



# Projet SFL

Début du projet : 02/03/2021

Echéance du projet : 10/05/2021

dernière mise à jour : 04/05/2021

**LIFAP4** - 2021 - Semestre 4

---

Université Claude Bernard Lyon 1

p1907037 SIONI Farès

p1909886 BAGNOL Stanislas

p1925126 CHOUGAR Lyes

## Chapitre 1 - Présentation du projet

- **Qui** : SIONI Farès, BAGNOL Stanislas et CHOUGAR Lyes
- **Contexte** : Suivent l'UE LIFAP4. Doivent réaliser un projet de groupe.

## Chapitre 2 - Description de la demande

- **Résultats visés** : Programmation d'un jeu 2D type Zelda Like dans un monde libre avec la possibilité d'importer une carte réalisée avec le logiciel Tiled. Dans ce jeu, le personnage que l'on incarne arrive dans un monde qui regorge d'ennemis équipés différemment. Nous devons combattre ces derniers et explorer la carte pour accomplir des mini-quêtes.

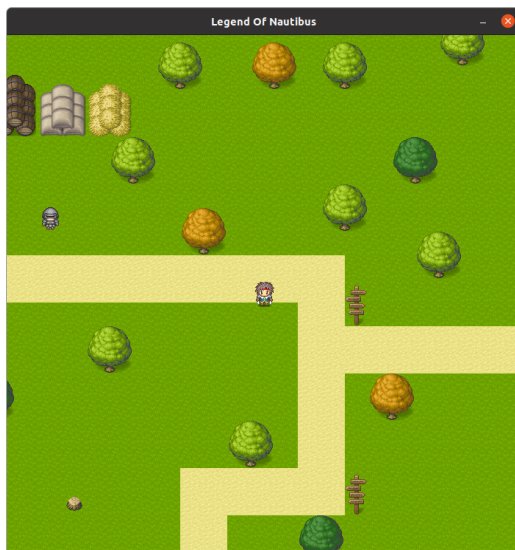


Figure numéro 1 : Aperçu du jeu

Le jeu sous version graphique aura l'aspect d'un jeu 2D vu du dessus reprenant des graphismes type "pixel art" haute résolution. La caméra suit le joueur afin de le garder au centre de l'écran. Le joueur peut se déplacer librement au sein de.s carte.s, interagir avec les personnages non joueurs, les ennemis...

- **Fonctionnalités** :
  - Le but est d'accomplir les quêtes données mais aussi de découvrir l'environnement dans lequel le joueur évolue.



- Présence de PNJ (personnages non-jouables) à certains endroits de la carte permettant des interactions avec le joueur (augmentation du nombre de points de vie restants, dialogues etc..)
- Possibilité d'améliorer ses compétences en réussissant les mini-quêtes (amélioration des armes, augmentation de la vitesse de déplacement ...)
- Possibilité d'infliger des dégâts et d'en subir selon les armes utilisées et les ennemis combattus
- Possibilité d'importer des cartes au format .tmx (Tilemap XML), créées avec le logiciel Tiled par exemple.
- Mode graphique à l'aide du framework SFML.
- Utilisation d'un système de 'State Manager' qui permet d'avoir plusieurs scènes (menu principale, jeu...) ayant chacune leurs propriétés d'initialisation, de boucle d'exécution, pause, fin...
- Présence d'ennemis avec qui le joueur peut combattre.
- Gestion des déplacements et action des entités (PNJ et ennemies) avec un Finite State Machine (automate fini).

## Chapitre 3 - Contraintes

- Durée de développement : 10 semaines (du 02/03/2021 au 10/05/2021)
- Utilisation du langage C++ avec POO
- Application de la méthode AGILE
- Mise en place d'un [diagramme des classes](#) (UML)
- Planification du temps à l'aide d'un [diagramme de Gantt](#)
- Utilisation d'un [gestionnaire de version](#)
- Documentation des headers à l'aide de Doxygen
- Modularité de l'interface. Passage aisé d'un mode texte à un mode graphique
- Utilisation d'une bibliothèque graphique (ici SFML)
- Qualité de programmation
- Compilation à l'aide d'un Makefile
- Organisation des fichiers du projet claire et compréhensible

- Garder les documents à jour (UML, Gantt, cahier des charges, documentation, README)
- Rester au maximum en adéquation avec le cahier des charges
- Présentation du projet complet lors d'une soutenance

## Chapitre 4 - Déroulement du projet

### Liste des tâches

#### Tâche 1 (temps estimé : 3 semaines) - Lyes

- Réalisation du joueur
- Livrable -> Module Joueur
- Réalisé quand les principes de bases du joueur sont fonctionnels (mouvements, collision, vie). Fonction et exécutable fonctionnels pour tests de régression.

#### Tâche 2 (temps estimé : 3 semaines) - Farès et Stan

- Réalisation de l'architecture de la carte du niveau
- Livrable -> Module Carte
- Réalisé quand la structure principale composant une carte est fonctionnelle (plusieurs couches : textures de bases -sol-, textures additionnelles -arbres, structures, coffres...-, collision, spawn du joueur/ennemi. Récupération des données depuis un fichier .tmx -> analyse du fichier et stockage dans des classes...). Fonction et exécutable fonctionnels pour tests de régression.

#### Tâche 3 (temps estimé : 1 semaine) - Stan

- Initialisation de l'affichage graphique
- Livrable -> Module AffichageSFML avec inclusion des librairies
- Réalisé quand une fenêtre s'ouvre correctement avec SFML. Fonction et exécutable fonctionnels pour tests de régression.

#### Tâche 4 (temps estimé : 2 semaine) - Farès

- Initialisation de l'affichage en mode texte
- Livrable -> Module AffichageTexte
- Réalisé quand une fenêtre de terminal permet un affichage simplifié du jeu. Fonction et exécutable fonctionnels pour tests de régression.

### Tâche 5 (temps estimé : 2 semaines) - Lyes

- Réalisation des ennemies
- Livrable -> Module Ennemis
- Réalisé quand les principes de bases d'un ennemi sont fonctionnels (mouvements, collision, vie). Fonction et exécutable fonctionnels pour tests de régression.

### Tâche 6 (temps estimé : 3 semaines) - Stan

- Réalisation des affichages .
- Livrable -> Gestion de l'affichage des assets (textures, sons, polices..), complétion du Module AffichageGraphique et AffichageTexte.
- Réalisé quand le personnage, les ennemis, etc. sont affichés sur un monde, graphique et texte. Fonction et exécutable fonctionnels pour tests de régression.

### Tâche 7 (temps estimé : 2 semaines) - Farès

- Réalisation du système de collision.
- Livrable -> Gestion des collisions entre le joueur et son environnement
- Réalisé quand le joueur peut se déplacer en pouvant heurter d'autres éléments, qu'un mode debug permette de voir les boîtes de collisions en mode graphique et qu'elles soit affiché en mode texte.

### Tâche 8(temps estimé : 2 semaines) - Farès

- Réalisation du système de gestion de scène.
- Livrable -> Gestion des différents états du programme (en jeu, menu principal, menu pause, splashscreen...).
- Réalisé quand les scènes peuvent être jouées, passées de l'une à l'autre, être en pause, etc. sans problème. Fonction et exécutable fonctionnels pour tests de régression.

### Tâche 9 (temps estimé : 2 semaines) - Lyes

- Réalisation des personnages non-joueurs (PNJ).
- Livrable -> Module PNJ.
- Réalisé quand les PNJ sont fonctionnels (réalisent ce qu'on leur demande : rester immobile, régénérer la vie du joueur ou donner un indice sur la quête...). Fonction et exécutable fonctionnels pour tests de régression.

### Tâche 10 (temps estimé : 1 semaines) - Farès

- Réalisation des menus.
- Livrable -> Les différents menus (principal, pause, paramètres...).
- Réalisé quand les différents menus sont fonctionnels en utilisant le système de gestion de scène. Fonction et exécutable fonctionnels pour tests de régression.

### Tâche 11 (temps estimé : 2 semaines) - Stan

- Réalisation du système de combat avec le Finite State Machine
- Livrable -> Module FiniteStateMachine et fonction d'attaque du joueur.
- Réalisé quand le joueur peut infliger des dégâts aux ennemis et en recevoir. Fonction et exécutable fonctionnels pour tests de régression.

### Tâche 12 (temps estimé : 2 semaines) - Stan, Farès et Lyes

- Finition de tous les tests de régression
- Livrable -> Tous les tests de régression.
- Réalisé quand tous les tests de régression permettent de vérifier complètement l'intégrité des classes et fonctions.

### Tâche 13 (temps estimé : 3 semaines) - Lyes, Farès et Stan

- Réalisation de l'histoire.
- Livrable -> Le scénario et son implémentation (carte, histoire, objectifs...).
- Réalisé quand l'histoire se déroule correctement. Fonction et exécutable fonctionnels pour tests de régression.

### Tâche 14 (temps estimé : 3 semaines) - Lyes, Farès et Stan

- Finalisation du projet
- Livrable -> Le jeu complètement jouable.
- Réalisé quand le jeu est sans bugs apparents, s'exécute et est facile à prendre en main, que tous les modules, le cahier des charges, le diagramme des classes, les exécutables, le dépôt git ont été passés en revue et les tâches vérifiées.

## Chapitre 5 - Diagramme de Gantt

Tâche\Semaine	1	2	3	4	5	6	7	8	9	10
1 : Réalisation du Joueur	L	L	L							
2 : Réalisation de l'architecture de la carte du niveau	S&F	S&F	S&F							
3 : Initialisation de l'affichage en mode graphique		S								
4 : Initialisation de l'affichage en mode texte			F	F						
5 : Réalisation des ennemies				L	L					
6 : Réalisation des affichages				S	S	S				
7 : Réalisation du système de collision				F	F					
8 : Réalisation du système de gestion de scène					F	F				
9 : Réalisation des personnages non-joueurs (PNJ)						L	L			
10 : Réalisation des menus							F			
11 : Réalisation du système de combat avec FSM							S	S	S	
12 : Finitions de tous les test de régressions								SFL	SFL	
13 : Réalisation de l'histoire								F	SFL	SFL
14 : Finalisation du projet								F	F	SFL

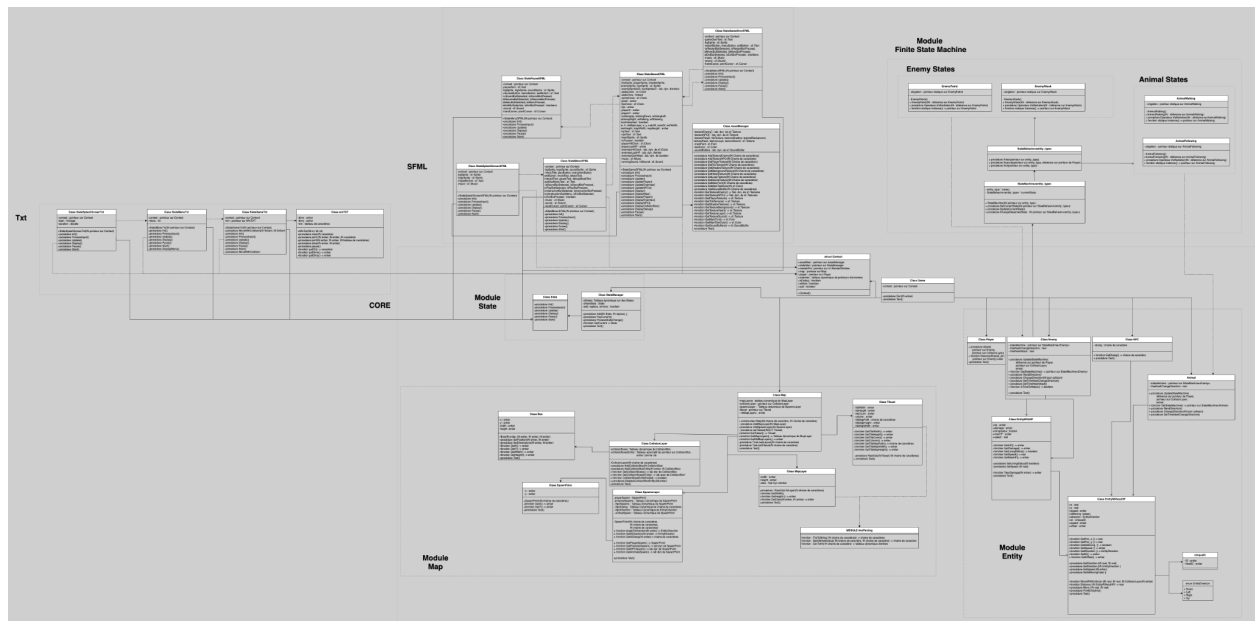
Les lettres indiquent lesquelles est la personne chargée à la tâche de la ligne correspondante

S : Stanislas

F : Farès

L : Lyes

## Chapitre 6 - Diagramme des classes



[Diagramme des classes](#)

## Chapitre 7 - Gestionnaire de versions

[Gestionnaire de versions](#)

## Chapitre 8 - Tableau Trello

[Tableau Trello](#)

## Chapitre 9 - Documentation

[Documentation](#)