

Highway to L2

Licence Informatique 2ème année

Remy GEORGET - Remy.Georget.Etu@univ-lemans.fr

Mathéo GALLAIS - Mathéo.Gallais.Etu@univ-lemans.fr

Nathan GAUTERON - Nathan.Gauteron.Etu@univ-lemans.fr

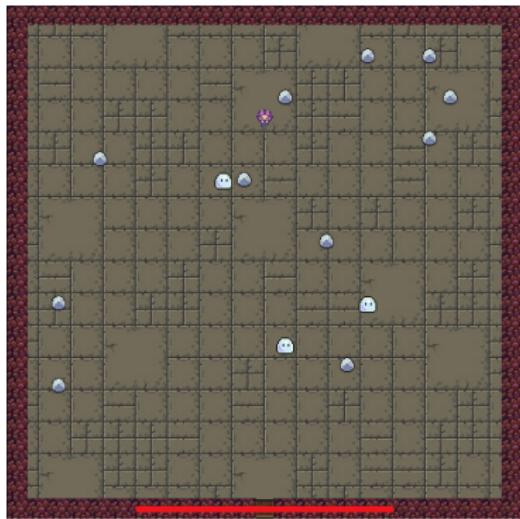
Théo DULUARD - Theo.Duluard.Etu@univ-lemans.fr



12 avril 2023

Définition du concept

- Jeu d'action-aventure en 2D
- Parcours de différents niveaux
- Combat contre des ennemis et des boss
- Collecte d'objets et amélioration des compétences
- Accessible à tous les niveaux de joueurs
- Commandes simples et intuitives
- Possibilité de sauvegarde



Organisation du travail

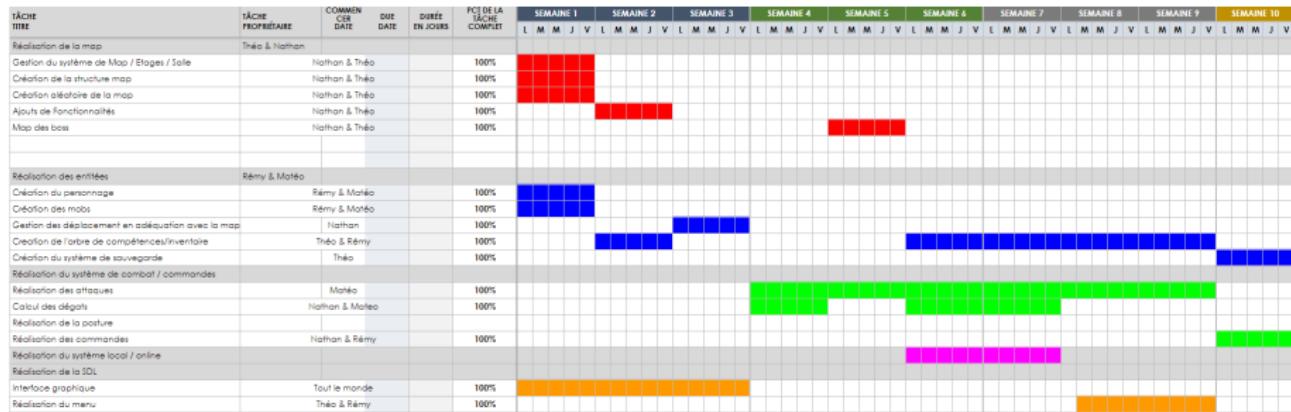


Diagramme de Gantt utilisé pour le projet

Conception du niveau

Jeu composé d'un seul niveau, composé de :

- trois étages, incluant chacun :
 - huit à douze salles
 - une salle de boss
 - une salle de marchand

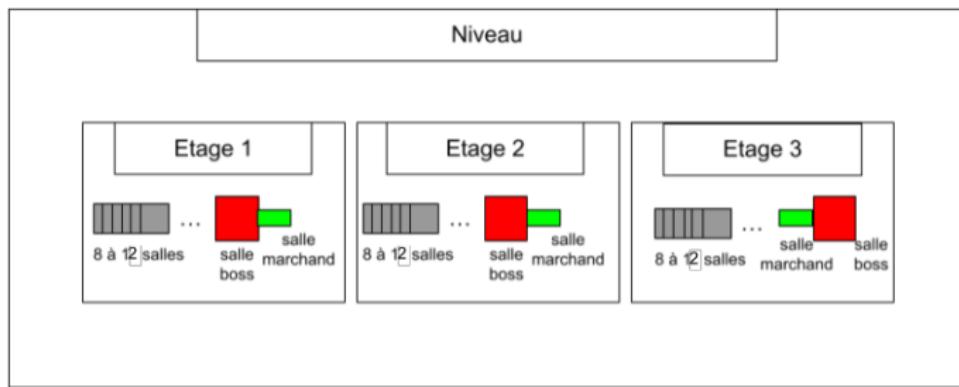
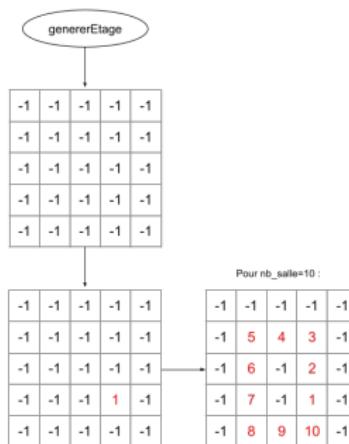
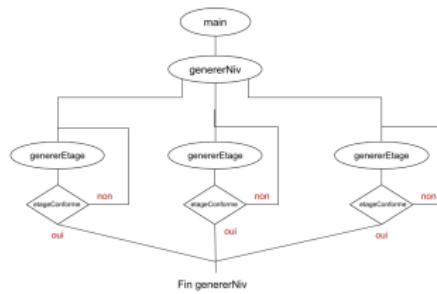


Schéma d'organisation du niveau

Génération du niveau et des étages

- Etage : Matrice 5x5
- Nombre de salles choisi aléatoirement entre huit et douze.
- Salles placées une par une dans la matrice.
- Placement de la salle de boss.
- Lien entre salle de boss et salle du marchand



Cas particuliers :

-1	-1	-1	6	7
-1	-1	-1	5	4
-1	10	-1	-1	3
-1	9	8	1	2
-1	-1	-1	-1	-1

-1	-1	-1	-1	-1
-1	5	4	3	-1
-1	6	9	2	-1
-1	7	8	1	10
-1	-1	-1	-1	-1

Génération des salles

- Matrice 25x25
- Composé d'éléments divers : murs, porte, obstacles, monstres, personnage.
- Nombre d'obstacle choisi aléatoirement entre trois et douze.
- Nombre de monstres choisi aléatoirement entre un et cinq.

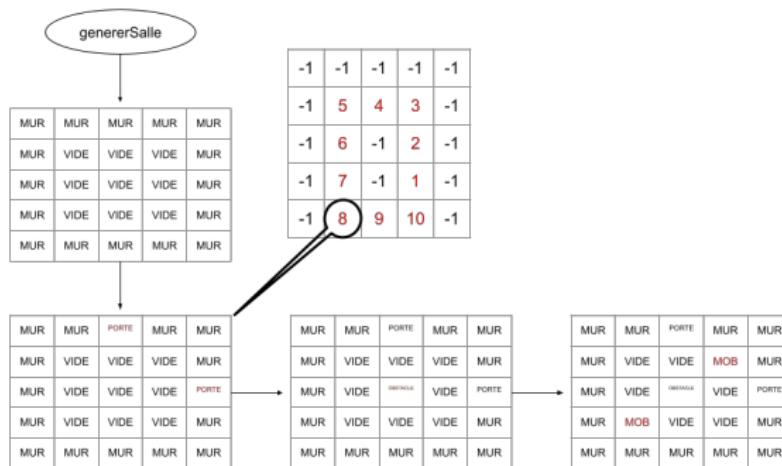


Schéma des fonctions de génération de salle

Conception des entités

- Nous avons trois types d'entités :
 - > Personnage
 - > Mobs (Montres)
 - > Boss
- Chaque entité possède des caractéristiques qui lui sont propres
- Certains possèdent ou pas ces caractéristiques
- Nous avons deux tableaux qui référence nos mobs et boss en dur dans le code du jeu

Conception des Objets et de l'Inventaire

- Un objet est constitué d'un nom et d'une description
- Nous avons un tableau d'objets afin de les avoir en dur dans le code du jeu
- Un inventaire est composé d'un tableau d'entiers et un tableau d'objets

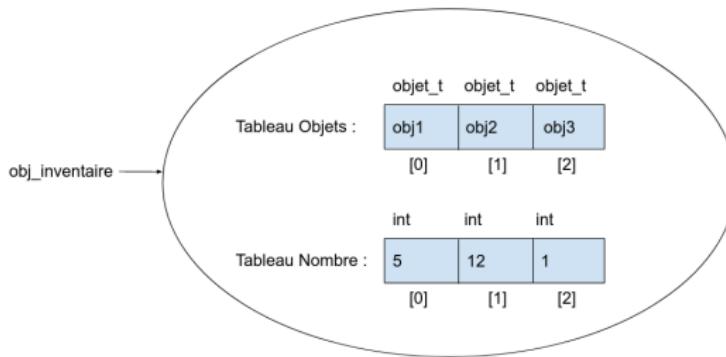


Schéma de fonctionnement d'un inventaire

```

typedef struct competence{
    t_acquis competence_acquise;
    char nom[T_NOM];
    char desc[T_DESC];
    t_buff buff;
    struct competence **precedentes;
    int nb_prec;
    struct competence **suivantes;
    int nb_suiv;
    char *name_img;
    objet_inv_t *obj_necessaires;
    int taille_tab_obj_nec;
    int etage;
} t_competence;

```

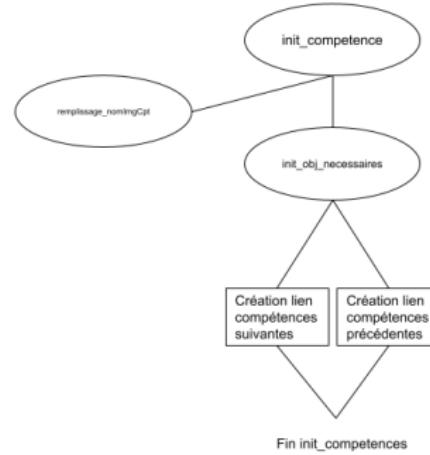


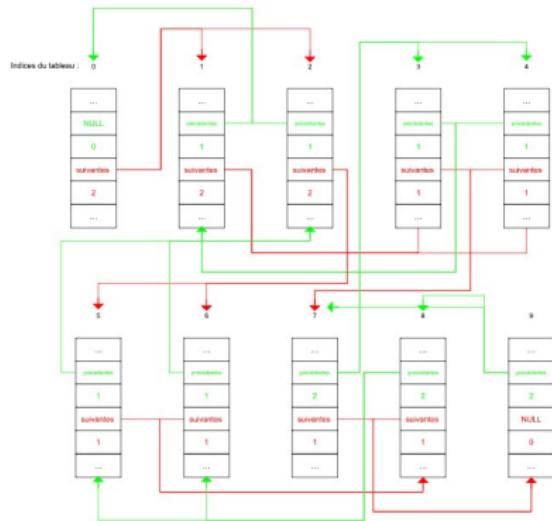
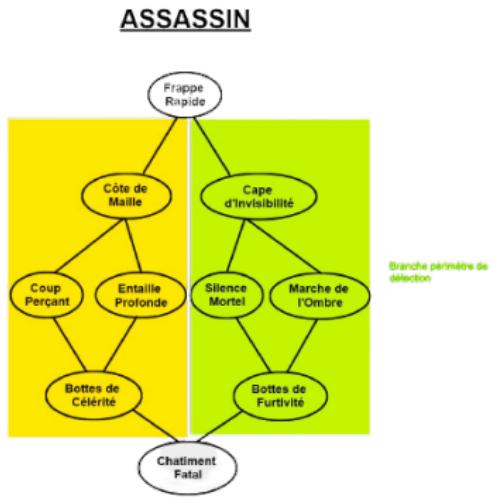
Schéma d'initialisation des compétences

Structure des compétences

Conception des arbres

Struture de l'arbre :

- Type de classe (Archer, Guerrier, Mage ou Assassin).
- Tableau de pointeurs sur compétences.
- Tableau d'acquisition de compétence par étage.



Images de l'arbre assassin et des liens entre les compétences

Système d'attaques

Comment sont gérés les attaques :

```
void (*fond_attaque)(proj_t, entite_t*, t_salle*)
```

attaque_cac

Assassin, Guerriers



attaque_proj

Archer, Mage



Systeme d'attaques

Corps à corps :



Corps à corps

Systeme d'attaques

Un projectile est composé de :

- Vitesse de déplacement
- Portée
- degats
- Touché
- Coordonnées en x précédentes
- Coordonnées en y précédentes
- Coordonnées en x actuelles
- Coordonnées en y actuelles
- Direction du projectile

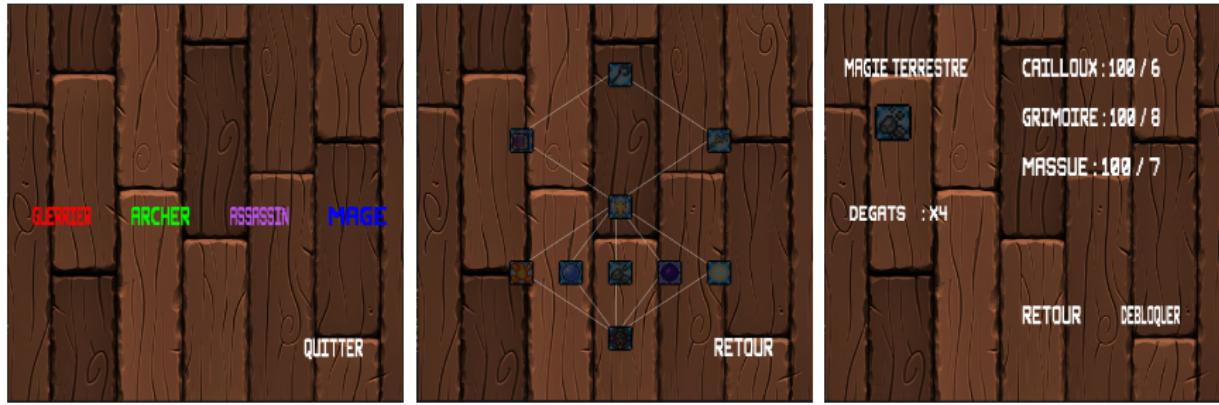
Système d'attaques

Projectiles :



Fonctionnement des projectiles

Menus



Exemples du menu de l'arbre de compétences et de l'inventaire



Menus



Exemples des différentes interfaces de menu



Conclusion

- Nous avons réussi à développer un jeu d'action-aventure en 2D.
- Des améliorations restent possibles.
- Nous avons travaillé de manière efficace en équipe.
- Amélioration possibles : multi-joueur, déplacement plus fluide, système de posture, système de dash.