

Diagramme de Gantt

Liste de tache

Tache 0 : rédiger le cahier des charges

Tache 1 : définir le diagramme des classes

Tache 2 : développement d'un premier prototype en mode texte

Tache 2.1 : écriture et test du module Pos (Pos.h/Pos.cpp)

Défini par deux nombres réels représentant un point en 2D.

Fonction membre capable d'ajouter deux vecteurs, de multiplier un vecteur par un entier...

Tache 2.2 : écriture et test du module Joueur (Joueur.h/Joueur.cpp)

Le joueur est défini par une position, une largeur, une hauteur, un score, un temps d'invincibilité et une direction. Il possède également, une épée et une des armes.

Tache 2.3 : écriture et test du module Monstre (Monstre.h/Monstre.cpp)

Le monstre est défini par une position, une vitesse, une largeur, une hauteur, une direction et un état de vie.

Il peut se déplacer aléatoirement dans une pièce.

Tache 2.4 : écriture et test du module Pièce (Piece.h/Piece.cpp)

Ce module permettra de coder différents types de pièce. Dans chacune d'entre elle, les défis ne seront pas les mêmes. Donc pour chaque type de pièces, les entités ne seront pas les mêmes.

Le joueur pourra naviguer de pièce en pièce par des portes. On récupère la position du joueur et celle des portes donc pour savoir s'il fait ou non changer de pièce.

Tache 2.5 : écriture et test du module Porte (Porte.h/Porte.cpp)

Une porte est définie par une position, une hauteur, une largeur, et une information concernant le fait si elle existe ou non.

Tache 2.6 : écriture et test du module Mur (Mur.h/Mur.cpp)

Un mur est défini par une position, une hauteur, une largeur.

Tache 2.7 : écriture et test du module Obstacle (Obstacle.h/Obstacle.cpp)

Un obstacle représente dans le jeu une barre avec un centre et une extrémité. L'implémentation de rotation y est faite également.

Tache 2.8 : écriture et test du module Arme (Arme.h/Arme.cpp)

Une arme est définie par une position, une vitesse, une force, une hauteur, une longueur, une masse, une direction, et une information permettant de définir l'état de l'arme.

Tache 2.9 : écriture et test du module Plateau (Plateau.h/Plateau.cpp)

Le plateau prend en données membre deux entier pour la dimension du celui-ci. Il consiste en un tableau 1D de pièces mais représenté en 2D.

Dans ce module, on pourra initialiser les pièces (leur donner un type).

Tache 2.10 : écriture et test du module Jeu (Jeu.h/Jeu.cpp)

Ce module permet gérer le jeu en appelant toutes les fonctionnalités du jeu mais ne permet pas de faire de l'affichage. Il permet donc l'initialisation du jeu. Ainsi, on récupère le plateau, le joueur et la pièce où se trouve actuellement le joueur afin de n'afficher uniquement cette pièce. Les actions clavier et les actions automatiques, tels que le déplacement des monstres, du joueur, de la barre en rotation sera implanté dans ce module. La gestion du changement de pièce en fonction de la porte prise par le joueur est également traité ici. De plus, les autres fonctionnalités du jeu est aussi traité, tels que la mort des monstres si le joueur lance une arme, ou bien brandit son épée.

Tache 2.11 : écriture et test du module JeuModeTexte (JeuModeTexte.h/.cpp)

Récupère les données de Jeu et les affiche au format texte. Pour cela on crée un tableau 2D de caractère avec des fonctions membres permettant d'afficher une fenêtre, de dessiner en fonction de la boucle du jeu.

Tache 2.12 : mise en commun des modules et tests.

Tache 3 : développement d'un deuxième prototype en mode SDL2

Tache 3.1 : écriture et test du module Image (Image.h/Image.cpp)

Ce module a pour but de charger une image depuis un fichier vers SDL2. On utilisera des fonctions comme chargerImage pour récupérer une image et dessiner une image pour l'afficher sur une fenêtre SDL2.

Tache 3.2 : écriture et test du module JeuModeGraphique (.h/.cpp)

Ce module permettra l'affichage du jeu dans une fenêtre SDL2 qu'on aura initialisé. Une boucle d'évènement permettra d'update le jeu en fonction des actions du joueur.

Tache 3.3 : Test et debug

	27/01	06/03	13/03	20/03	27/03	03/04	10/04	17/04	24/04
0	L, C								
1	L, C								
2.1		C							
2.2		I	L						
2.3		L							
2.4			C	C	C, L	C, L			
2.5			L						
2.6				C					
2.7					C				
2.8								L	L
2.9				C	C				
2.10					L, C	L, C	L, C	L, C	L, C
2.11			C	C	C				
2.12					L, C	L, C			
3.1			L						
3.2				L	L	L, C	L, C	L, C	L, C
3.3					L, C	L, C	L, C	L, C	L, C

Cyril : C, Linda : L, Ismael : I.