graphiques	
16		Test de l'ensemble des fonctionnalites de l'application	4 days	Thu 15/03/18	Sun 18/03/18	12;13;14	test des fonctionnalites	
17		redaction du rapport final	15 days	Mon 19/03/18	Mon 02/04/18	16	rapport final	
18		soutenance	1 day	Thu 26/04/18	Thu 26/04/18			
19		Exposition des travaux effectues	1 day	Mon 19/03/18	Mon 19/03/18	16	exposition des travaux effectues	
20		Demonstration de l'application	1 day	Mon 19/03/18	Mon 19/03/:	16	demonstraton de l'application	

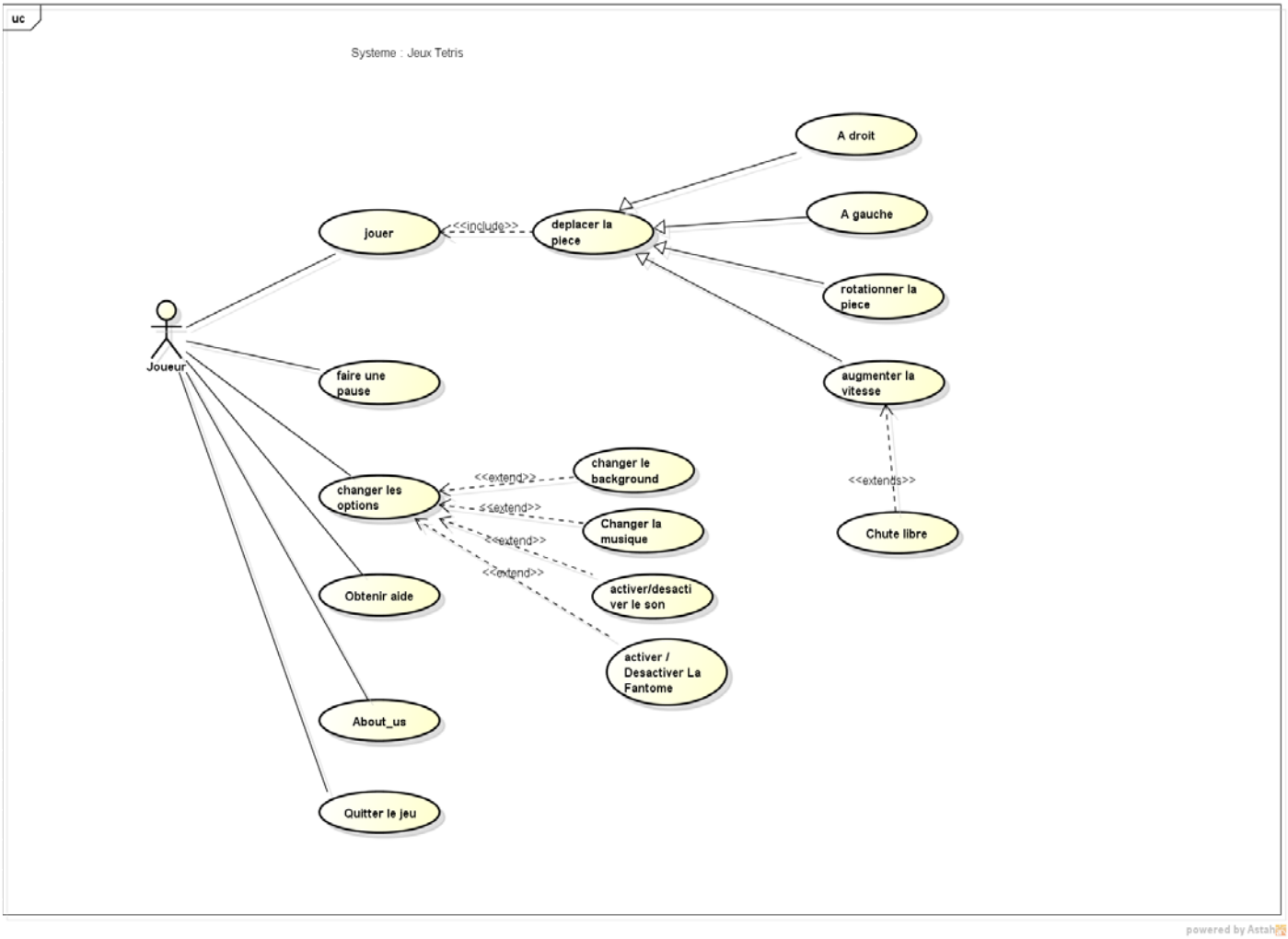
GANTT CHART



3. Diagramme de cas d'utilisation

Les exigences fonctionnelles globales sont représentées par le modèle de cas d'utilisation et la description textuelle. Le modèle de cas d'utilisation se concentre sur les facteurs critiques de succès du système, en termes de fonctionnalités ou caractéristiques que les utilisateurs ont besoin d'interagir avec.

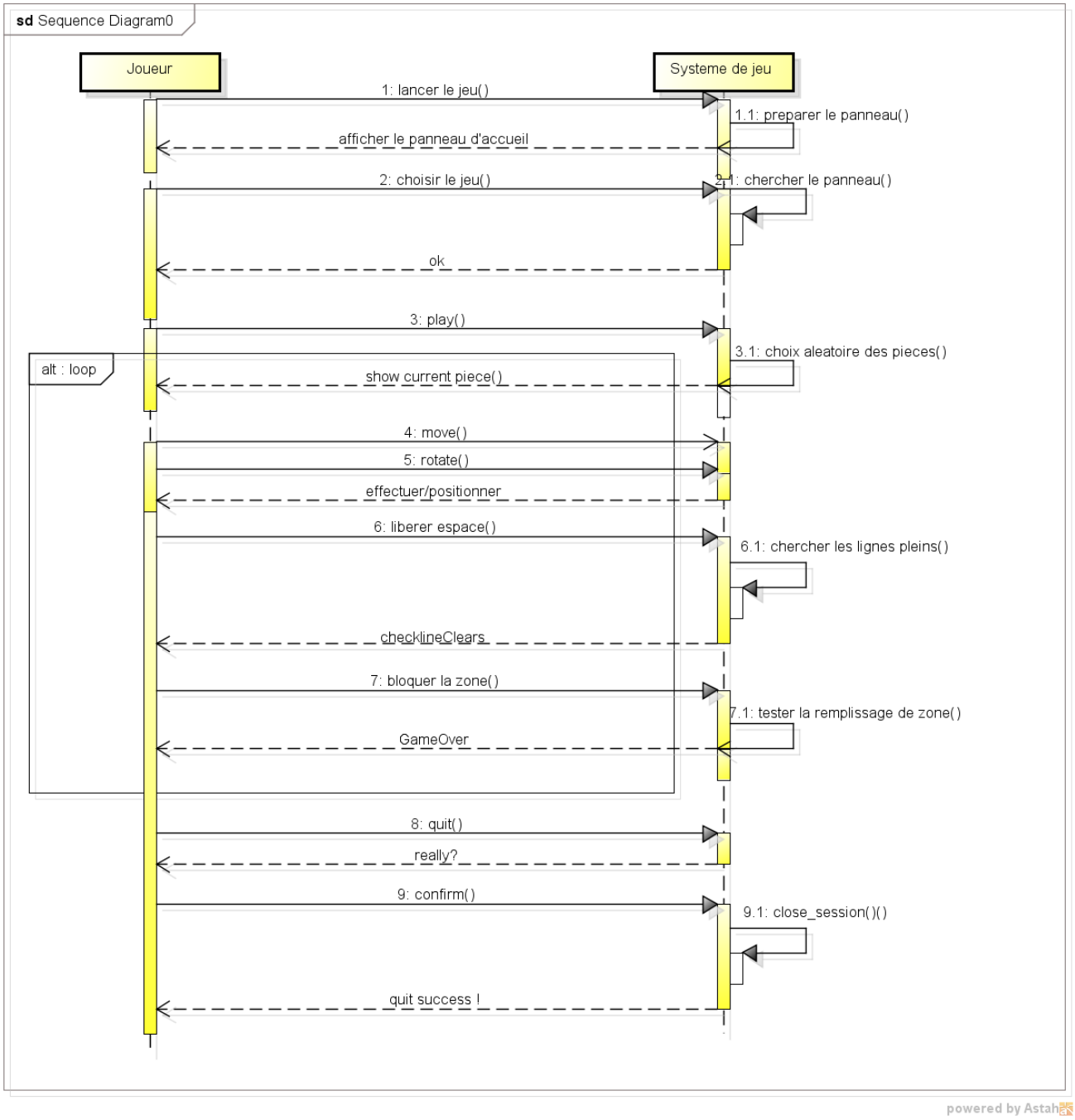
Spécification des auteurs : Le joueur.



4. Diagrammes de Séquence

Un diagramme de séquences se charge de représenter séquentiellement, en fonction du temps, le chaînage, déroulement des traitements et des interactions et l'échange d'informations entre les différents objets impliqués dans une fonctionnalité donnée du système.

On présentera ci-après les diagrammes de séquences que nous avons jugés importants de notre application.



1. **dans ce diagramme on a décrit de manière dynamique et interactive le cas d'utilisation :jouer**
2. **--le joueur lance l'application, le système affiche le menu ,le joueur choisit d'abord le niveau qu'il veut, et il commence la partie.**
3. **le système affiche la pièce qui chute et les autres suivants , le joueur la contrôle avec des touches, le système chercher les lignes remplis et les supprime pour libérer d'espace au utilisateur, et a chaque fois il teste si la zone est bloquante pour afficher le GameOver.**

4. **cette opération se répète plusieurs fois jusqu'a ce que le joueur perd la partie.**
5. **après la terminaison de la partie le système donne au joueur la possibilité de reprendre une nouvelle partie ou bien quitter.**

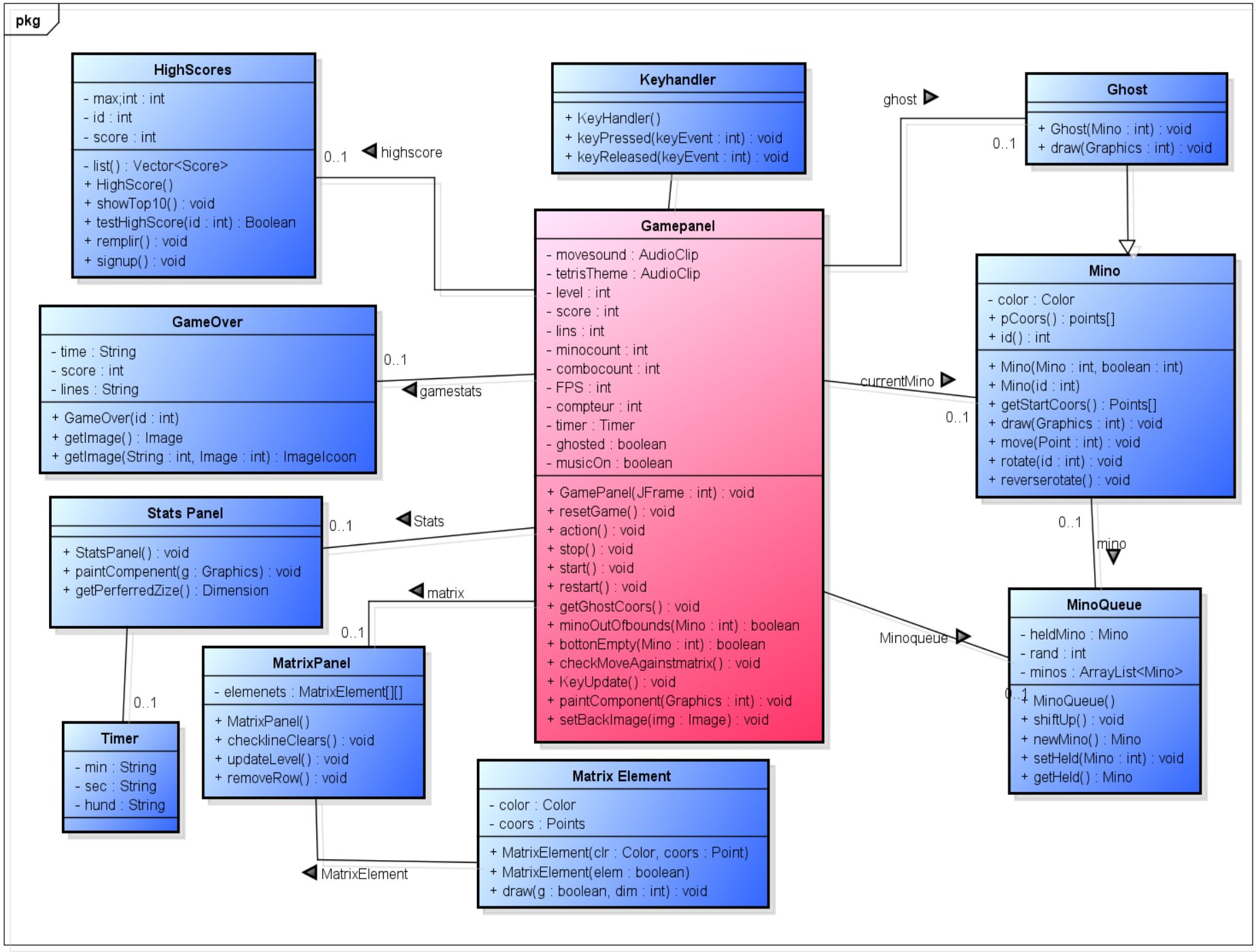
5. Diagramme de classes

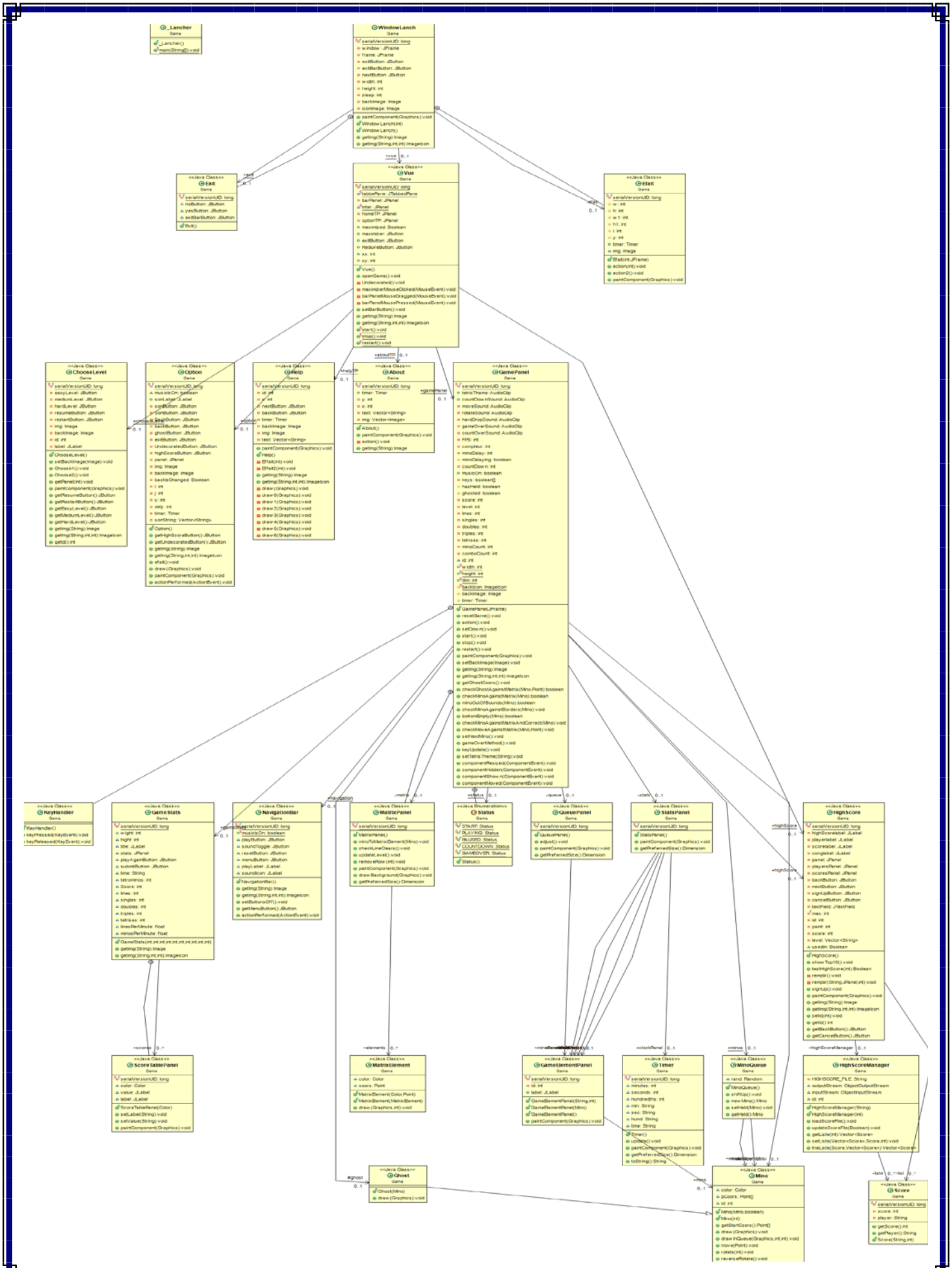
Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.

Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Chaque langage de Programmation orienté objet donne un moyen spécifique d'implémenter le paradigme objet (pointeurs ou pas, héritage multiple ou pas, etc.), mais le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier. Les principaux éléments de cette vue statique sont les classes et leurs relations : association, généralisation et plusieurs types de dépendances, telles que la réalisation et l'utilisation.

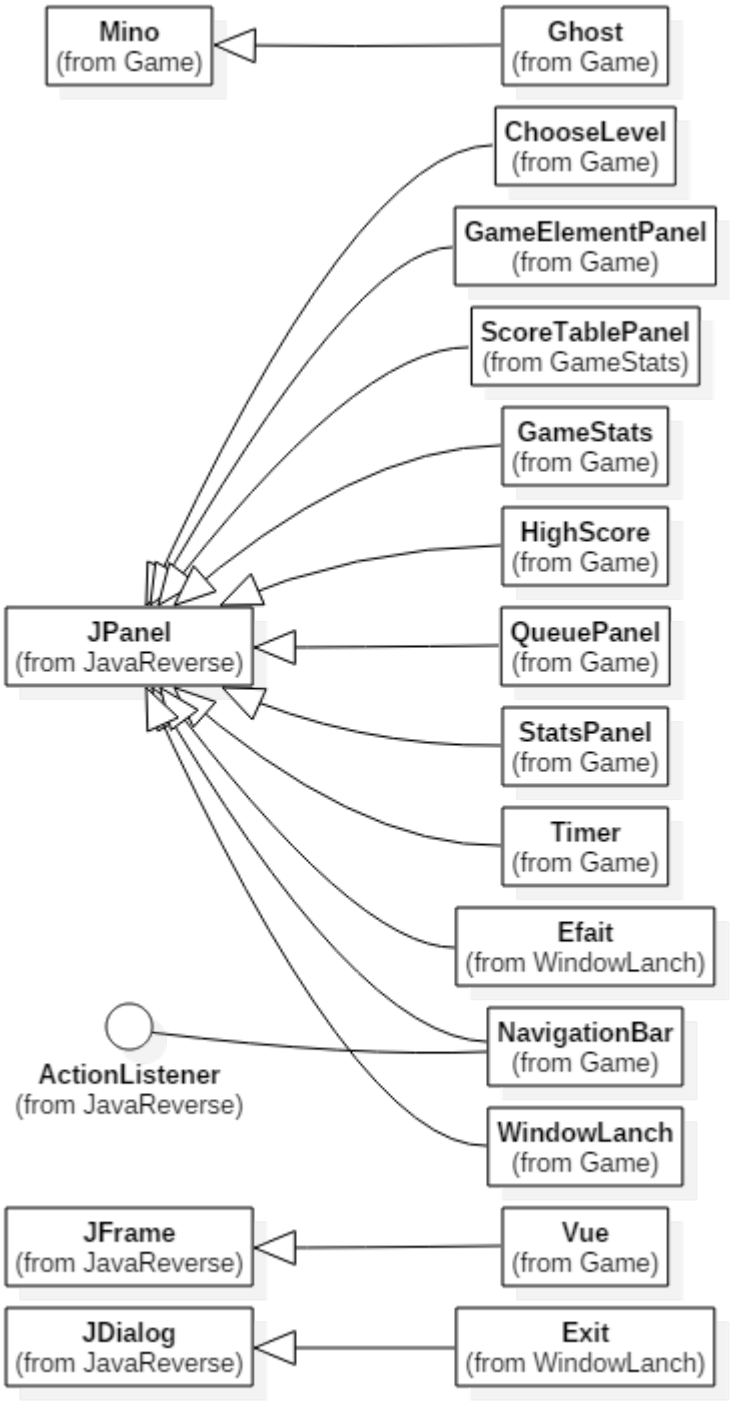
- ✓ **Classe** : description formelle d'un ensemble d'objets ayant une sémantique, des propriétés et un comportement commun.
- ✓ **Association** : relation sémantique entre deux ou plusieurs classes.
- ✓ **Propriété** (attribut) : élément permettant de décrire une classe ou une association.
- ✓ **Méthode** : opération réalisée par les objets d'une classe.

Le diagramme ci-dessous nous donne une idée sur l'implémentation de différentes classes antes notre application.

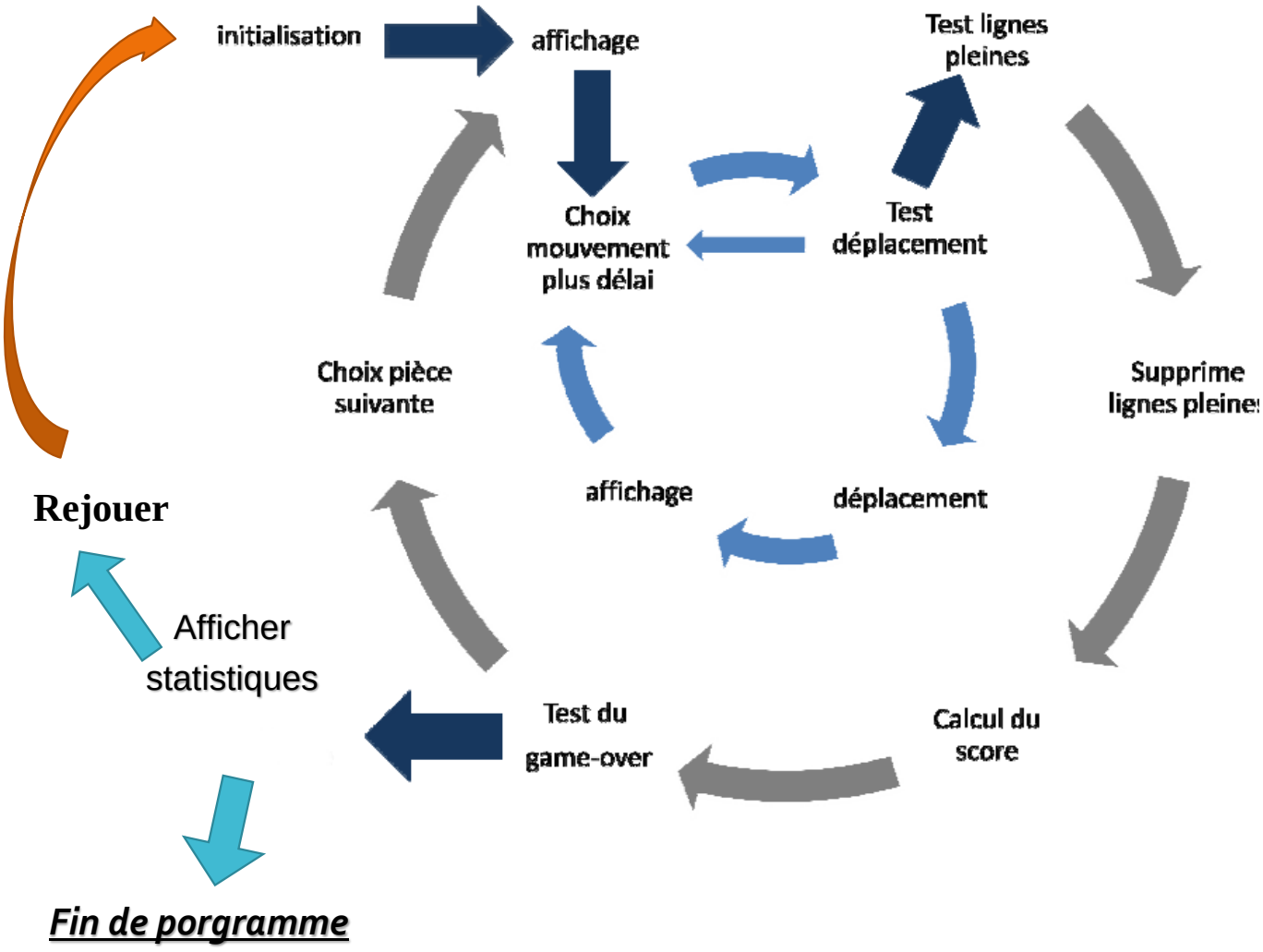

pkg
HighScores
- max;int : int
- id : int
- score : int
- list() : Vector<Score>
+ HighScore()
+ showTop10() : void
+ testHighScore(id : int) : Boolean
+ remplir() : void
+ signup() : void
0..1
highscore
Keyhandler
+ KeyHandler()
+ keyPressed(keyEvent : int) : void
+ keyReleased(keyEvent : int) : void
ghost
Ghost
+ Ghost(Mino : int) : void
+ draw(Graphics : int) : void
0..1
Gamepanel
- movesound : AudioClip
- tetrisTheme : AudioClip
- level : int
- score : int
- lins : int
- minocount : int
- combocount : int
- FPS : int
- compteur : int
- timer : Timer
- ghosted : boolean
- musicOn : boolean
+ GamePanel(JFrame : int) : void
+ resetGame() : void
+ action() : void
+ stop() : void
+ start() : void
+ restart() : void
+ getGhostCoors() : void
+ minoOutOfbounds(Mino : int) : boolean
+ bottonEmpty(Mino : int) : boolean
+ checkMoveAgainstmatrix() : void
+ KeyUpdate() : void
+ paintComponent(Graphics : int) : void
+ setBackImage(img : Image) : void
Mino
- color : Color
+ pCoors() : points[]
+ id() : int
+ Mino(Mino : int, boolean : int)
+ Mino(id : int)
+ getStartCoors() : Points[]
+ draw(Graphics : int) : void
+ move(Point : int) : void
+ rotate(id : int) : void
+ reverserotate() : void
currentMino
0..1
GameOver
- time : String
- score : int
- lines : String
+ GameOver(id : int)
+ getImage() : Image
+ getImage(String : int, Image : int) : ImageIcoon
0..1
gamestats
Stats Panel
+ StatsPanel() : void
+ paintCompenent(g : Graphics) : void
+ getPerferredZize() : Dimension
0..1
Stats
matrix
0..1
0..1
mino
MinoQueue
- heldMino : Mino
- rand : int
- minos : ArrayList<Mino>
+ MinoQueue()
+ shiftUp() : void
+ newMino() : Mino
+ setHeld(Mino : int) : void
+ getHeld() : Mino
Minoqueue
0..1
Timer
- min : String
- sec : String
- hund : String
0..1
MatrixPanel
- elemenets : MatrixElement[][]
+ MatrixPanel()
+ checklineClears() : void
+ updateLevel() : void
+ removeRow() : void
Matrix Element
- color : Color
- coors : Points
+ MatrixElement(clr : Color, coors : Point)
+ MatrixElement(elem : boolean)
+ draw(g : boolean, dim : int) : void
MatrixElement


_Lancher
WindowLanch
Exit
Vue
Etat
ChooseLevel
Option
Help
About
GamePanel
KeyHandler
GameStats
NavigationBar
MatrixPanel
Status
QueuePanel
StatsPanel
HighScore
ScoreTablePanel
MatrixElement
GameElementPanel
Timer
MinoQueue
HighScoreManager
Ghost
Mino
Score

6. Diagramme Hiérarchique



7. Diagrammes d'activités



Chapitre 3 : Réalisation

Après avoir élaboré l'étude et la conception de notre application, nous abordons dans ce chapitre le dernier volet de ce rapport, qui a pour objectif d'exposer la phase de réalisation. Cette dernière présente le résultat du travail effectué durant ce projet.

Nous présentons en premier lieu notre choix de l'environnement de travail, où nous spécifions l'environnement matériel et logiciel qu'on a utilisé pour réaliser notre application, ainsi que le choix du langage de développement.

Nous clôturons ce chapitre par une présentation des captures d'écran de quelques interfaces réalisées pour illustrer le fonctionnement des activités du système.

1. Environnement de développement

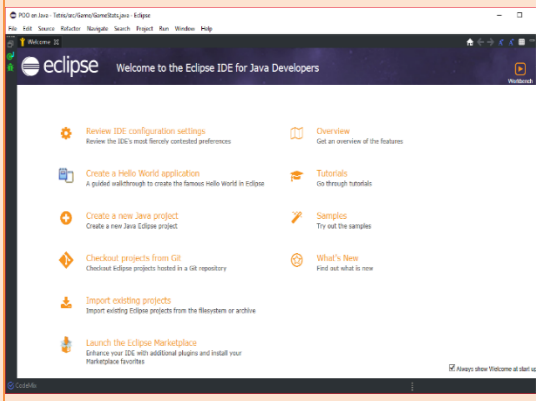
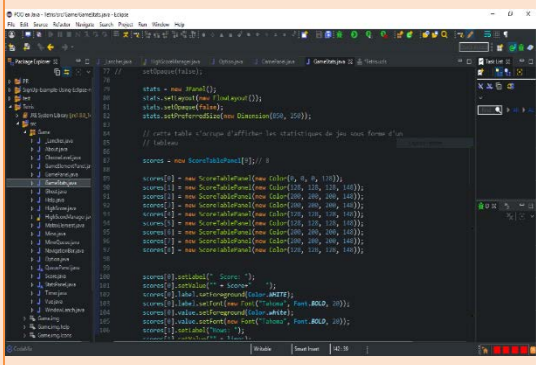
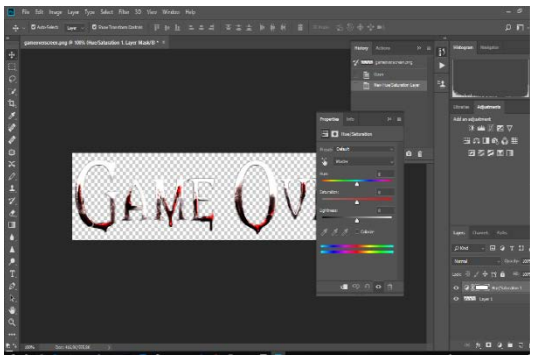
1.1. Environnement Matériel

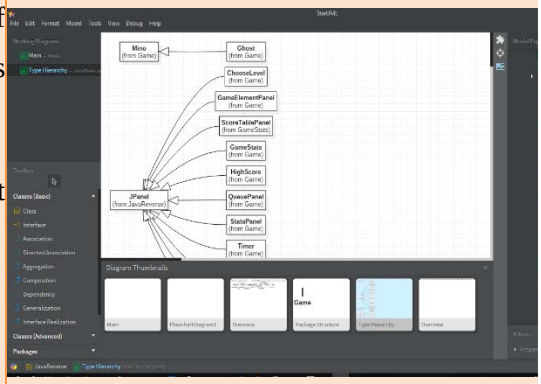
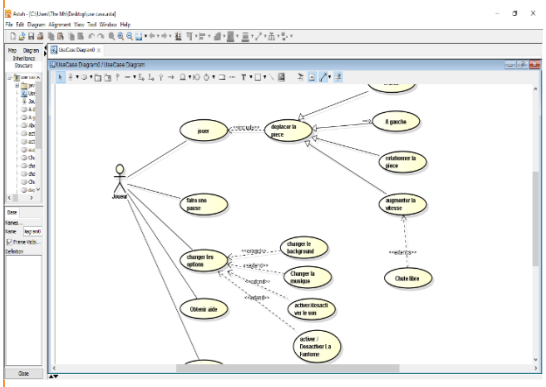
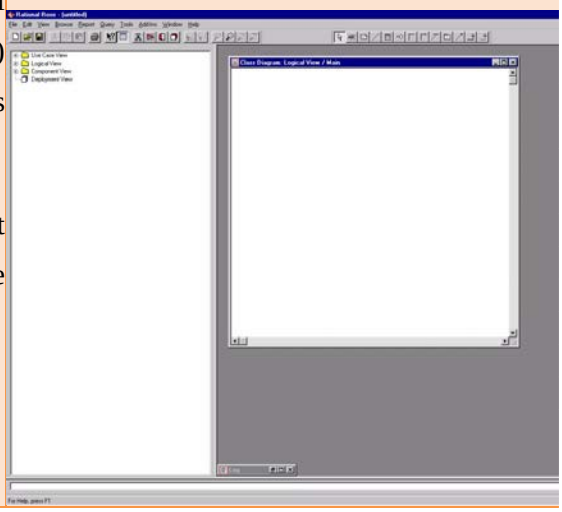
Le développement de l'application est réalisé via deux ordinateurs portables ayant les caractéristiques suivantes :

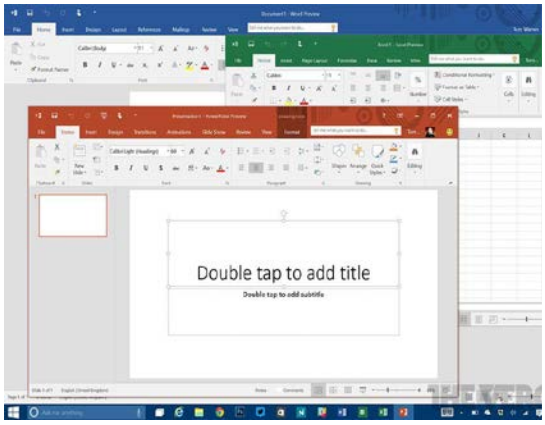
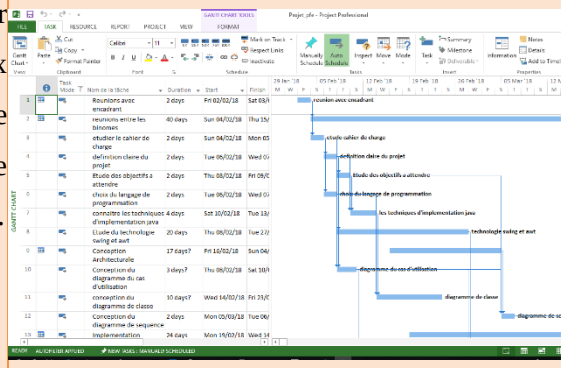
Caractéristique	HP 250	Pavilion g6
Marque	DELL	HP
Processeur	Intel(R) Core (TM) i7-4600U CPU @ 2,10GHz 2.70GHz 64bits	Intel(R) Core (TM) i3-CPU @ 2,60GHz 64bits
RAM	8,00 Go	4,00 Go
Disque dur	256 Go	500 Go
Système d'exploitation	Microsoft Windows 10 Professionnel N Version 1803	Microsoft Windows 10 Professionnel N Version 1709

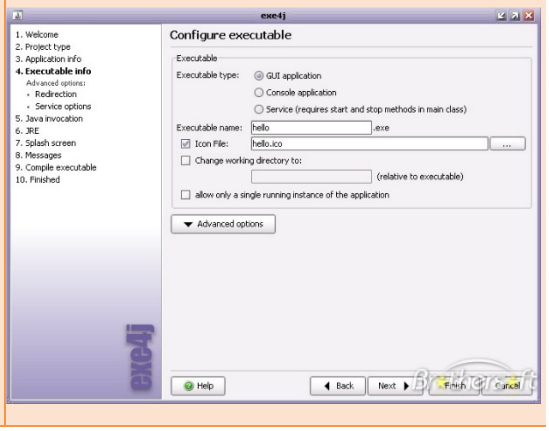
Tableau6: Materiel de base

1.2. Environnement Logiciel

Outils	Caractéristique	Aperçu de l'écran principal
Eclipse IDE	**Eclipse** est un IDE, Integrated Development Environment (EDI environnement de développement intégré en français), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation. Au fur et à mesure que nous programmons, Eclipse compile automatiquement le code que nous écrivons, en soulignant en rouge ou jaune les problèmes qu'il décèle. Il souligne en rouge les parties du programme qui ne compilent pas, et en jaune les parties qui compilent mais peuvent éventuellement poser problème Il est déconseillé de continuer d'écrire le programme quand il contient des erreurs, car Eclipse est dans ce cas moins performant pour vous aider à écrire le programme. aussi de modélisation, de conception, De test, etc.	 
Adobe Photoshop CC 2018	Avec **Photoshop CC**, la meilleure application d'imagerie et de design au monde, vous pouvez donner vie à toutes vos idées. Créez et embellissez des photos et illustrations 2D/3D. Créez des sites web et des applications mobiles. Montez des vidéos, simulez des tableaux réalistes, etc. Cette application vous permet de concrétiser toutes vos idées.	 

StartUML	C’est un logiciel de modélisation UML open source sous une licence modifiée de GNU GPL. L'objectif de ce projet est de se substituer à des solutions commerciales comme IBM Rational Rose ou Borland Together. StarUML gère la plupart des diagrammes spécifiés dans la norme UML 2.0.	
astah community	Pour la modélisation UML.	
Rational Rose	C’est un logiciel édité par l'entreprise Rational Machines (plus tard renommée Rational Software) utilisé pour créer et éditer les différents diagrammes du modèle UML d'un logiciel. Rational Rose permet également de sauvegarder et d'imprimer ces diagrammes, ainsi que de Générer le code source Java ou C++ qui leur correspondent.	

	Microsoft Office 2016 est une suite d'applications de productivité comprenant Microsoft Word, Microsoft Excel, Microsoft PowerPoint et Microsoft Outlook. C'est le successeur de Microsoft Office 2013 pour Windows et d'Office pour Mac 2011.	
Ms project	Est un logiciel de gestion de Project édité par Microsoft. Il permet aux chefs de projet et aux planificateurs de planifier et piloter les projets, de gérer les ressources et le budget, ainsi que d'analyser et communiquer les données des projets.	
Adobe Acrobat DC	Le programme Adobe Acrobat Reader DC (auparavant appelé Adobe Acrobat Reader, puis Adobe Reader) est disponible gratuitement sur le site Web d'Adobe et permet l'affichage et l'impression des fichiers PDF. Les nombreux programmes commerciaux de la gamme Acrobat permettent d'effectuer quelques modifications minimales et d'ajouter des fonctionnalités aux documents PDF (format PDF par exemple). En outre, ils sont fournis avec d'autres modules comprenant notamment un pilote d'imprimante pour créer des fichiers PDF à partir d'applications Microsoft Excel.	

Exe4j 	Lors du développement d'un logiciel, il est nécessaire d'y intégrer un installateur qui permet d'installer ledit logiciel sur l'ordinateur des utilisateurs. Pour répondre à ce besoin On a le logiciel gratuit **exe4j** qui donne les moyens de créer rapidement des installateurs pour les applications. Lorsque vous achetez install4j. exe4j est utile si vous voulez créer un lanceur Java sans installateur. Exe4j est une application logicielle simple à utiliser qui permet aux programmeurs de créer rapidement des applications Java au format EXE et de les intégrer dans Windows. Il est fait d'un assistant avec dix étapes nécessaires pour terminer la tâche.	

1.3. Environnement des langages de développements

Technologie	Description
	Java est un langage de programmation moderne développé par Sun Microsystems (aujourd'hui racheté par Oracle). Une de ses plus grandes forces est son excellente portabilité : une fois votre programme créé, il fonctionnera automatiquement sous Windows, Mac, Linux, etc. Java intègre les concepts les plus intéressants des technologies informatiques récentes dans une plate-forme de développement riche et homogène. ***Caractéristiques du JAVA :*** * Orienté Objet. * Distribué. * Multithread. * Robuste. * Dynamique * Portable. * Haute performance.
	Java Swing est un langage de programmation orienté interface graphique. Cette technologie est en faveur des applications internet riches qui se comportent sur des multiples plateformes. Java Swing évolue vers le multimédia à travers le traitement des images, la création de media Player, et l'intégration audio et vidéo. Ces applications ont l'avantage d'être exécuté sur le web, sur le bureau, (**Desktop**) ainsi que sur mobile, et même sur télévision.

Pour la réalisation de l'interface graphique de notre application on a utilisé deux bibliothèques AWT et Swing :

AWT

Abstract Window Toolkit (AWT) est une bibliothèque graphique pour Java, faisant partie de JFC (Java Foundation Classes). Cette bibliothèque a été introduite dès les premières versions

de Java ; depuis Java 2, la bibliothèque de gestion de fenêtre officielle est Swing. Toutefois, AWT sert encore de fondement à Swing, dans la mesure où de nombreuses classes Swing héritent de classes AWT.

AWT emploie les composants natifs de la plate-forme, alors que Swing utilise des composants en pur Java.

Swing :

Swing est une bibliothèque graphique pour le langage de programmation Java, faisant partie du package Java Foundation Classes (JFC), inclus dans J2SE. Swing constitue l'une des principales évolutions apportées par Java 2 par rapport aux versions antérieures.

Swing offre la possibilité de créer des interfaces graphiques identiques quel que soit le système d'exploitation sous-jacent, au prix de performances moindres qu'en utilisant Abstract Window Toolkit (AWT). Il utilise le principe Modèle-Vue-Contrôleur (MVC, les composants Swing jouent en fait le rôle du contrôleur au sens du MVC) et dispose de plusieurs choix d'apparence (de vue) pour chacun des composants standards.

Relation entre Swing et AWT :

Depuis les premières versions de Java, Abstract Window Toolkit (AWT) fournit une API indépendante du système d'exploitation pour mettre en œuvre des composants graphiques. Dans AWT, chaque composant est dessiné et contrôlé par un composant tiers natif spécifique au système d'exploitation. C'est pourquoi les composants d'AWT sont appelés composants lourds.

Au contraire, les composants Swing sont décrits comme légers. En effet, ils ne requièrent pas d'allocation de ressources natives de la part du gestionnaire de fenêtres sous-jacent, mais « empruntent » les ressources de leurs ancêtres.

Une grande partie de l'API Swing est une extension complémentaire à AWT plutôt qu'un remplaçant direct. L'affichage est fourni par Java2D, un autre composant des JFC. Cependant, l'usage conjoint de composants légers et lourds au sein d'une même fenêtre est généralement déconseillé à cause de problèmes de gestion de la profondeur

Chapitre 4 : Présentation du Jeu

1. Introduction

L'interface graphique est une partie très importante pour la réalisation d'une application convenable, et conviviale offrant un certain plaisir à l'utilisateur lors de sa navigation. Ainsi ce critère peut faire la différence entre une application et un autre bien qu'elles aient les mêmes fonctionnalités.

Voici maintenant un ensemble de captures d'écrans sur les principaux points d'entrées de l'application

2. Les interfaces graphiques

- **Lancement du jeu**

Lorsqu'on lance le jeu une fenêtre s'apparait se compose de :

-Une image comme arrière-plan.

-un texte qui donne une vue générale sur le jeu en donnant quelques instructions au joueur qui définissent les règles générales.

Au-dessus, on a positionné deux boutons :

-**BACK** : si le joueur veut quitter le jeu une fenêtre **JFrame**

Contient le message « voulez-vous vraiment quitter ? », le joueur peut confirmer son choix en cliquant sur le bouton **'' YES''**, ou annuler son choix en cliquant sur le bouton **''NO''**.

(L'image de panneau de confirmation).

Se lance pour confirmer son choix s'il veut vraiment quitter.

- **NEXT** : si le joueur clique sur ce bouton, il va le rediriger vers le panneau d'accueil du jeu.

- ## L’interface d’accueil

Il y a deux objets de type **JPanel** qui composent L’interface d’accueil de notre application qui sont les suivants :

→ Une petite barre qui comporte le Logo à gauche et des petits boutons de gestion de la fenêtre.

→ Un menu de type **JTabbedPane** qui comporte les panneaux principaux du jeu.

1-home : contient le plateau du jeu.

2-Option : panneau de configuration.

3-Help : panneau d’aide.

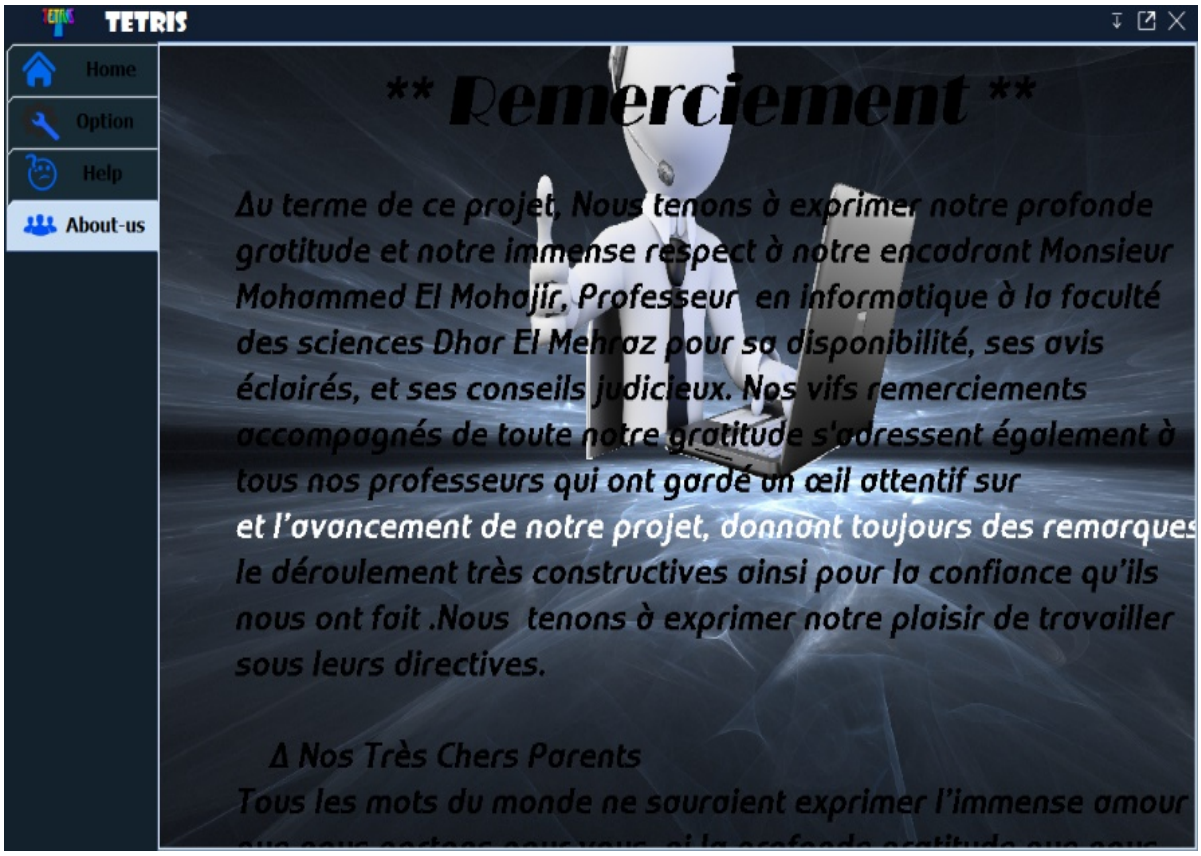
4-About-us : présente des informations à propos de ce travail.

Chaque panneau a ses méthodes spécifiques

About-us

Un panneau qui donne au joueur les coordonnées des développeurs de ce jeu (email, numéro de téléphone, nom …) qui lui permettre de nous contacter. Ainsi qu'un remercîment et une dédicace.

Ce panneau s'affiche de manière dynamique (un générique qui monte du bas vers le haut de manière verticale) pour lui donner une touche d'esthétique.

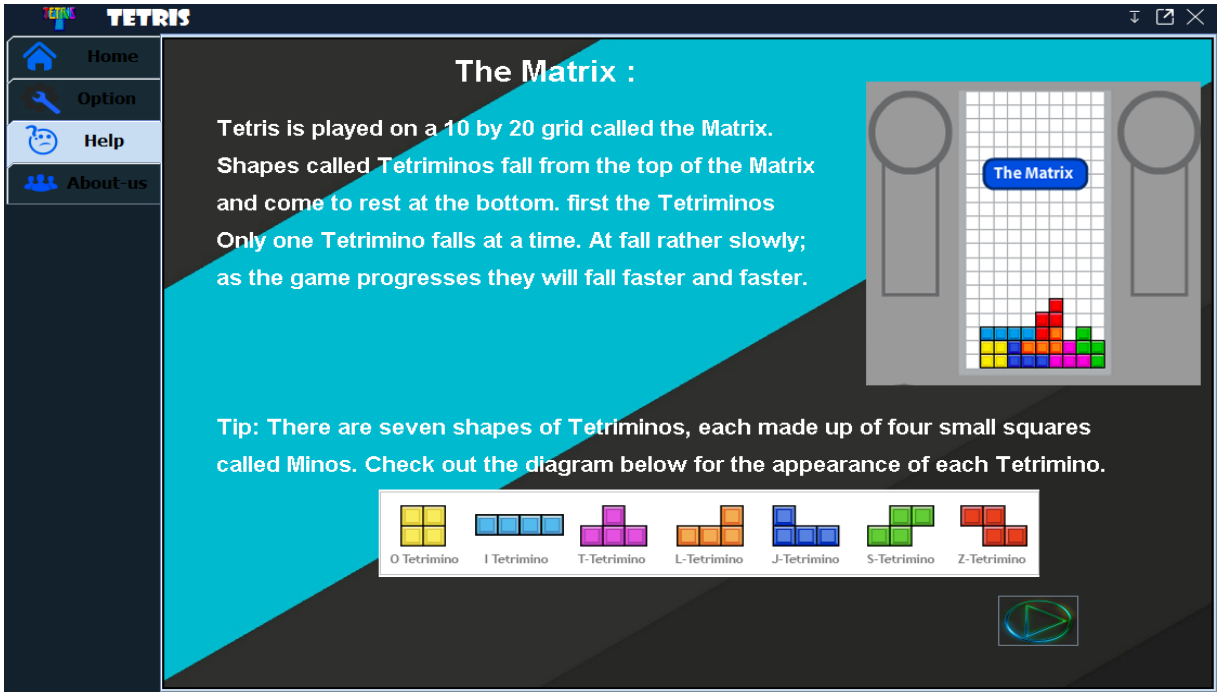


→ Mais comment cela fonctionne-il ?

-Tout d'abord ce panneau se compose des images ainsi qu'un texte, ensuite on a ajouté un **Timer** pour contrôler la vitesse du mouvement du texte.

- le mouvement de ce panneau s'arrête dès que l'utilisateur quitte le panneau existant à l'aide de la méthode **stop()**.

- ## Help

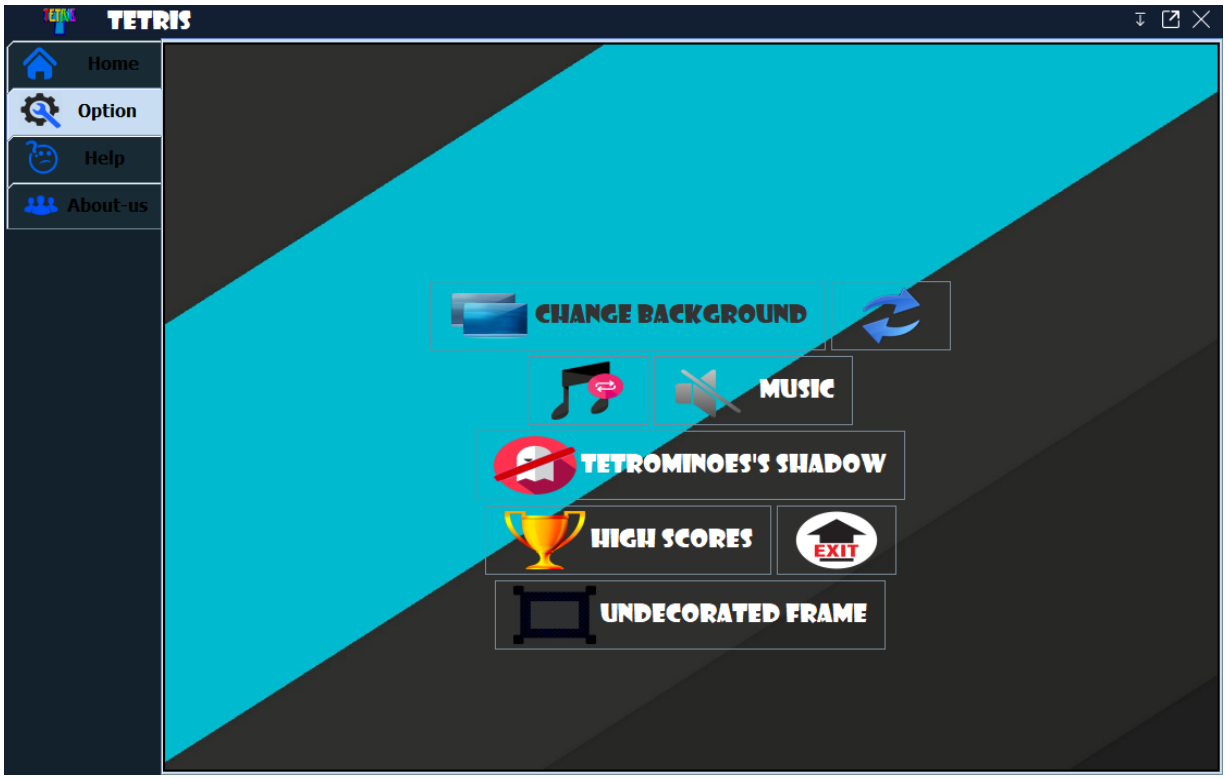


Ce panneau montre au joueur les règles du jeu de manière claire et simple avec des figures et des explications adéquates.

-Il est constitué de 5 parties et deux boutons qui lui permettent de naviguer entre elles.

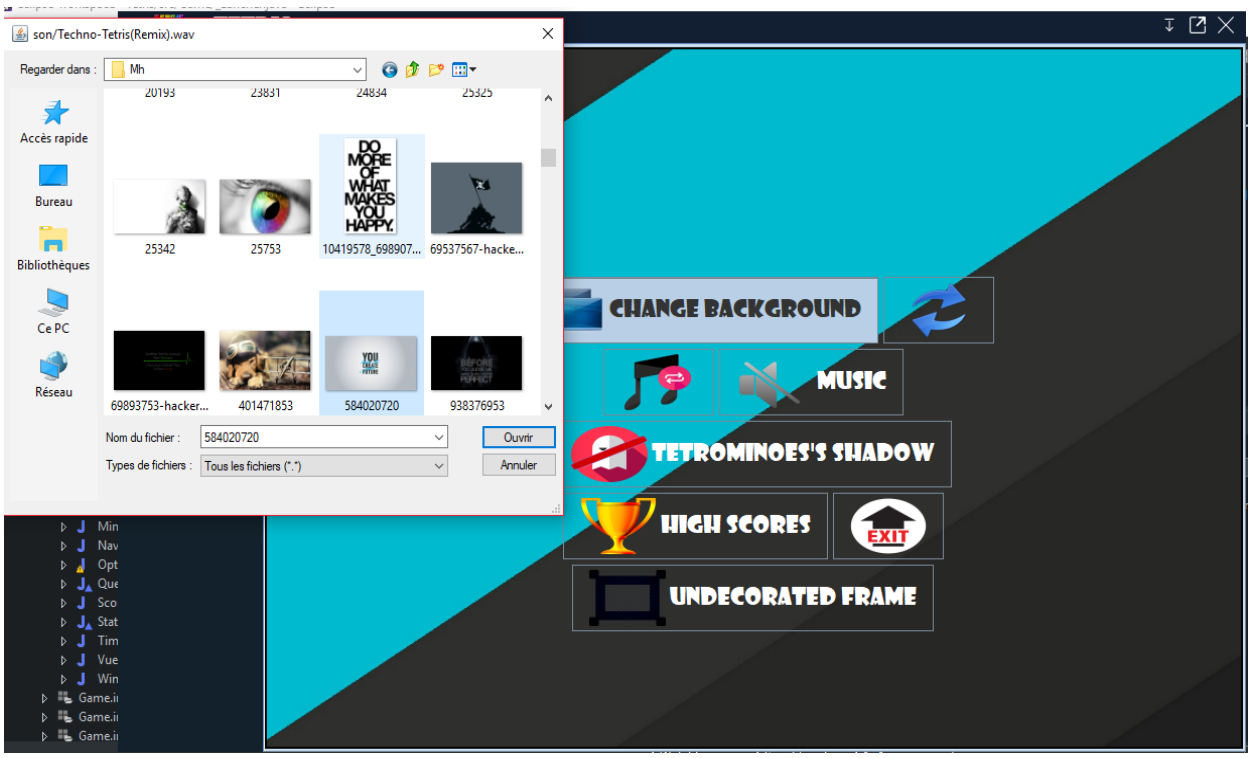
Chacune d'elles possède un identificateur unique de 0 à 4 qui permet au joueur de passer d'une partie à celle qui la suit afin de l'afficher.

- **Option**



Ce panneau est formé de huit boutons, chacun d'eux a un rôle spécifique :

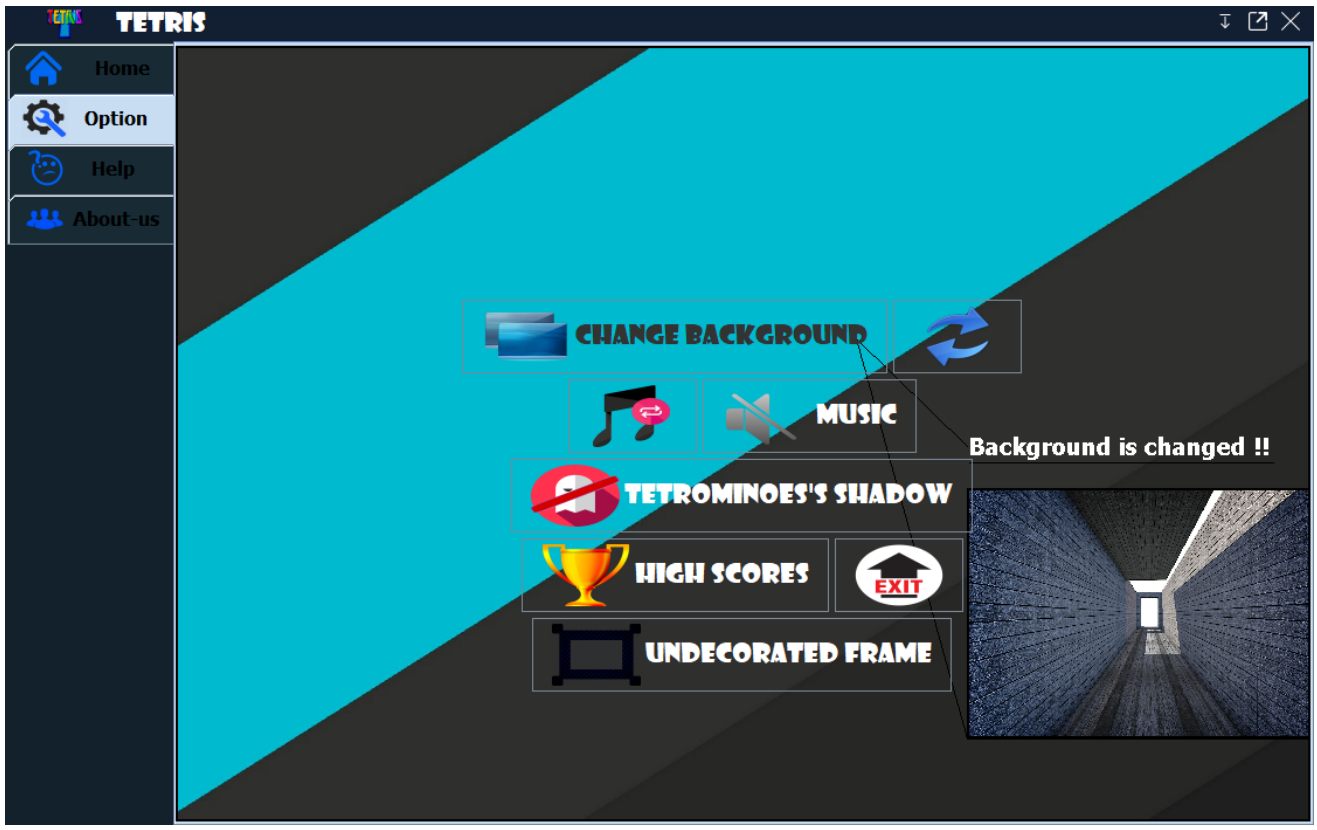
CHANGEBACKGROUND : ouvre une fenêtre de type FILEDialog, son rôle principal est de permettre au joueur de choisir l'image d'arrière-plan, en cliquant sur le bouton 'ok', alors l'image choisie sera affectée automatiquement à l'arrière-plan du jeu.



Juste à côté un bouton de modification aléatoire des images d'arrière-plan, en Effet, on a stocké 16 images repères avec des numéros (0→15).

Chaque clique incrémente l'identificateur dans la liste d'images préchargée.

Le fait de changer l'image à partir de ces deux boutons affiche automatiquement l'image courante de manière dynamique verticale dans le panneau (de haut en bas).



Music :

Cette fonctionnalité permet d'arrêter ou activer la musique pendant le jeu.

Avant :

Après :

Le bouton :

Permet de changer la musique.

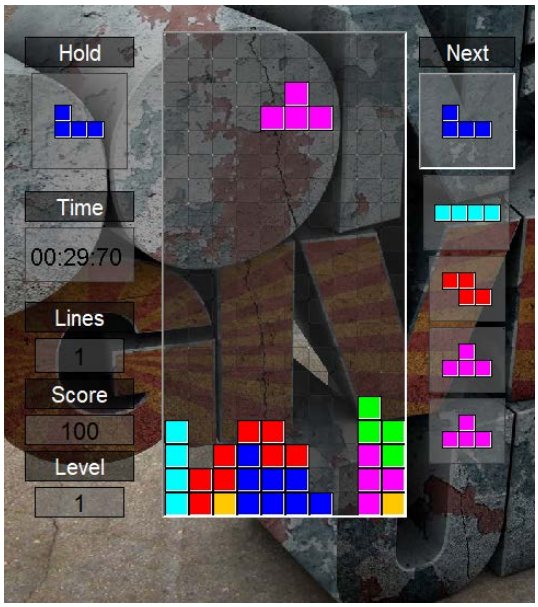
TETRIMINOES’S SHADOW :

Cette fonctionnalité permet de simuler l’ombre de la pièce courante, elle a pour but de faciliter le jeu en montrant la place de son positionnement.

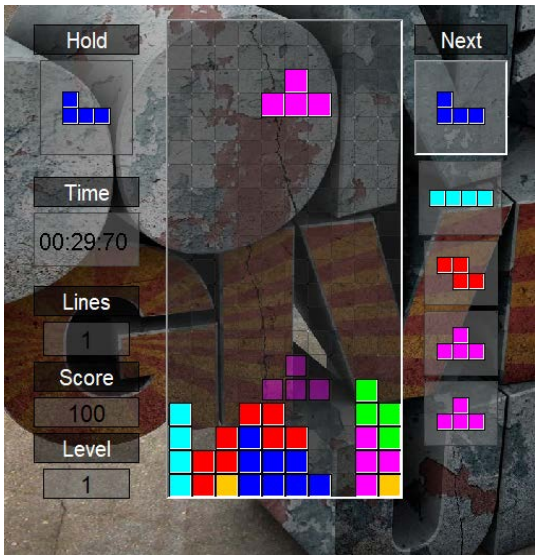
→ Comment ça fonctionne :

Si on clique le bouton, la variable booléenne **ghosted** prend comme valeur true, et elle va activer l’ombre de chaque pièce descendante dans la base de la zone du jeu.

Avant :



Après



HIGHSCORES :

Cette fonctionnalité permet de lister les dix meilleurs scores obtenus par les anciens joueurs.

Ce panneau de 'High Score' contient deux boutons pour naviguer entre les listes de score de chaque niveau.

Fonctionnalité :

Les données de chaque liste sont obtenues à partir d'un fichier qui se trouve dans notre application .EXE .

Quand on est dans la liste des scores de niveau facile (Easy), le bouton BACK

Change de couleur

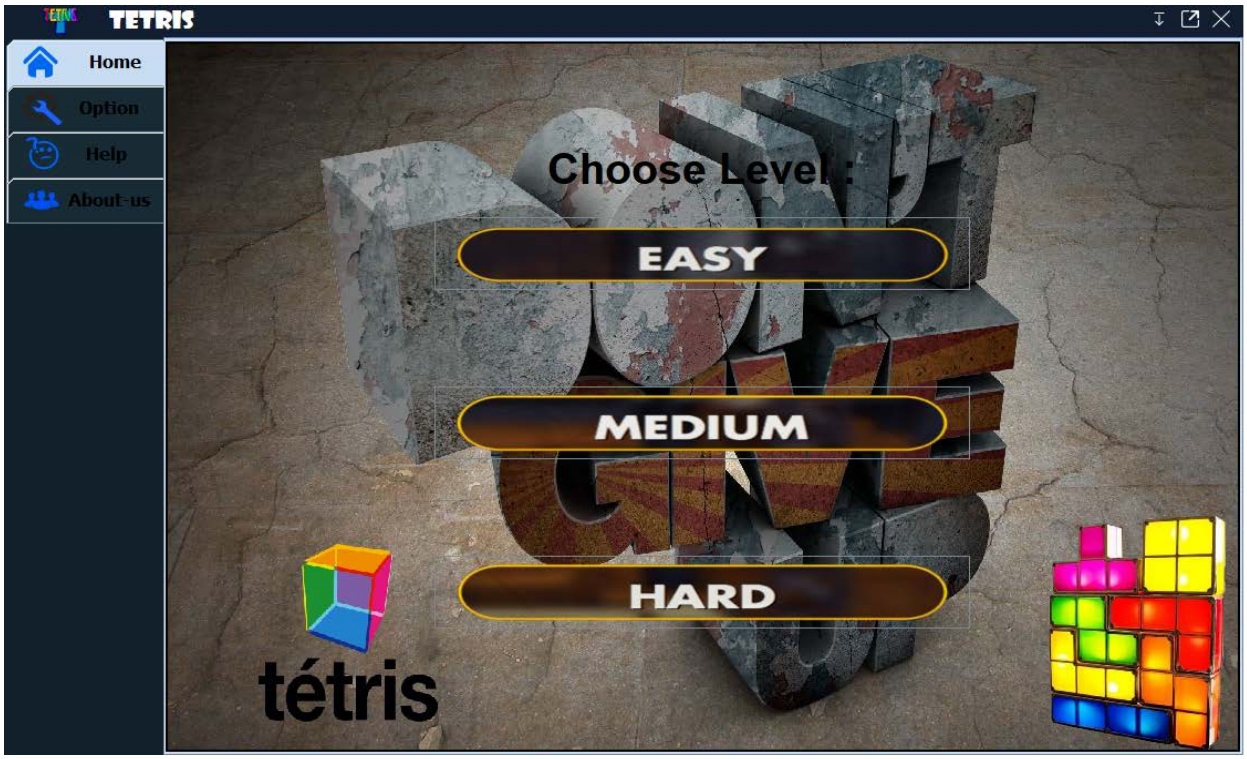
Et le redirige au panneau d'options.

UNDECORATED FRAME :

Afficher la barre de tâches de la fenêtre du jeu.

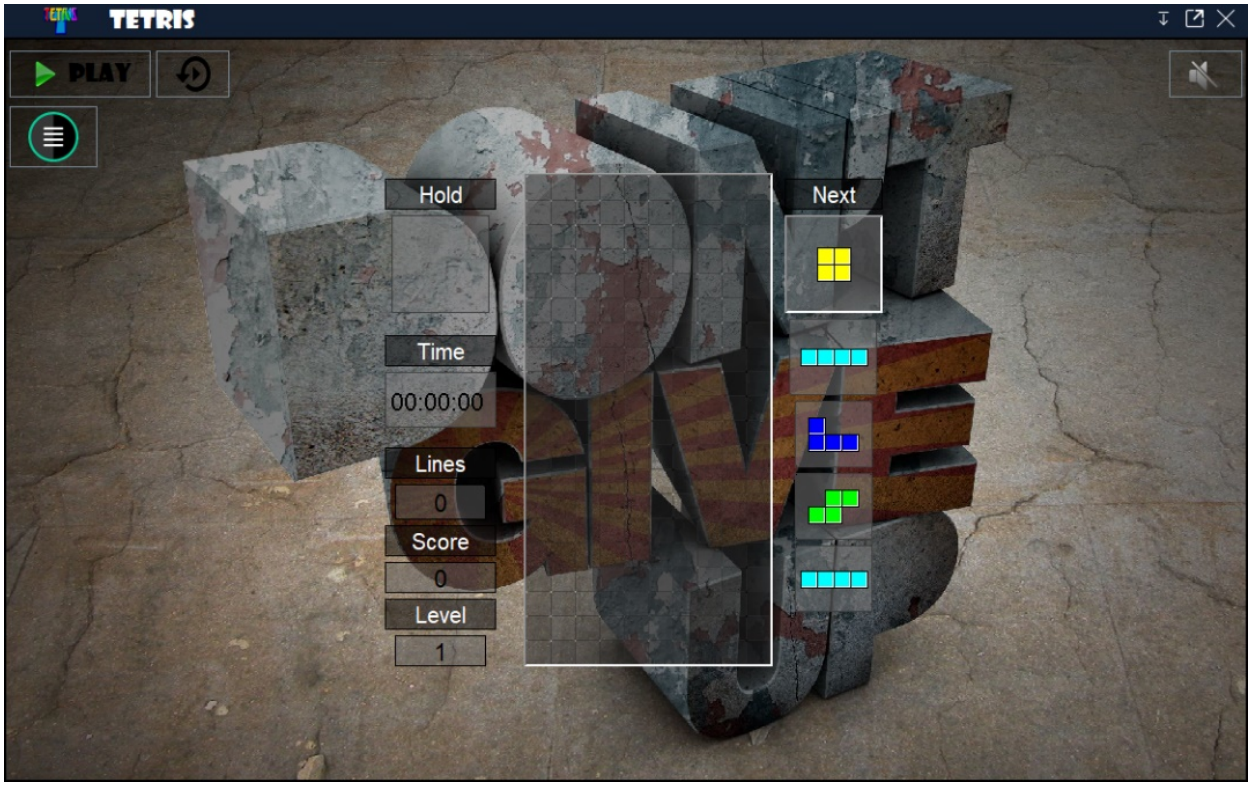
- ## Home

Le premier panneau qui s'affiche est **ChooseLevel**



Qui donne la possibilité de choisir le niveau qui représente la difficulté du jeu, après avoir cliqué sur l'un de ces trois boutons, pour rediriger le joueur vers le plateau du jeu.

- ## Le plateau de jeu



Le plateau contient :

→ En haut un Navigation Bar.

→ Au milieu un sous plateau qui représente l'espace du jeu.

Navigation Bar : possède 4 boutons :

-PLAY/PAUSE : permet de débuter le jeu

Ou bien l'interrompre temporairement

-RESET :

Réinitialiser le jeu

-SOUND : arrêter ou activer la musique.

-Menu :

Permet le retour à la barre de menu du jeu et affiche un panneau donnant la possibilité de reprendre la partie ou bien la réinitialiser.

- ## Le sous-plateau

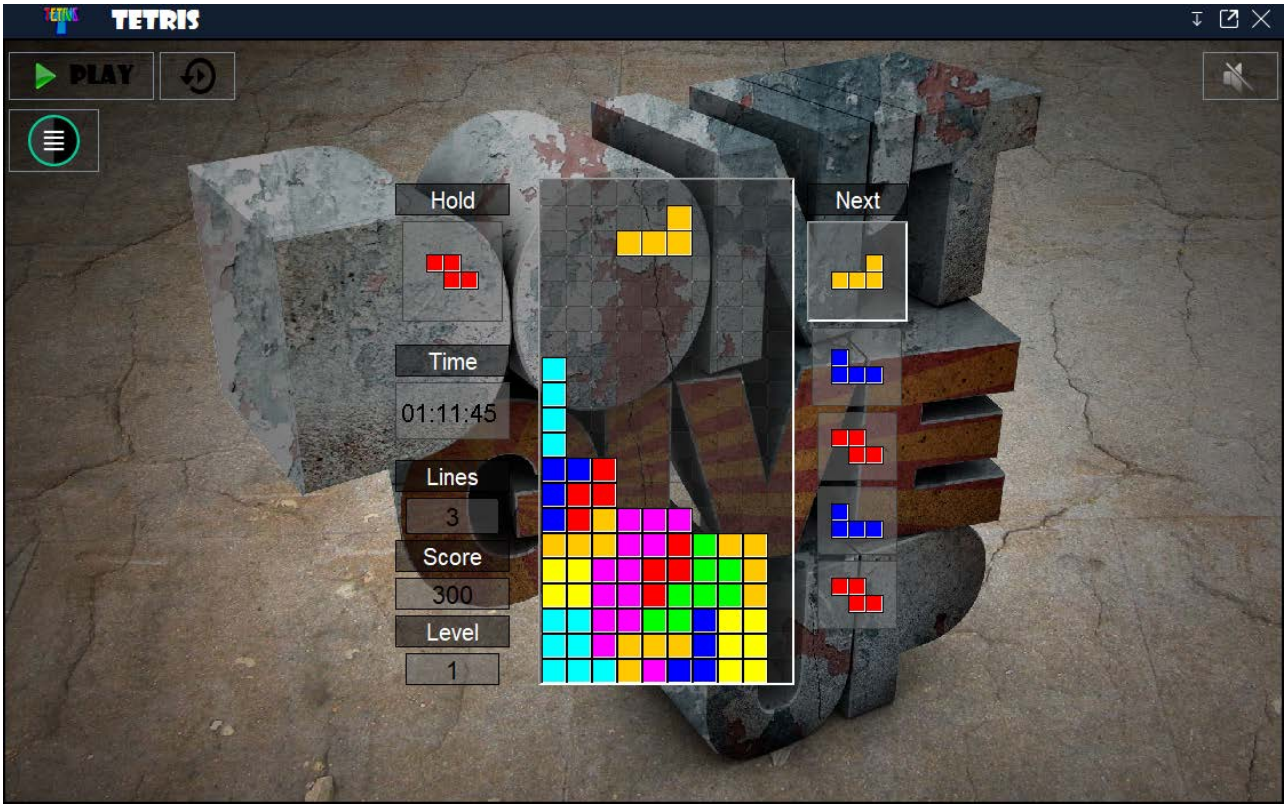
Le sous plateau du jeu est souvent appelé « puit » ou « matrice » ou « Well ». Il s'agit de l'espace dans lequel descendent les pièces. Il dispose toujours d'une grille en arrière-plan, visible ou non, dont les cases sont de la même grandeur que les carrés des pièces, et que celles-ci suivent dans leur chute. Il est également entouré par une armature appelée « tétrion », infranchissable, qui pose les limites du champ de jeu.

D'après le Tetris Guidline, la taille du champ de jeu en norme est de dix cases de largeur et de vingt cases de hauteur

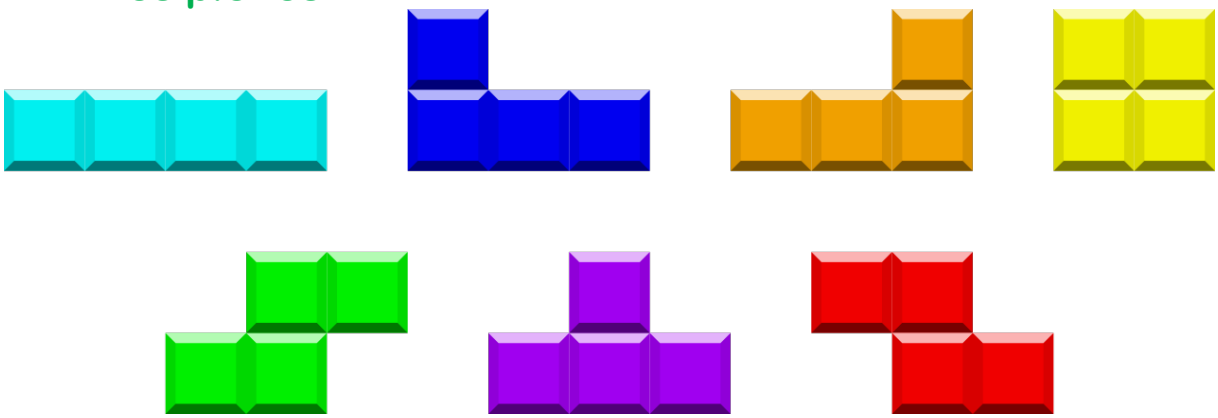
Voici un guide rapide des Règles de notre jeu vidéo :

Des pièces de Couleurs différentes descendent une après l'autre du haut de la grille, puis s'entassent sur les anciennes pièces qui se sont posées en bas. Le jeu s'arrête lorsque la dernière pièce introduite est bloquée en haut.

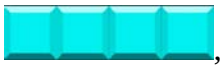
Chaque pièce est tirée aléatoirement dans la liste des 7 pièces possibles. Elles sont constituées de petits carrés qui sont toujours positionnés exactement sur les cases de la grille.

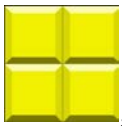


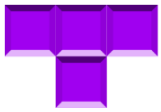
- **Les pièces**



Les pièces dans ces jeux sont des assemblages de quatre carrés.au total Il y a sept pièces :

 , forme **I**, aussi appelée « bâton », « ordinaire », « barre » . Elle est constituée de quatre carrés formant une ligne.

 , forme **O**, aussi appelée « carré » ou « bloc ». Elle est constituée de quatre carrés formant un méta-carré de 2x2.

 , forme **T**. Elle est constituée de trois carrés en ligne et d'un quatrième carré sous le centre.

, forme **L**. Elle est constituée de trois carrés en ligne et d'un quatrième carré à droite.

, forme **J**, aussi appelée « L'inversé » ou « gamma ». Elle est constituée de trois carrés en ligne et d'un quatrième carré à gauche.

, forme **Z**, aussi appelée « biais ». Elle est constituée de quatre carrés formant un méta-carré de 2x2, dont la rangée supérieure est glissée d'un pas vers la gauche.

, forme **S**, aussi appelée « biais inversé ». Elle est constituée de quatre carrés formant un méta carré de 2x2, dont la rangée supérieure est glissée d'un pas vers la droite.

Chacune des sept pièces dispose d'une couleur qui lui est propre. Il n'y a aucune couleur prédéterminée pour chacune des pièces.

Le joueur ne peut pas stopper la pièce en chute, mais il peut diriger celle ci avec les flèches latérales.

- ## Les mouvements latéraux

Le joueur peut à l'aide de certaines touches déplacer latéralement les pièces, c'est-à-dire vers la droite (touche : flèche en haut) ou vers la gauche. Cependant si la pièce touche les bords du jeu, le déplacement ne sera pas possible.

- ## Les rotations

Le joueur peut faire tourner plusieurs fois à gauche et/ou à droite selon la version, de 90° n'importe quel bloc pour le poser de la façon désirée pendant que le bloc descend.

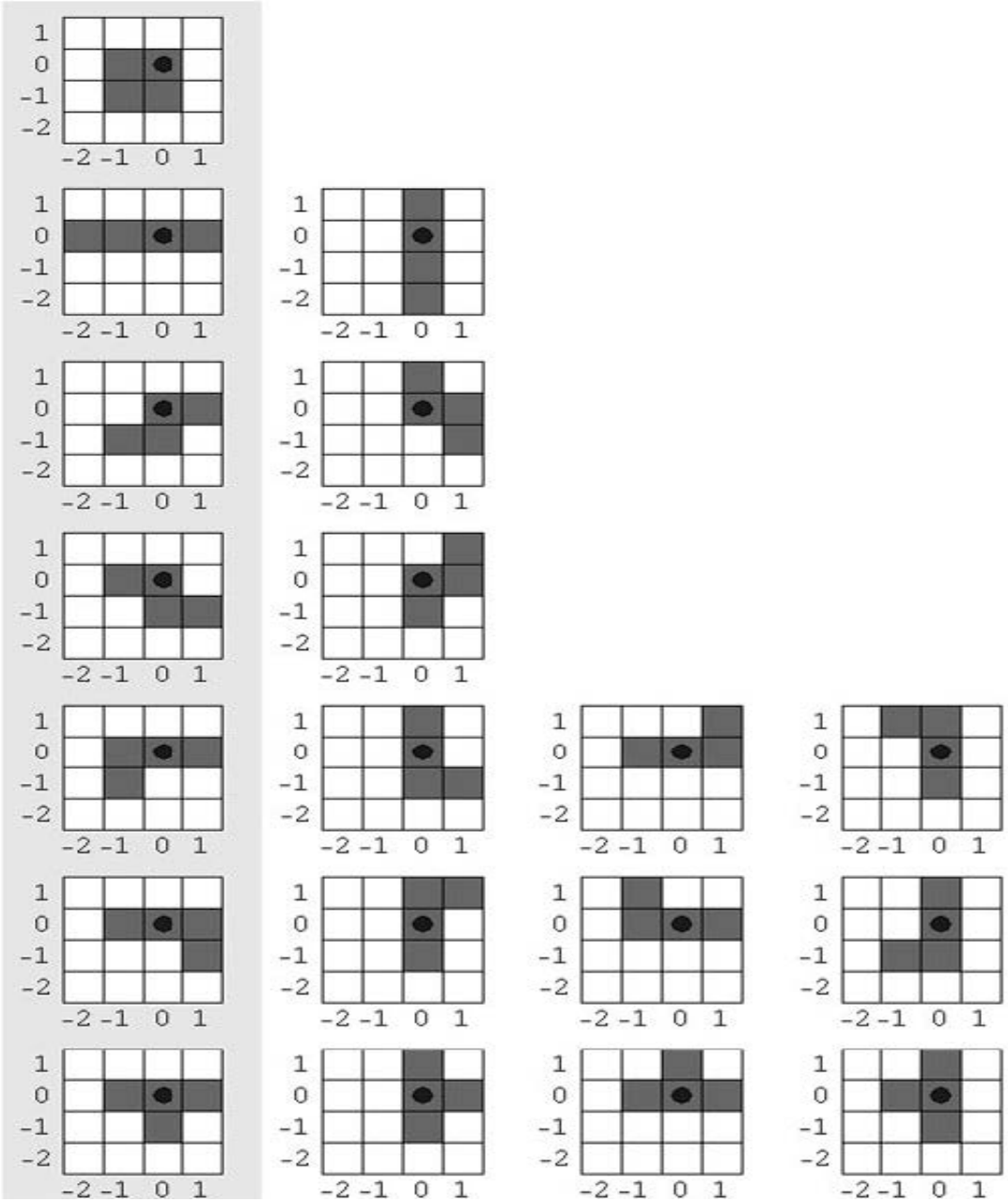


Figure : rotation de la pièce.

• Commandes du jeu

On utilise les touches directionnelles :

BAS : descente de la pièce

DROITE : déplacement de la pièce à droite

GAUCHE : déplacement de la pièce à gauche

HAUT : faire tourner la pièce

SHIFT ou **H :**garder la pièce.

SPACE : chute libre.

P : Pause / Résume.

• Suppression des lignes

Lorsque la pièce est bloquée par un obstacle situé dessous, les lignes complètes sont supprimées et les blocs supérieurs descendent vers le bas. Si après cela il reste encore un morceau de la pièce et que celle-ci n'est pas bloquée par un obstacle, elle continue à tomber et peut provoquer d'autres suppressions de lignes (c'est l'effet cascade). Enfin lorsque (le reste de) la pièce ne peut plus tomber, une nouvelle pièce apparait en haut de la grille et commence à descendre.

- ## Les Niveaux

La vitesse du jeu varie en fonction du niveau du jeu, à un niveau avancé les blocs tombent beaucoup plus vite, par conséquent, ils sont plus difficiles à manœuvrer.

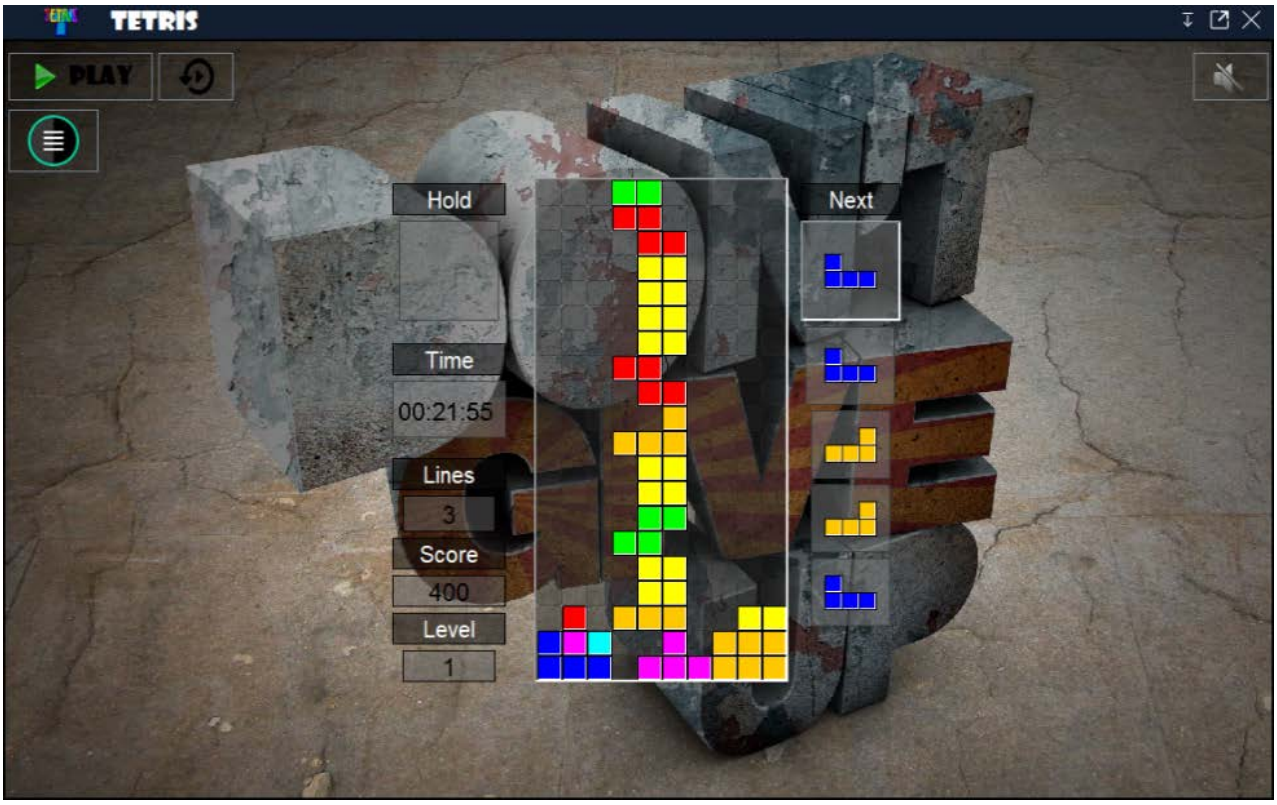
Notre Jeu comporte 10 niveaux : ils vont de 1 à 10 et déterminent la fréquence de chute des pièces (la vitesse à laquelle tombent les pièces).
Le score est calculé de manières différentes selon les versions du jeu. Il est ici déterminé de la façon suivante :

- ## Score

En faisant disparaître une ligne après l'autre, vous obtiendrez le minimum de points mais si vous en faites disparaître deux ou trois en même temps, vous pourrez gagner plus de points. Toutefois, l'objectif ultime est de faire disparaître quatre lignes d'un seul coup (ça s'appelle un 'Tetris') pour obtenir le score maximum.
Cependant, on ne peut réaliser un 'Tetris' que si l'on utilise le bloc en forme de barre en posant les autres pièces de manière à laisser un espace vertical

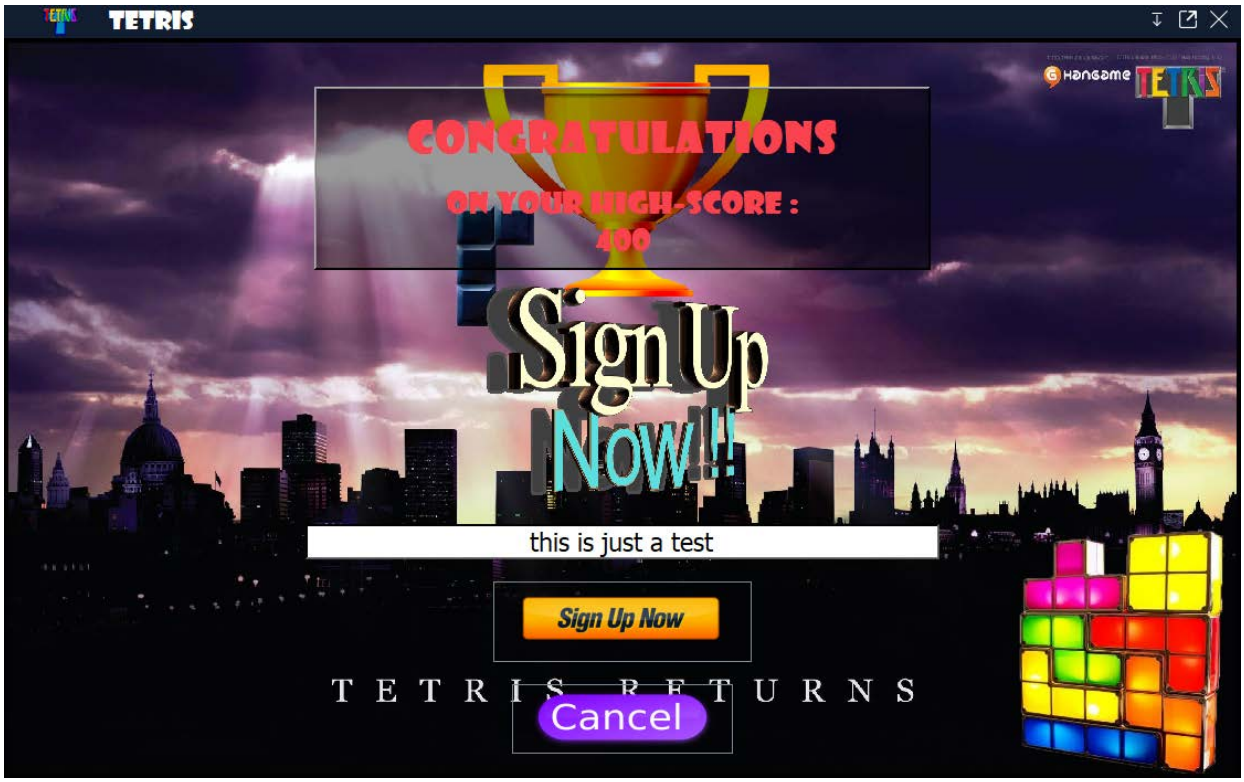
1 ligne : Single ➔ 100 points.
2 lignes : Double ➔ 300 points.
3 lignes : Triples ➔ 500 points.
4 lignes : Tetris ➔ 700 points.

- ## Game Over



Si le joueur échoue à faire disparaître des lignes, les blocs ne tarderont pas à s'accumuler et s'ils parviennent jusqu'en haut de l'écran, c'est le 'Game Over'.

Si le score obtenu est parmi les 10 meilleurs scores, un nouveau panneau demande au joueur d’enregistrer son nouveau score.



En cliquant le bouton « SignUp now », le score est enregistré dans la liste de TOP10.

Sinon un PANEL visualise les statistiques de la partie du jeu :

→Score

→Rows : le nombre de lignes supprimées.

→Lines per minutes : le quotient (Rows/minutes).

→minosLockedDown : le nombre de pièces qui ont tombe pendant la partie.

→Minos per minutes: minosLockedDown/minutes.

→Singles : le nombre de lignes individuelles écrasées.

→ Doubles : le nombre de lignes doubles écrasées.

→Triples : le nombre de lignes triples écrasées.

→Tetrises : le nombre de lignes quads écrasées.

- ## High Scores

Apres l'inscription, le score est enregistré dans le fichier relié à la difficulté du niveau.

Et le panneau correspond au high Scores s'affiche comme suit :

Et le joueur a toujours la possibilité de quitter le jeu.

3. Conclusion

Dans ce chapitre, nous avons décrit tous les aspects et les éléments graphiques de ce jeu, avec des captures illustrant les interfaces les plus essentielles.

Conclusion générale

Tout au long de ce rapport, nous avons présenté les différentes étapes de réalisation de ce jeu.

Ce projet, nous a donné l'opportunité de s'initier à la vie professionnelle dans un milieu réel et avoir un début d'expérience significative. Il nous a poussés vers un travail de groupe, résoudre les problèmes au cas où ils se présentent, être attentives aux indications de nos supérieurs, et être bien organisés pour accomplir ce projet dans les meilleurs délais.

En travaillant sur ce projet nous avons appliqué la majorité des acquis que nous avons reçu tout au long de notre cursus universitaire : de la modélisation UML jusqu'au principe de programmation orienté objet.

Au cours de la réalisation de notre projet, nous avons été contraints par quelques limites, notamment la contrainte du temps qui était relativement un obstacle devant l'ajout de certaines fonctionnalités.

Cependant, il était une occasion pour mettre en évidence et déployer sur le plan pratique nos connaissances en informatique.
Le projet peut être amélioré, et ce, en lui ajoutant quelques modules ou interfaces pour mieux l'adopter aux besoins de l'utilisateur et pour qu'il soit toujours fiable et pratique.

Webographie et bibliographie

http://www.oracle.com/technetwork/java/index.html

https://www.java.com/fr/

https://github.com/

https://sourceforge.net/

https://stackoverflow.com/

www.futura-sciences.com

http://www.mit.edu/people/nathanw/java/classes/

http://www.mathmaurer.com/info_java/14_graphics/code/info_java_code_14_graphics.html

http://www.java2s.com/Tutorial/Java/0240__Swing/CreatingFrameswithabackgroundimage.htm

https://codes-sources.commentcamarche.net/

https://java.developpez.com/telecharger/Codes-sources

https://myegy.tv/apps/windows/4991/Complete%20Java%20Masterclass

http://www.dicodunet.com/definitions/sports-loisirs/jeu-video.htm

http://www.granddictionnaire.com/ficheOqlf.aspx?Id_Fiche=8355614

https://www.oqlf.gouv.qc.ca/ressources/bibliotheque

https://dailygeekshow.com/9-anecdotes-tetris/

Pour les photos :

https://www.pexels.com/

https://unsplash.com/search/photos/4k

https://wallpapershome.com/

https://www.flaticon.com/

https://fontawesome.com/icons?from=io

Pour les Song :

https://www.youtube.com/watch?v=ZuurdGpRCuU

https://www.youtube.com/watch?v=sN8Cq5HEBug

https://www.youtube.com/watch?v=c5EyHpd3cPk

https://www.youtube.com/watch?v=MRq_pRbtSM4

Pour le PowerPoint :

https://www.huffingtonpost.fr/2013/11/12/jeux-video-bienfaits-sante_n_4264120.html

https://fr.statista.com/statistiques/778329/game-of-thrones-produits-populaires-internet-france/

NB : Tous les sons et les images son open source.