

Projet Logiciel Transversal

Khames Abdelmajid

Dia Mamadou

Le Guen de Kerneizon Paul

Chelki Adam

ENSEA 2021-2022

Table des matières

1	Présentation Générale	1
1.1	Archétype	1
1.2	Règles du jeu	1
1.2.1	Mécanique de Pierre-Feuilles-Ciseaux	1
1.2.2	Personnages	2
1.2.3	Déroulement d'une partie en solo	2
1.2.4	Déroulement d'une partie multijoueur	3
1.3	Ressources	3
2	Description et conception des états	5
2.1	Description des états	5
2.2	Conception Logiciel	5
3	Rendu : Stratégie et Conception	6
3.1	Stratégie de rendu d'un état	6
3.2	Conception logiciel	6
4	Règles de changement d'états et moteur de jeu	7
4.1	Règles	7
4.2	Conception logiciel	8
5	Intelligence Artificielle	8
5.1	Stratégies	8
5.2	Conception logiciel	9
6	Modularisation	9
6.1	Organisation des modules	9
6.2	Conception logiciel	10

1 Présentation Générale

1.1 Archétype

Dans ce projet, nous réaliserons un jeu au tour par tour en C++. Nous avons donc décidé de nous inspirer des mécaniques du jeu mobile **Raid : Shadow Legends**, c'est-à-dire un jeu se basant exclusivement sur l'affrontement entre divers champions ou des boss, qu'ils soient contrôlés par l'IA ou par un autre joueur.



Illustration du système de combat du jeu **Raid : Shadow Legends**

La thématique du jeu aura lieu dans l'ENSEA, où le joueur incarnera des personnages représentant les Options disponibles en 3A, les attaques, statuts ainsi que les objets feront référence à des thématiques étudiées à l'ENSEA ou à des éléments de la vie étudiante. Réaliser un jeu ayant lieu dans l'ENSEA nous permettra d'ajouter une touche d'originalité à un système déjà bien connu, et d'apporter notre propre empreinte aux ressources visuelles.

Le jeu inclura une sélection de personnages avec des statistiques et des archétypes (tank, soutien, mage,...) ainsi qu'un élément appelé **Majeur** suivant une mécanique de Pierre-Feuille-Ciseaux.

1.2 Règles du jeu

1.2.1 Mécanique de Pierre-Feuilles-Ciseaux

Chaque personnage du notre jeu-vidéo est défini par une **Majeur** parmi une liste de 5 Majeurs :

- Majeur Elec
- Majeur Info
- Majeur Auto
- Majeur Signal
- Majeur Sciences-Humaines

Voici le tableau montrant les forces et faiblesses de chaque Majeur, chaque attaque sera amélioré ou réduite en fonction de la **Majeur** de la cible :

Attaquant / Cible	Elec	Info	Auto	Signal	Sciences-Humaines
Elec	×1	×1	×0.5	×2	×1
Info	×1	×1	×2	×0.5	×1
Auto	×2	×0.5	×1	×1	×1
Signal	×0.5	×2	×1	×1	×1
Sciences-Humaines	×1	×1	×1	×1	×2

1.2.2 Personnages

Avant chaque partie, le joueur devra choisir quatre champions parmi une sélection prédéfinis de 9 champions listés ci-dessous :

- Champion **IS** : Majeur Info
- Champion **SIA** : Majeur Signal
- Champion **RT** : Majeur Signal
- Champion **AEI** : Majeur Auto
- Champion **MSC** : Majeur Auto
- Champion **ESE** : Majeur Elec
- Champion **EVE** : Majeur Elec
- Champion **ESC** : Majeur Elec
- Champion **Audencia** : Majeur Sciences-Humaines

A la fin de la partie, le joueur sera amené à affronter un *boss final*, ce boss ne possède pas de **Majeur** fixe, en effet il possédera une ou plusieurs attaques pour chaque **Majeur** qui, une fois lancée, lui attribuera la **Majeur** correspondante. L'équipe du joueur devra donc être polyvalente pour pouvoir affronter le boss.

1.2.3 Déroulement d'une partie en solo

Une partie est composée de quatre arènes que l'on nommera *Arène 1*, *Arène 2*, *Arène 3* et *Arène 4*. Au début de la partie, le joueur choisit le champion avec lequel il veut commencer.

1. Dans l'*Arène 1*, le joueur affrontera successivement trois champions parmi les huit restants, une fois l'*Arène 1* complétée les trois champions rejoignent l'équipe du joueur.
2. L'*Arène 2* est elle aussi composée de trois combats successifs face à trois champions sur les cinq restants, mais une fois l'*Arène 2* complétée, le joueur pourra remplacer l'un de ses quatre champions parmi l'un des trois champions vaincus dans l'*Arène 2*.
3. L'*Arène 3* quand à elle est composée des deux derniers champions et le joueur pourra de même, remplacer un de ses quatre champions parmi les 2 vaincus.
4. Une fois les 3 premières *Arène* complétées, le joueur va désormais affronter le *boss final* dans l'*Arène 4* avec son équipe de 4 champions.

La suite de champions que le joueur va affronter est générée en début de partie. Elle peut être purement aléatoire, mais être aussi calculée intelligemment : par exemple, si le joueur choisit en début de partie un champion IS, le jeu peut décider de lui faire affronter en début de partie des champions avec des vulnérabilités de la majeure Info pour que le début de partie soit plus facile.

Entre chaque combat, des événements aléatoires peuvent se produire, comme l'apparition d'un coffre dont la condition pour l'ouvrir est de répondre à un QCM portant sur une majeure. Une bonne réponse donne un avantage au joueur (statistiques du joueur améliorées, objets, etc) mais à l'inverse une mauvaise réponse donne un malus au joueur (statistiques du joueur diminuées, etc).

Enfin, il est à noter que le jeu fonctionne sur la base d'une partie unique : si le personnage du joueur meurt, la partie recommence depuis le début.

Voici un schéma explicatif montrant le déroulé d'une partie :

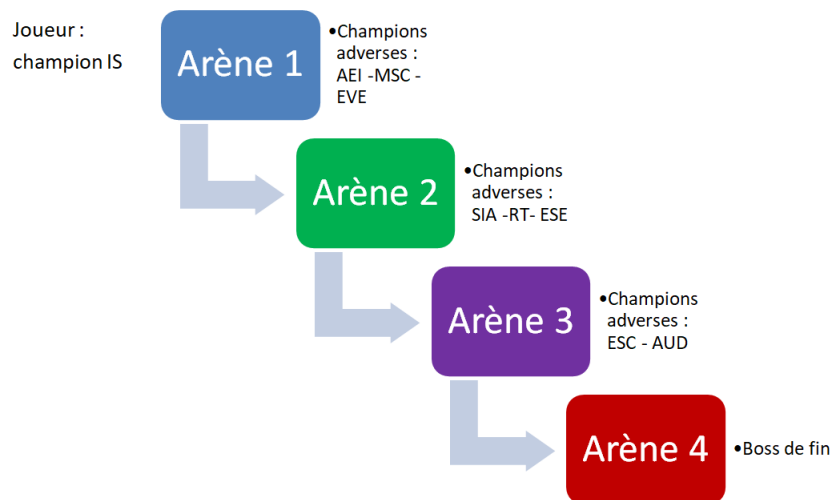


Schéma d'une partie type

1.2.4 Déroulement d'une partie multijoueur

Chaque joueur choisira 4 champions parmi les 9 proposés, ils s'affronteront ensuite directement dans un combat en 4v4.

1.3 Ressources

L'idée générale du rendu est d'avoir un aspect "rétro" 16 bits, dans le même genre que les RPG du début des années 1990 (Final Fantasy 5, Dragon Quest...). Pour cela, nous allons utiliser des ressources sous licences libres, dont la seule obligation est de mentionner les auteurs en crédits. La thématique du jeu étant centrée sur l'ENSEA, et le jeu prenant part dans ses locaux, on utilisera des sprites ayant une consonance moderne. De même, pour que nous ayons des décors fidèles à l'école, nous créerons des sprites spécifiques à l'école (logo...) en utilisant un filtre 16 bits, donnant ainsi aisément à une image un aspect "rétro".

L'interface utilisateur (IU) sera quant à elle rendue par la librairie graphique, et sera uniquement interfacée au moyens de la souris.



A gauche : Un exemple de l'aspect du rendu du jeu (Final Fantasy 5)

A droite : Une image de la rue filtré en 16 bits, qui pourrait être un décor d'arène

Pour le rendu des sprites du joueur et de ses coéquipiers, les sprites de personnage suivant seront utilisés (issus de la collection de sprites *Modern Interior* du créateur LimeZu)



Sprite du joueur et de ses coéquipiers

2 Description et conception des états

2.1 Description des états

2.2 Conception Logiciel

3 Rendu : Stratégie et Conception

3.1 Stratégie de rendu d'un état

3.2 Conception logiciel

4 Règles de changement d'états et moteur de jeu

4.1 Règles

4.2 Conception logiciel

5 Intelligence Artificielle

5.1 Stratégies

5.2 Conception logiciel

6 Modularisation

6.1 Organisation des modules

6.2 Conception logiciel