
FILIERE INGENIEUR SOUS STATUT ETUDIANT

Année 2023/2024

PRO3600

PROJET INFORMATIQUE

MULTIPLAYER SNAKE GAME

Pré-rapport V2

*Par REIG Tanguy, COUSSON Antoine, DENYS DE BONNAVENTURE
HéloïRse, AMRAOUI Idriss et DELFORGE Raphaël*



IP PARIS

Réalisé le 09/02/2024

Encadrant : LALLET Éric

Table des matières

1.	Contexte et Historique	1
2.	Description de la demande (cahier des charges).....	1
2.1.	Les objectifs	1
2.2.	Déroulement du jeu	1
2.3.	Cahier des charges.....	2
2.4.	Les produits du projet.....	2
2.5.	Les fonctions du produit.....	3
2.6.	Les critères d’acceptabilité et de réception	3
3.	Contraintes.....	3
3.1.	Les contraintes de coûts	3
3.2.	Les contraintes de délais.....	3
3.3.	Les contraintes techniques.....	4
3.4.	Les clauses juridiques, etc.	4
4.	Déroulement du projet – planification des phases et ressources	4
4.1.	Plan de Charge.....	5
5.	Authentification – date et signature du chef de projet et du maître d’ouvrage.....	7
6.	Annexes - lister et joindre au cahier des charges les éventuels documents que le client peut mettre à disposition.....	7
6.1.	Annexe 1 : Lexique / Abréviations / Termes techniques	7
6.2.	Annexe 2 : Bibliographie	7

1. CONTEXTE ET HISTORIQUE

Dans la deuxième moitié du XXème siècle, les jeux vidéo d'arcade sont une source importante de divertissement pour les jeunes. Elle marque un premier pas vers la massification du divertissement numérique. Cependant, leur popularité diminue dans les années 90. En 1998, Nokia intègre Snake, inspiré d'un jeu d'arcade nommé Blockade, et lui redonne ainsi de la popularité. Dans cette version du jeu, un serpent doit manger des cubes, s'allonge et devient un obstacle pour lui-même au fur et à mesure de la partie. Depuis, de nombreuses variations existent, notamment Slither.io, développée en 2016 par Steve Howse qui est une version multijoueur du concept.

Dans ce projet, nous concevrons une version de Snake, en multijoueur dans un réseau local.

2. DESCRIPTION DE LA DEMANDE (CAHIER DES CHARGES)

2.1. Les objectifs

Pour mener à bien notre projet, nous nous focaliserons d'abord sur nos objectifs principaux avant de rajouter des fonctionnalités supplémentaires si le temps nous le permet. Nous identifions 2 objectifs principaux.

Le premier objectif est la réalisation du modèle et la visualisation du celui-ci sur une interface graphique. Autrement dit, avoir un jeu qui fonctionne correctement, c'est-à dire un serpent qui se déplace dans une fenêtre. Ce livrable intermédiaire pourra potentiellement faire l'objet d'une version du jeu final en solo, sous réserve d'y intégrer quelques fonctionnalités supplémentaires.

Notre deuxième objectif est l'implémentation du jeu sur un réseau local pour qu'un deuxième joueur puisse se connecter. Dans ce cas, le serpent de chaque joueur apparaît au centre de la fenêtre de jeu du joueur en question.

Finalement, nous essayerons de faire fonctionner le jeu avec plus de joueurs, et en ajoutant des fonctionnalités pour rendre le jeu plus intéressant.

2.2. Déroulement du jeu

Lors du lancement du jeu, un menu du jeu sera affiché. L'utilisateur aura le choix de jouer en local (donc seul) ou alors en multijoueur. On pourra rajouter des options supplémentaires sur ce menu ultérieurement si possible.

Pour le mode multijoueur, le serpent apparaît aléatoirement sur la carte sauf si le point d'apparition est sur un serpent. La fin du jeu est signée lorsqu'il ne reste qu'un serpent. Les utilisateurs auront à ce moment le choix de relancer une partie ou de retourner sur le menu.

2.3. Cahier des charges

Sur le menu, nous voulons retrouver :

- Un fond d'écran adéquat
- Le titre du jeu avec sa version
- 2 boutons : « Local » et « Multi »
 - Les boutons doivent sur-briller lorsque la souris les survole
 - Les boutons doivent changer lorsqu'on clique dessus
 - Le bouton local envoi directement sur le jeu
 - Le bouton Multi envoi vers un page de connexion

Sur le menu de connexion, :

- Il est scindé en 2 parties : une partie serveur, une autre partie client
- Dans chaque partie on retrouve, (cette partie pourra être adapté en fonction du mode de connexion développé lors du projet. Cette partie est donc sujet à être modifié)
 - Une zone de texte éditable pour renseigner l'adresse IP (qui peut être renseigné automatiquement)
 - Une zone de texte pour renseigner le port utilisé pour la connexion (il faut que ce nombre soit compris entre 1025 et 65535 et soit identique pour tous les utilisateurs)
- Un bouton cliquable exécutable uniquement pour l'utilisateur-serveur afin qu'il puisse lancer la partie

Sur le jeu :

- Les serpents ont un point d'appariation aléatoire à l'exception des serpents déjà présent sur la carte de jeu.
- Les serpents apparaissent avec une taille minimale (fonctionnalité bonus)
- Les serpents se déplace en fonction de la position de la souris par rapport au centre de la fenêtre
- Les serpents grandissent en "mangeant" des billes de couleurs (fonctionnalité bonus)
- Les billes de couleurs apparaissent de manière aléatoire
- En fonction de la couleur, les billes donnent plus ou moins de « nourriture » (fonctionnalité bonus)
- Le jeu s'arrête lorsque le serpent meurt (en solo), ou lorsqu'il ne reste plus qu'un serpent dans la partie (en multijoueur). Un serpent meurt lorsqu'il touche :
 - Le bord de la carte du jeu
 - Un autre serpent

2.4. Les produits du projet

Ce jeu étant un jeu multijoueur, il faut que les joueurs puissent tous se connecter. Nous avons donc choisi un système dans lequel un serveur central gère tout ce qui se passe dans la partie. Les différents joueurs (clients) se connectent à ce serveur. Si c'est l'ordinateur d'un des joueurs

qui joue le rôle du serveur, ce dernier devra tout de même se connecter à son propre serveur comme s'il était un joueur comme un autre (architecture serveur-client avec protocole tcp).

Ce projet sera donc composé de deux produits intégrés dans un seul exécutable : le programme du serveur et le programme des clients.

2.5. Les fonctions du produit

Le but du client est d'envoyer au serveur tout ce que les mouvements que le joueur fait, et de recevoir du serveur les informations qui caractérisent l'état courant du système afin que le client puisse afficher correctement la situation à l'écran.

Le serveur quant à lui, doit s'occuper de tout l'arrière-plan du jeu. C'est ce dernier qui reçoit des différents clients chacune des modifications de l'espace de jeu, qui traite toutes ces informations (déplacement, mort d'un joueur, allongement d'un joueur...) et qui ensuite les renvoie à tous les clients pour que ces derniers restent à jour sur la situation du jeu.

2.6. Les critères d'acceptabilité et de réception

Il faut que l'ensemble des contraintes techniques soit résolu. Pour cela, nous adopterons la méthode agile qui consistera à faire plusieurs itérations du jeu en rendant des livrables intermédiaires avec une fréquence régulière.

3. CONTRAINTES

3.1. Les contraintes de coûts

Pour notre projet de développement du jeu Snake en mode multijoueur local, étant donné qu'il s'agit d'un projet d'école à but non lucratif sans ingérence extérieure, nous pouvons minimiser les coûts en exploitant les ressources disponibles au sein de notre groupe de travail. Cela pourrait impliquer l'utilisation de logiciels open source et gratuits pour le développement et la conception graphique, ainsi que l'exploitation des compétences existantes parmi les membres de l'équipe.

Pour le matériel, nous utiliserons les équipements disponibles dans les installations de l'école et/ou nos propres ordinateurs personnels. En adoptant une approche de planification minutieuse et en exploitant les ressources disponibles localement, on peut réaliser notre projet de manière efficace¹.

3.2. Les contraintes de délais

Les contraintes de délai seront celles imposées par le calendrier prévu dans le cadre du projet PRO36000. Ci-dessous un diagramme de Gaant et un suivi des tâches détaille nos prévisions en termes de répartition et de volume horaire.

3.3. Les contraintes techniques

Nous relevons plusieurs contraintes techniques dans la réalisation de notre projet.

- Coder en Java un serpent qui se déplace en suivant la souris
- Modéliser ce serpent sur une interface graphique avec la bibliothèque JavaFX
- La fenêtre de jeu doit être de taille finie et fixe, le serpent étant confiné à ce cadre
- Si le serpent atteint un des rebords de la fenêtre (mur), le serpent meurt
- Le serpent doit être à tout moment au centre de l'écran d'ordinateur du joueur
- Lorsque la tête d'un serpent entre en collision, soit avec un mur soit avec un autre serpent, le serpent meurt
- Plusieurs joueurs doivent pouvoir se connecter sur un même serveur et jouer ensemble dans la même fenêtre de jeu
- La partie doit se terminer lorsqu'il ne reste qu'un seul serpent vivant sur le serveur
- Lorsqu'un joueur se connecte sur un serveur, son serpent doit avoir une position initiale et une direction initiale aléatoire dans la fenêtre de jeu. (Peut-être excepté les cas où le serpent est trop proche du bord)
- Il faut connaître la position de la tête du serpent en temps réel et sur un certain nombre d'intervalles de temps précédent (de la longueur du serpent) qu'il faut donc garder en mémoire

3.4. Les clauses juridiques, etc.

Pour notre projet de développement informatique, nous devons prendre en compte plusieurs aspects juridiques. Il est crucial de clarifier la propriété intellectuelle du jeu et d'établir des accords de collaboration pour définir les responsabilités individuelles et les attentes en matière de travail en équipe.

Nous devons également respecter les termes de licence des ressources externes utilisées et veiller à la protection des données personnelles des utilisateurs conformément aux lois applicables. Des clauses de non-responsabilité dans les conditions d'utilisation du jeu sont importantes pour limiter notre responsabilité en cas de dommages. Enfin, la rédaction de conditions générales d'utilisation pour le jeu est essentielle pour définir les droits et responsabilités des utilisateurs.

En revanche, il est clair que le projet n'a pas vocation à être publié. C'est pour cela qu'il n'est pas vraiment question de prendre en compte les aspects juridiques susdits.

4. DEROULEMENT DU PROJET – PLANIFICATION DES PHASES ET RESSOURCES

Afin d'assurer le bon déroulement du projet, nous nous organisons autour de deux points centraux.

Le premier c'est un planning sous la forme d'un diagramme de Gantt pour suivre les différentes tâches périodiquement ainsi que leur avancé. Cela nous permet aussi d'être vigilant vis-à-vis du chemin critique afin d'éviter tout retard.

Le deuxième point est le développement du projet autour de la méthode agile et également avec l'utilisation d'une plateforme git qui sera dans notre cas GitHub. Ce dernier point nous permettra de travailler en simultané sur notre projet tout en ayant une trace des évolutions et des changements effectués au cours du développement avec une méthode de travail qui se prête au développement information.

4.1. Plan de Charge

1er jalon :

Partie modèle : création des serpents (liste doublement chaînée), création de méthodes permettant le mouvement du serpent. (Raphaël)

Partie contrôleur : crée les liens entre modèle et interface graphique et test avec la partie modèle : mise en place d'un chatbox. (Idriss et Tanguy)

Partie interface graphique : réalisation grâce aux informations transmises par le contrôleur du serpent qui bouge. (Antoine Héloïse)

2ième jalon :

Partie modèle : création du cadre fini et des conditions de fin de jeu et mort des serpents. (Raphaël et Héloïs)

Partie réseau : envoi et analyse des trames (Antoine)

Partie interface graphique : faire disparaître le serpent. (Idriss et Tanguy)

Chacun s'occupera de ses parties jusqu' à la fin. Une fois la partie modèle finie (surement plus rapide), Raphaël et Héloïse rejoindront Antoine sur la partie contrôleur sur la partie implémentation.

Nous présentons notre diagramme de Gantt prévisionnel effectué en début de projet.

Project Multiplayer Snake game

Premier prototype et régler les problèmes				deuxième prototype, amélioration de l'aperçu du jeu				End of project	
FÉVRIER ET MARS					AVRIL ET MAI				
19/02	26/02	3/03	17/03	1/04 au 14/04	22/04	1/05 au 15/05	22/05		
premier aperçu du jeu statique	le serpent est capable de bouger et disparaître	la partie s'arrête quand il ne reste plus qu'un serpent	billes de couleurs et taille variable du serpent	partie en réseau entre deux joueurs					
Planning	Création de l'objet serpent et des méthodes caractéristiques des actions du serpent (Héloïse)	Serpent peut disparaître selon les règles du jeu	création des bulles de couleur et taille variable du serpent	réduction du livrable	serpent grandit quand il passe sur une bille				
	Réalisation d'une chatbox (Antoine)	envoi et analyse des frames		implémentation					
	représentation du serpent et d'un cadre fini (Idriss et Tanguy)	serpent bouge		serpent disparaît et partie s'arrête		aperçu des billes de couleurs et de la taille variable du serpent			

Commentaire à faire sur la figure une fois réalisé

5. AUTHENTIFICATION – DATE ET SIGNATURE DU CHEF DE PROJET ET DU MAITRE D'OUVRAGE

Cette partie consiste à approuver les faits susdits dans leur réalisation et également à confirmer que les objectifs attendus soient conformes aux attentes du projet informatique PRO3600. Ainsi, les différents points abordés constituent des éléments fondamentaux dans la réalisation du projet mais certains points seront assujettis à des modifications ultérieures en fonction de leur faisabilité, du temps restant, et des attentes du projet informatique.

<u><i>Maître d'ouvrage</i></u> Fait à Le Signature :	<u><i>Chef de projet</i></u> Fait à Le Signature :
---	---

6. ANNEXES - LISTER ET JOINDRE AU CAHIER DES CHARGES LES EVENTUELS DOCUMENTS QUE LE CLIENT PEUT METTRE A DISPOSITION

6.1. Annexe 1 : Lexique / Abréviations / Termes techniques

Git outil collaboratif

6.2. Annexe 2 : Bibliographie

- [1] Nom_du_site. Nom_de_la_page [en ligne]
 Disponible sur : <URL> (consulté le jj/mm/2024)
- [2]