



## 1 Présentation

L'objectif de ce projet est de développer un clone du jeu **Space Invaders**. Tout le monde connaît certainement ce jeu, ou en tout cas certaines versions de ce jeu. Néanmoins, une petite présentation pour fixer les idées s'impose. Le déroulement du jeu s'effectue selon le *gameplay* suivant (extrait de Wikipedia, [http://fr.wikipedia.org/wiki/Space\\_Invaders](http://fr.wikipedia.org/wiki/Space_Invaders)) :

"Space Invaders est un shoot them up fixe en deux dimensions. Le joueur contrôle un canon laser qu'il peut déplacer horizontalement, au bas de l'écran. Dans les airs, des rangées d'aliens se déplacent latéralement tout en se rapprochant progressivement du sol et en lançant des missiles. L'objectif est de détruire avec le canon laser une vague ennemie, qui se compose de cinq rangées de onze aliens chacune, avant qu'elle n'atteigne le bas de l'écran. Le joueur gagne des points à chaque fois qu'il détruit un envahisseur. Le jeu n'autorise qu'un tir à la fois et permet d'annuler ceux des ennemis en tirant dessus. La vitesse et la musique s'accroissent au fur et à mesure que le nombre d'aliens diminue. L'élimination totale de ces derniers amène une nouvelle vague ennemie plus difficile, et ce indéfiniment. Le jeu ne se termine que lorsque le joueur perd, ce qui en fait le premier jeu sans fin.

Les aliens tentent de détruire le canon en tirant dessus pendant qu'ils s'approchent du bas de l'écran. S'ils l'atteignent ou arrivent jusqu'au sol, ils ont réussi leur invasion et le jeu est fini. De temps en temps, un vaisseau spatial apparaît tout en haut de l'écran et fait gagner des points bonus s'il est détruit. Quatre bâtiments destructibles permettent au joueur de se protéger des tirs ennemis. Ces défenses se désintègrent progressivement sous l'effet des projectiles adverses et de ceux du joueur. Le nombre de bâtiments n'est pas le même d'une version à l'autre."

Une capture d'écran est présentée à la figure 1. Sur cette dernière, on peut observer les différents éléments du jeu :

- le vaisseau : il peut se déplacer de gauche à droite ;
- les aliens : ils se déplacent de gauche à droite, et à chaque fois qu'ils touchent un bord, ils descendent d'un niveau vers le bas ;
- un missile : projectile envoyé par le vaisseau pour détruire les aliens.
- des éléments d'information.

Dans la description faite plus haut, les aliens peuvent également tirer. Nous ne considérerons pas ce point comme obligatoire dans le projet. De la même manière, l'animation observée lors de la destruction d'un alien n'est pas demandée.

Afin de vous faciliter le travail, le découpage suivant du projet est proposé :

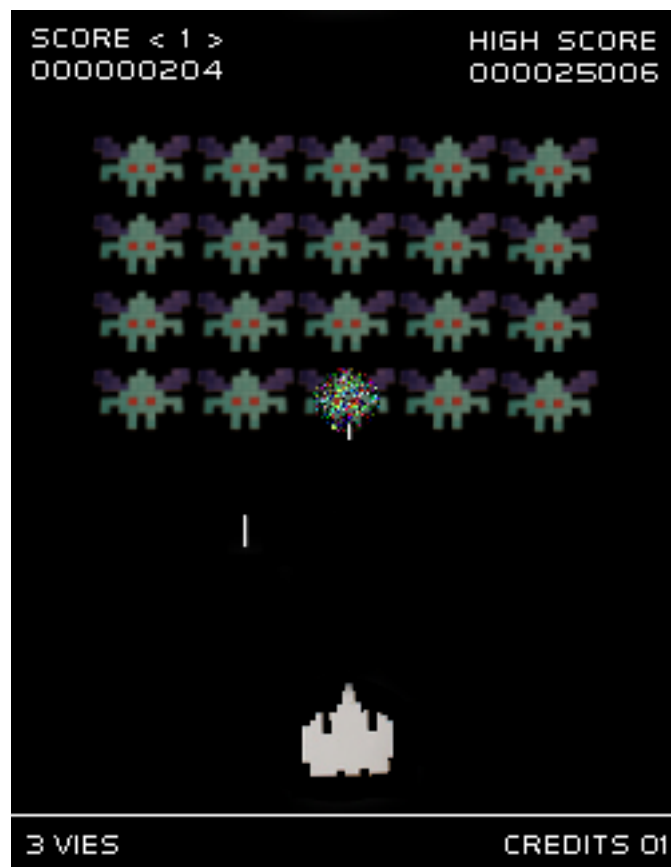


FIGURE 1 – Capture d’écran d’une reproduction du jeu *Space Invaders*.

1. construction du modèle,
2. construction de l'interface,
3. animation des éléments,
4. détection de collision,
5. affichage des éléments d'information,
6. configuration de l'application

## 2 Construction du modèle

La première étape dans la construction du jeu est la définition du modèle du jeu. L'analyse du jeu met en évidence, au minimum, les classes suivantes : Spaceship, Alien, Bullet et Game.

Chacun des trois premiers éléments est caractérisé par des capacités de déplacements ainsi qu'un sprite (image affichée). La classe Game permet en outre de conserver les différents éléments indispensables au jeu ainsi que d'autres informations telles que le score.

## 3 Interface principale

L'interface du jeu doit comporter deux parties. La première est l'affichage du plateau de jeu et la deuxième est le panneau d'information indiquant l'état courant du jeu. Cette organisation est représentée de manière schématique sur la figure 2. Le choix de l'organisation de l'aire d'information est laissé à votre appréciation en fonction des informations que vous aurez prises en compte dans votre modèle.

Également, l'ajout d'un menu est nécessaire afin de pouvoir effectuer les actions suivantes : *Nouvelle partie*, *Pause*, *Quitter* et obtenir de l'aide sur l'utilisation du jeu. Ces différents éléments seront à compléter au fur et à mesure du développement du jeu.

La mise en place de l'aire de jeu nécessite l'utilisation du modèle afin de le dessiner. Pour cela, vous pourrez utiliser par défaut les sprites fournis (voir figure 3).

À l'issue de cette partie, vous devez avoir une interface permettant d'afficher l'interface principale avec son menu, ses deux zones d'affichage ainsi que les aliens et le vaisseau de manière statique.

## 4 Animation des éléments et interaction avec l'utilisateur

L'animation des éléments du jeu doit mettre en oeuvre la notion de boucle de jeu afin de rafraîchir régulièrement l'affichage. Pour cela, je vous invite à consulter le tutoriel présent à l'adresse suivante : <http://z/etcode.com/tutorials/javagamestutorial/animation/>.

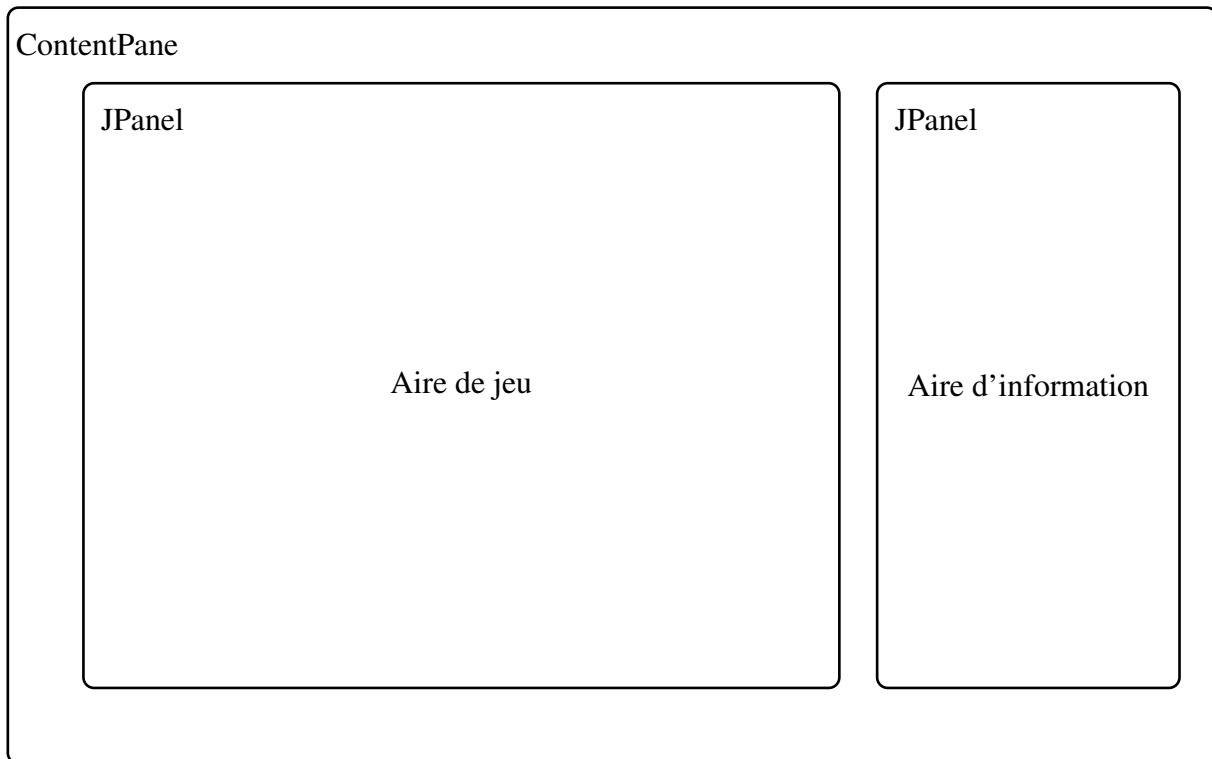


FIGURE 2 – Architecture de l'interface graphique principale

En appliquant ce tutoriel, vous devez mettre en place le déplacement autonome des aliens. Notez que cela se traduit par l'écriture d'une boucle de jeu, exécutée à intervalle régulier qui va demander la mise à jour des entités présentes dans le jeu et ensuite rafraîchir l'affichage.

L'étape suivante consiste à permettre au vaisseau de se déplacer en fonction des actions de l'utilisateur. Il s'agit donc de mettre en oeuvre un *KeyListener* permettant d'interpréter les appuis de touche suivants :

- flèche gauche/droite : déplacement du vaisseau,
- espace : envoi d'un missile
- 'p' : mise en pause du jeu

Comme introduit précédemment, l'appui sur la barre espace doit provoquer l'envoi d'un mis-



FIGURE 3 – Sprites utilisables pour les aliens, le vaisseau et les projectiles

sile. Cela se traduit par la création d'un objet *Bullet* qui doit être ajouté au jeu, et qui va ensuite se déplacer.

## 5 Détection des collisions

La détection des collisions est un élément important du jeu qui va permettre de prendre en compte qu'un alien est touché (et donc comptabiliser des points pour l'utilisateur) et également vérifier si les aliens arrivent en bas de l'écran, auquel cas l'utilisateur a perdu.

## 6 Affichage des éléments d'information

Au cours du jeu, certains éléments du modèle évoluent et nous souhaitons permettre à l'utilisateur de consulter ces informations. Par exemple, le score, le nombre d'aliens encore présents, le nombre de vies restantes (si cela est géré) peuvent être affichés dans le panneau d'information de l'interface principale.

La mise en oeuvre de cet affichage correspond à une vue du modèle représenté par le jeu. Il devient alors naturel d'utiliser le pattern MVC pour obtenir une synchronisation entre la vue et le modèle et faire évoluer les informations au cours de la partie.

## 7 Configuration du jeu

Enfin, afin de rendre le jeu moins austère, il est demandé d'offrir un contrôle sur certains paramètres à l'utilisateur :

- modifier de l'image de fond,
- modifier des sprites utilisés (aliens, vaisseau, missile)
- changer l'intervalle entre les tirs
- modifier le nombre d'aliens (ligne et colonne)
- adapter la vitesse de déplacement des aliens

Le contrôle de ces différents éléments doit passer par la création d'une seconde fenêtre accessible par un menu. Encore une fois, le pattern MVC devra être mis en place pour gérer la synchronisation entre données du modèle et vue.

## 8 Travail à réaliser

Vous devez réaliser l'application décrite précédemment. La note du projet sera fonction, bien entendu, de la qualité de la modélisation employée, de la qualité d'écriture du code source, mais également du niveau de finalisation du jeu.

De plus, suivant votre aisance, vous pouvez également ajouter des éléments qui n'auraient pas été décrits ici (niveaux supplémentaires, sauvegarde de la configuration, ...). Ces éléments ne seront pris en compte que si tout le reste est fonctionnel.

Vous devrez rendre une archive contenant les éléments suivants :

- le code source de l'application ;
- un fichier jar exécutable ;
- un rapport d'une dizaine de pages décrivant votre réalisation **au format PDF** ;
- une vidéo de démonstration du fonctionnement de votre application (screencast).

L'archive devra être nommée de la manière suivante : *NOM\_prenom.zip*. N'utilisez pas d'accent dans le nom de l'archive.

Le rapport doit bien entendu être structuré (introduction, ...) et comporter un jeu d'essai de votre application. L'objectif de ce rapport n'est pas d'écrire un catalogue des différents composants utilisés mais de présenter votre application, son fonctionnement ainsi que les difficultés rencontrées et le niveau de réalisation final atteint.

Ce travail est à réaliser en binôme.