

Evenat Théo
Hervault Antoine
Prieur Maxime
Groupe B

Rapport de projet informatique



Sommaire

I. Descriptif du cahier des charges	2
A. Descriptif détaillé.....	2
B. Liste des fonctionnalités attendues	2
C. Les différentes versions prévues	3
 II. Conception globale.....	4
A. Conception initiale	4
a. <i>Analyse descendante</i>	4
b. <i>Types de données définis et signature des procédures</i>	4
B. Conception finale	6
a. <i>Analyse descendante</i>	6
b. <i>Types de données définis et signature des fonctions et procédures</i>	7
c. Changement par rapport à la conception globale initiale	9
 III. Guide d'utilisation du programme	11
 IV. Travail du groupe	11
A. Répartition du travail	11
B. Planning et délais à respecter	12
C. Difficultés rencontrées	12
D. Perspectives d'évolution du programme	12

I. Descriptif du cahier des charges

Le cahier des charges a été réalisé au début de ce projet informatique dans le but de présenter le projet, ses fonctionnalités mais également pour fixer les délais à respecter dans la réalisation des différentes versions de notre programme.

A. Descriptif détaillé

Notre projet informatique « Space Invaders » est un jeu de type « shoot'em up » dans lequel l'utilisateur pilote un vaisseau et dont l'objectif est de détruire des vagues d'ennemis.

Le joueur contrôle son vaisseau à l'aide des touches directionnelles du clavier. Celles-ci lui permettent de se déplacer de droite à gauche et de tirer des missiles.

Les aliens se déplacent horizontalement jusqu'à atteindre le bord de la fenêtre, ils descendant alors une ligne et repartent horizontalement dans l'autre sens. Les aliens tirent également des missiles de façon aléatoire.

Si l'ensemble des ennemis est détruit, une nouvelle vague d'aliens apparaît, celle-ci se déplace plus rapidement que la précédente vague, le jeu devient alors plus difficile.

Le joueur peut abriter son vaisseau des tirs ennemis derrière des protections, qui sont détruites si elles sont touchées par un missile.

Au cours d'une partie, le joueur cumule des points et se constitue un score. Le joueur gagne des points en détruisant des ennemis ou encore en touchant avec ses missiles des soucoupes qui apparaissent à intervalles de temps réguliers au cours de la partie.

La partie s'arrête lorsque le joueur est touché par un missile ennemi ou bien si la ligne d'ennemis a réussi à atteindre la ligne du joueur.

B. Liste des fonctionnalités attendues

- Menu avec tableau des 5 meilleurs scores associés au nom des joueurs, une touche pour lancer le jeu et une autre pour le quitter
- Déplacement latéral du joueur
- Déplacement des ennemis au cours de la partie
- Tirs de missiles par le joueur et par les ennemis
- Destruction des ennemis touchés par un missile
- Destruction progressive des défenses présentes devant le joueur
- Passage aléatoire d'une soucoupe qui rapporte un bonus de points si celle-ci est touchée par un tir du joueur
- Vitesse progressive dans le niveau (en fonction du score et de la destruction totale des envahisseurs)
- Affichage du score à l'écran
- Enregistrement potentiel du score et du pseudo à la fin de la partie

C. Les différentes versions prévues

- **Version n°1, date prévue : 9 octobre 2017**

Fonctionnalités implémentées :

- Limite de la zone de jeu
- Déplacement du joueur
- Tir de missiles par le joueur
- Déplacement d'une ligne d'ennemis
- Destruction des ennemis s'ils sont touchés

- **Version n°2, date prévue : 6 novembre 2017**

Fonctionnalités implémentées :

- Comptage et affichage du score
- Déplacement de l'ensemble des lignes d'ennemis
- Tirs aléatoires des ennemis
- Ajout des défenses protégeant le joueur

- **Version n°3, date prévue : 27 novembre 2017**

Fonctionnalités implémentées :

- Création du menu
- Ajout des effets sonores et visuels
- Destruction progressive des défenses
- Ajout de l'accélération du jeu
- Enregistrement du score en fin de partie

- **Version n°4, date prévue : 11 décembre 2017**

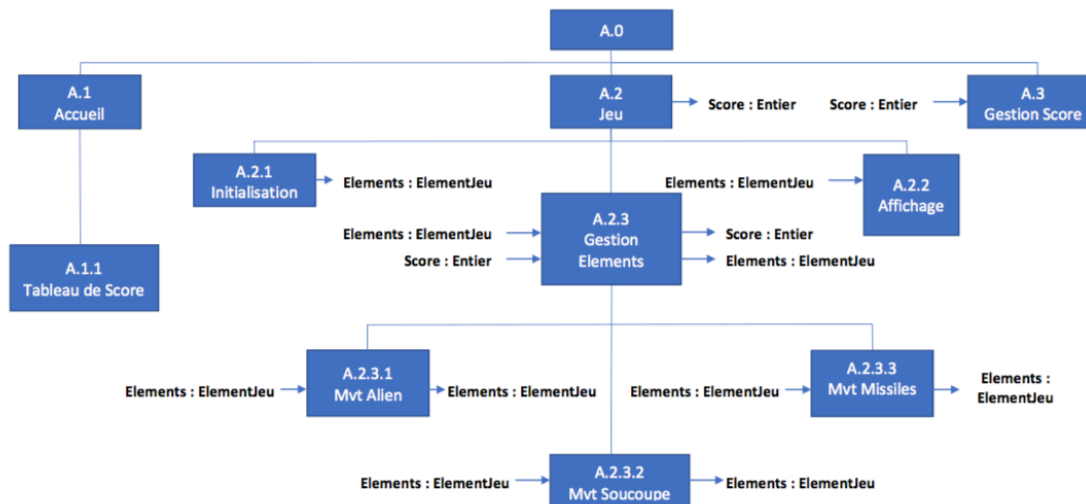
- Correction des bugs

Ce planning a pu être suivi convenablement au cours de l'avancée de notre projet. Certaines fonctionnalités ont même été implémentées dans des versions antérieures à celle où elles devaient initialement être codées. Malheureusement, le cours tardif sur le SDL nous a conduit à remanier de nombreuses procédures, notamment concernant l'affichage et le déplacement des éléments puisque celles-ci avaient déjà été codées.

II. Conception globale

A. Conception initiale

a. Analyse descendante



b. Types de données définis et signature des procédures

Les différents types définis :

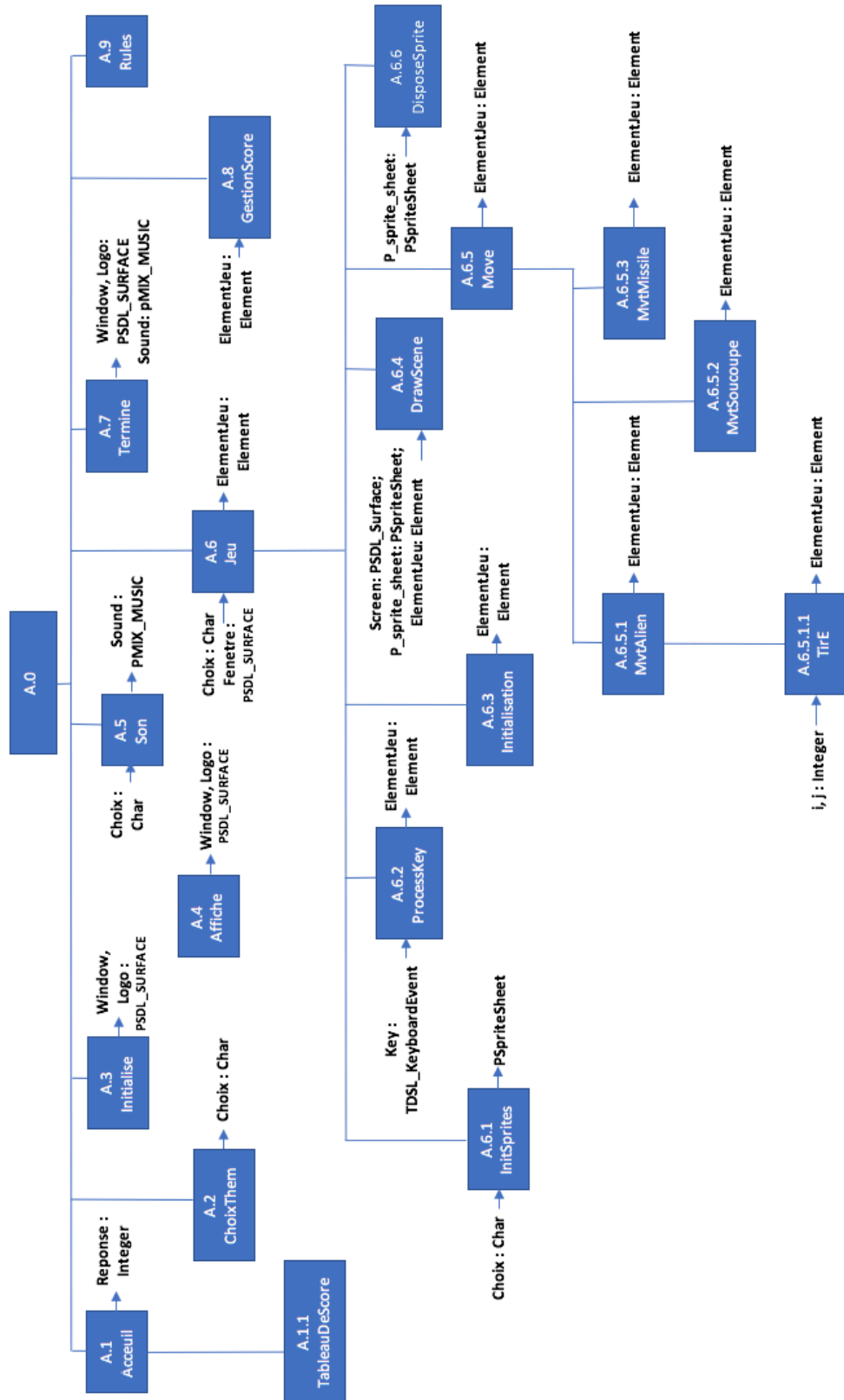
- **Type Coordonnee = Structure**
 - PosX : Entier ;
 - PosY : Entier ;
 - FinStructure ;
- **Type Emplacement = Tableau [1..20] de Coordonnee ;**
- **Type Element = Structure**
 - Grille : Tableau [1..15,1..50] d'Entier;
 - nbrMissileAllie, nbrMissileAlien, nbrAlien : Entier ;
 - MissileAllie, MissileAlien, Alien : Tableau [1..20] de Coordonnee ;
 - FinStructure ;
- **Type Scores = Structure**
 - Score : Entier ;
 - Pseudo : Chaîne de Caractères ;
 - FinStructure ;
- **Type TabScore = Structure**
 - Tab : Tableau [1..5] de Scores ;
 - NbrScore : Entier ;
 - FinStructure ;

Les signatures des procédures implémentées :

- **Procédure Affichage (E ElementJeu : Element ; Score : Entier)** : Affiche les différents éléments du jeu comme le vaisseau du joueur, les envahisseurs ou encore les missiles tirés par le joueur
- **Procédure Initialisation (S ElementJeu : Element)** : Initialise la grille de jeu en plaçant les différents éléments du jeu
- **Procédure MvtAlien (E/S ElementJeu : Element)** : Gère le déplacement des aliens
- **Procédure MvtMissiles (E/S ElementJeu : Element)** : Gère le déplacement des missiles
- **Procédure MvtSoucoupe (E/S ElementJeu : Element)** : Gère l'apparition et le déplacement de la soucoupe « bonus »
- **Procédure GestionElements (E/S ElementJeu : Element; score : Entier)** : Appelle les différentes procédures de mouvement, gère le déplacement de l'ensemble des éléments du jeu
- **Procédure Jeu (S Score : Entier)** : Base du programme, gère l'ensemble des fonctionnalités du jeu
- **Procédure GestionScore (E Score : Integer)** : Compare le score établi aux différents scores déjà existants pour le classer et le rentrer ou non dans le tableau de scores
- **Procédure TableauDeScore()** : Affiche le tableau de scores
- **Procédure Accueil()** : Appelle la procédure TableauDeScore, affiche le menu d'accueil du jeu

B. Conception finale

a. Analyse descendante



b. Types de données définis et signature des fonctions et procédures

Les différents types définis :

- **Type TSpriteSheet = Structure**
Earth, down, badguy, MissileAlien, MissileAllie,
Protection, Soucoupe, Alien1 : PSDL_Surface ;
FinStructure
- **Type PSpriteSheet = ^TSpriteSheet**
- **Type direction = (gauche, droite)**
- **Type coord = Structure**
x, y : Entier ;
FinStructure
- **Type Element = Structure**
Score : Entier ;
Grille : Tableau [1..19,1..50] d'Entier ;
CompteurSoucoupe, nbrMissileAllie, nbrMissileAlien, nbrAlien1, nbrAlien2,
NbrProtection : Entier ;
MissileAllie, MissileAlien, Protection : Tableau [1..20] de Coord ;
PresenceSoucoupe : Booléen ;
Soucoupe, Vaisseau : Coord ;
Alien : Tableau [1..15,1..2] de Coord ;
FinStructure
- **Type Scores = Structure**
Score : Entier ;
Pseudo : Chaîne de caractères ;
FinStructure
- **Type TabScore = Structure**
Tab : Tableau [1..5] de Scores ;
NbrScore : Entier ;
FinStructure

Les signatures des fonctions et procédures implémentées

- **Procédure Accueil** (S Réponse : Entier) : Affiche le menu, appelle la procédure TableauDeScore et récupère le choix du joueur pour continuer, afficher les règles du jeu ou pour quitter la partie.
- **Procédure ChoixTheme** (S Choix : Caractère) : Permet de sélectionner le thème de jeu.
- **Procédure Rules** () : Affiche les règles du jeu.
- **Procédure Initialise** (S window, Logo : PSDL_SURFACE) : Initialise des éléments de l’affichage.
- **Procédure Affiche** (S window, logo : PSDL_Surface) : Affiche le logo Space Invaders.
- **Procédure Son** (E choix : Caractère ; S sound : pMIX_MUSIC) : Lance la musique en fonction du thème choisi par le joueur.
- **Procédure TableauDeScore** () : Chargé depuis un fichier, affiche les 5 meilleurs scores avec les noms des joueurs associés.
- **Procédure Jeu** (E Choix : Caractère ; E fenêtre : PSDL_Surface ; S ElementJeu : Elément) : Pierre angulaire du programme, gestion de chaque niveau, et gestion des déplacements du joueur.
- **Fonction InitSprites** (E Choix : Caractère) : Charge les textures de jeu.
- **Procédure Initialisation** (E/S ElementJeu : Elément) : Initialisation des différents éléments.
- **Procédure DrawScene** (E screen : PSDL_Surface ; p_sprite_sheet ; ElementJeu : Elément) : Affiche tous les éléments du jeu.
- **Procédure Ecrire** (screen : PSDL_Surface ; txt : chaine de caractère : x, y, taille : Entier) : écrit un texte sur la fenêtre SDL.
- **Procédure Move** (E/S ElementJeu : Eléments) : Prise en compte des déplacements des éléments mobiles, analyse et gestion des interactions entre éléments du jeu, mise à jour du score.
- **Procédure MvtAlien** (E/S ElementJeu : Element) : Gestion du déplacement des Aliens.
- **Procédure MvtSoucoupe** (E/S ElementJeu : Element) : Gestion déplacement des soucoupes bonus.

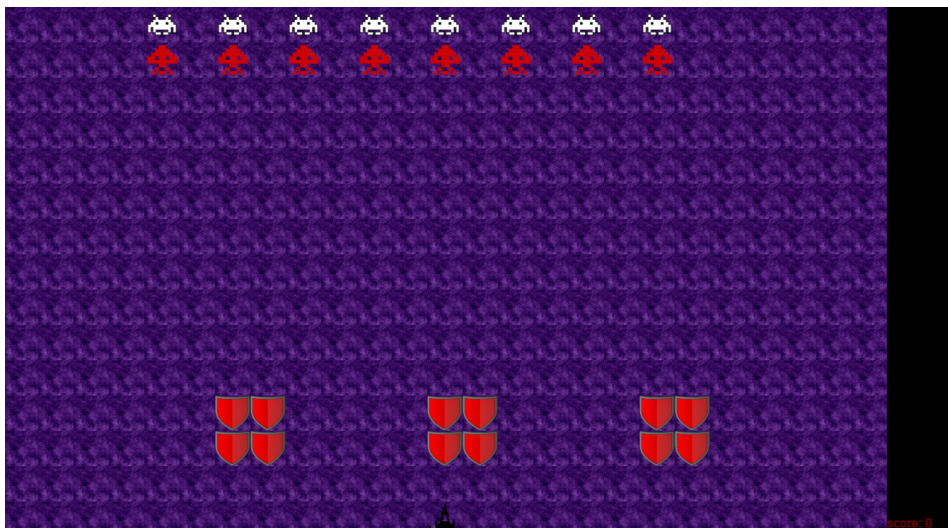
- **Procédure MvtMissiles** (E/S ElementJeu : Elément) : Gestion déplacement Missiles.
- **Procédure TirE** (E i, j : Entier ; E/S ElementJeu : Element) : Détermine si un Alien tire un missile (Probabilité).
- **Procédure ProcessKey** (key : TSDL_Keyboard ; E/S ElementJeu : Elément) : Gestion des actions du joueur.
- **Procédure termine** (E/S window, logo : PSDL_Surface ; sound : pMIX_MUSIC) : quitte proprement le SDL.
- **Procédure GestionScore** (E Score : Entier) : Gestion du score en fonction d'un fichier contenant le tableau de score.

c. Changement par rapport à la conception globale initiale

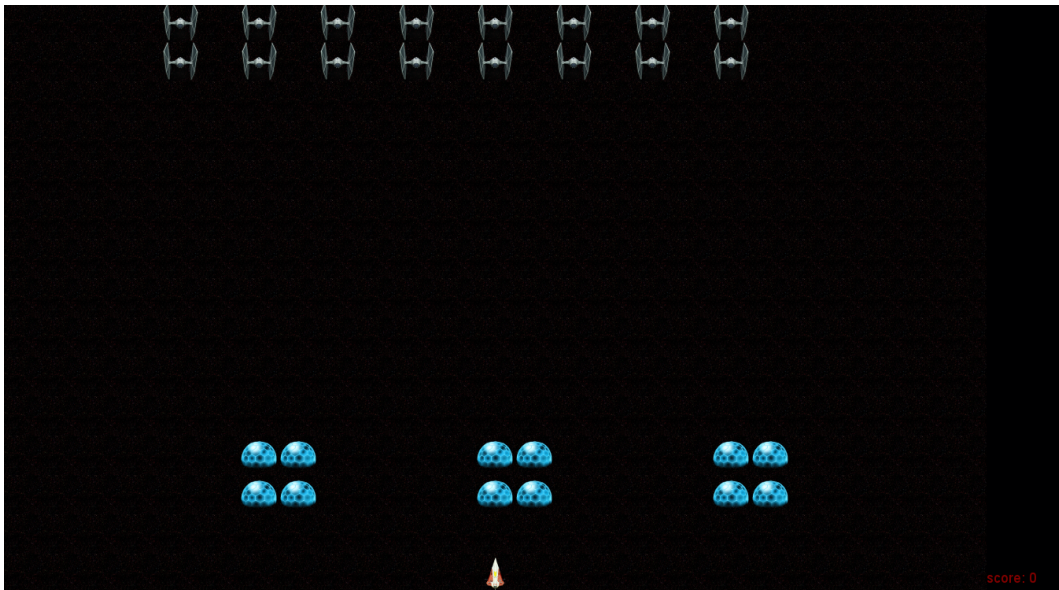
Lors de la conception globale de notre jeu, nous n'avions pas encore eu de cours sur le SDL, il était alors difficile de prendre en compte les procédures et fonctions liées à son utilisation. Malgré les changements effectués du fait de l'utilisation du SDL, l'ossature de notre programme n'a quasiment pas été modifiée : on retrouve dans notre projet final les procédures d'accueil, d'affichage, de gestion du score ou encore de gestion des déplacements des différents éléments du jeu comme dans notre conception globale initiale.

Par ailleurs, étant donné que nous avons terminé la conception de notre jeu avant la date prévue, nous avons décidé de le perfectionner et d'offrir au joueur la possibilité d'évoluer dans l'environnement qu'il souhaite. Ainsi, nous avons ajouté la possibilité de choisir entre trois thèmes de jeu à savoir :

- Un thème « Classic » reprenant les décors du jeu « Space Invaders » dont nous nous sommes inspirés pour réaliser notre programme



- Un thème « Star Wars » directement inspiré de la célèbre saga de science-fiction éponyme que nous avons décidé d'ajouter pour faire un clin d'œil à la sortie de l'épisode 8 de la saga quelques semaines avant le rendu de notre projet



- Un thème « Christmas » que nous avons ajouté à l'approche des fêtes de Noël.



Grâce à l'utilisation du SDL, nous avons également pu rendre notre jeu plus « vivant » et plus agréable à utiliser en y ajoutant des animations, comme l'apparition d'un logo au lancement du programme ou encore le lancement d'une musique d'ambiance. Celle-ci est lancée au début de la partie, après que le joueur ait choisi le thème dans lequel il allait évoluer, et non pas à partir de l'accueil comme ce qui était prévu initialement.

III. Guide d'utilisation du programme

À l'exécution du programme, le classement des 5 meilleurs scores s'affiche sur le terminal. Le joueur peut alors choisir de lancer une partie, d'afficher les règles du jeu ou bien de quitter le programme.

Si le joueur choisit « Play » en appuyant sur la touche **1** du clavier, le joueur peut alors choisir le thème du jeu, la musique est lancée après que ce choix ait été effectué et une fenêtre SDL, dans laquelle apparaît le logo du jeu, s'ouvre. Le jeu démarre après que l'utilisateur ait pressé une touche du clavier, le logo laisse alors place à la scène de jeu.

Le joueur doit alors protéger le monde de ces envahisseurs en déplaçant son vaisseau avec les flèches directionnelles « gauche » et « droite » et en tirant des missiles avec la flèche directionnelle « haut ». Chaque ennemi éliminé par un tir de missile rapporte 100 points, la destruction d'une soucoupe « bonus » rapporte quant à elle 1000 points.

Comme mentionné précédemment, le jeu se termine si le joueur est touché par un missile ennemi ou si un ennemi parvient à atteindre la ligne du joueur. À ce moment-là s'affiche pendant 5 secondes le message « Game Over » suivi du score réalisé par le joueur au cours de la partie. Le joueur est ensuite redirigé vers le menu où, s'il a su s'illustrer parmi les meilleurs pilotes, lui est demandé son pseudo. Le joueur a finalement le choix entre recommencer une partie ou quitter le programme.

IV. Travail du groupe

A. Répartition du travail

Au début de ce projet, nous avons tous les trois participé à l'élaboration du cahier des charges et de l'analyse descendante.

Nous avons ensuite choisi de nous répartir équitablement le travail, en prenant en compte les facilités et les difficultés de chacun, afin de progresser le plus efficacement possible dans la réalisation de notre programme.

Le travail à réaliser a donc été partagé de la manière suivante :

Antoine	Théo	Maxime
<ul style="list-style-type: none">– Déplacement des différents éléments mobiles (aliens, soucoupes, missiles)– Interaction entre les différents éléments du jeu	<ul style="list-style-type: none">– Élaboration du menu– Gestion du score– Gestion du déplacement du vaisseau joueur	<ul style="list-style-type: none">– Affichage des éléments– Conversion du programme pour l'adapter au SDL

Bien sûr, cette répartition du travail n'était pas définitivement arrêtée : chacun pouvait librement travailler sur la partie d'un autre s'il était en avance sur la sienne, ou bien demander de l'aide s'il rencontrait des difficultés. Cela nous a notamment permis de ne jamais rester bloqués sur un problème et ainsi d'avancer plus rapidement que ce qui était prévu.

En plus du travail personnel et des séances de projet prévues dans notre emploi du temps, nous nous sommes réunis chaque semaine afin de faire le point sur l'avancée de notre projet, et échanger sur les modifications à apporter, les difficultés rencontrées, les améliorations possibles, etc.

B. Planning et délais à respecter

15 au 21 septembre	22 au 29 septembre	Du 29 septembre au 26 novembre	Du 26 novembre au 21 décembre	22 décembre
Ecriture du cahier des charges	Rédaction de l'analyse descendante	Codage du programme	Finalisation du programme et rédaction du rapport	Présentation de notre projet final

C. Difficultés rencontrées

La plus grande difficulté que nous ayons rencontrée est la correction des différents bugs à la fin de la réalisation de notre programme. En effet, plus la réalisation du jeu avançait, plus il était difficile de déceler l'origine des différents bugs.

La gestion du déplacement des différents éléments a soulevé quelques difficultés : « Space Invaders » est un jeu dans lequel les différents éléments n'évoluent pas sur le même rythme et le langage Pascal ne permet pas de faire fonctionner deux programmes simultanément, il a donc fallu trouver un compromis pour pallier à ce problème.

D. Perspectives d'évolution du programme

Avec davantage de temps, nous aurions pu proposer de nouvelles fonctionnalités à l'utilisateur comme proposer une gamme de différents vaisseaux déblocables en fonction du score, proposer « l'achat » d'améliorations dans une « boutique » ou encore instaurer un système de points de vie permettant au joueur de continuer la partie même après avoir été touché par un missile ennemi.