Fiche de sous-traitance

# Description

Nous avons choisi de faire sous-traiter le système de pièces.

Pour rappel les pièces dans SUPER MARIO BROS fonctionne de la façon suivante : Mario ramasse des pièces au cours des différents niveaux lorsqu’il a collecté 100 pièces il gagne un UP ce qui correspond à un essai supplémentaire après une mort avant le GAME OVER.

Le but des sous-traitants sera donc de mettre en place tout le système des pièces (calculs, affichage) afin que notre équipe de développement puisse l’intégrer à notre code.

# Exigences concernées

PIE0 : Les pièces sont récupérées dans les niveaux ou en frappant des *bricks*

PIE1 : Lorsque le joueur collecte 100 pièces il gagne 1 UP et son compteur de pièce est remis à zéro

GP1 : Gestion des collisions avec les autres entités (bonus, ennemis, pièces)

# Précisions

-les pièces doivent pouvoir être disposées à des endroits précis par les level designer de notre équipe.

- les sprites de pièces seront fournis.s

-les sous-traitants implémentent les fonctions selon les règles données par notre équipe.

-les pièces devront tourner sur elles-mêmes.

-la pièce n’affecte pas les ennemis et inversement (les ennemis passent au travers)

-une fonction qui affichera la pièce animée à l’endroit voulu par le développeur. La pièce doit rester sur sa case.

void affichagePiece(piece model, int x, int y, camera cam);

-une structure **piece**qui contiendra toutes les informations de traitement de la pièce. Hitbox rectangulaire (la plus précise possible, correspond au maximum à la forme de la pièce). Chemin d’accés au fichier de sprites de pièces.

typedef struct Piece

{

    int width;

    int height;

    int rotation;

    char \*path;

    ALLEGRO\_BITMAP \*image;

} piece;

-une function **initPiece** qui initialise les données de la piece qui servira de model

void initPiece(piece \*model);

-une fonction **ramassePiece**qui ajoute dans les stats (structure précédemment créée par notre équipe) 1 UP toutes les 100 pièces ramassées et remets à 0 le compteur de pièces.

void ramassePiece(playerStats \*player);

-une fonction **touchePieces** qui enlève les pièces de la map et appelle la fonction ramassePieces pour ajouter cette fonctions dans les stats.

void touchePiece(playerStats \*player, int \*\*map, int x, int y);

# Conseils

Les animations des pièces peuvent être synchronisées pour faciliter le travail (être au même état de rotation au même moment).

La bibliothèque graphique utilisée est Allegro5 nous aiderons les sous-traitants à setup les espaces de travail ainsi que l’accès à notre Github.

DOCUMENTATION ALLEGRO 5 : <https://allegro.developpez.com/5/livre/>