



RAPPORT DE PROJET

REALISE PAR

ANTON RAVEENDRAN
Christian
BONNOT Théo

POUR

Projet commun
d'algorithmique et d'image
IMAC1 - Année 2023-2024

Présentation, guide et lore

INTRODUCTION

Bienvenue dans l'univers de **I Want to Talk to Your Manager**, un jeu de tower defense où les batailles ne se déroulent pas sur les champs de guerre, mais dans les allées animées d'un fast-food. Ici, votre mission n'est pas de repousser des envahisseurs ou des monstres, mais de satisfaire les appétits voraces et les exigences des clients affamés avant qu'ils ne plaignent au manager. Échouez à cette tâche, et vous serez viré !

Concept du Jeu

Dans **I Want to Talk to Your Manager**, vous avez à votre disposition une équipe de chefs de cuisine dans un fast-food où les clients affluent. Vous pouvez compter sur trois types de chefs avec des compétences culinaires spécifiques :

Le Pizzaiolo : Expert en pizzas, il peut préparer et servir une pizza toutes les deux secondes, offrant un pouvoir de rassasiement moyen.

Le Colonel Sonders : Maître des poulets frits, il est capable de servir des morceaux de poulet peu rassasiants mais à une vitesse fulgurante, idéal pour les petites zones.

Le Chef Ranold : Spécialiste des maxi burgers, il concocte des burgers extrêmement rassasiants toutes les trois secondes, parfaits pour les clients les plus affamés.

Mécaniques de Jeu

Les clients arrivent avec différents niveaux de mécontentement et de faim. Certains sont relativement faciles à satisfaire, mais d'autres nécessitent plusieurs portions avant d'être calmés. Attention, les clients les moins affamés sont souvent les plus rapides à traverser le restaurant pour aller se plaindre au manager !

Objectif

Votre objectif est de nourrir efficacement tous les clients avant qu'ils n'atteignent le manager pour se plaindre. Utilisez vos chefs judicieusement pour répondre aux besoins variés des clients. Si trop de clients parviennent à parler au manager, la partie est perdue et vous êtes viré !

Répertoire Git :

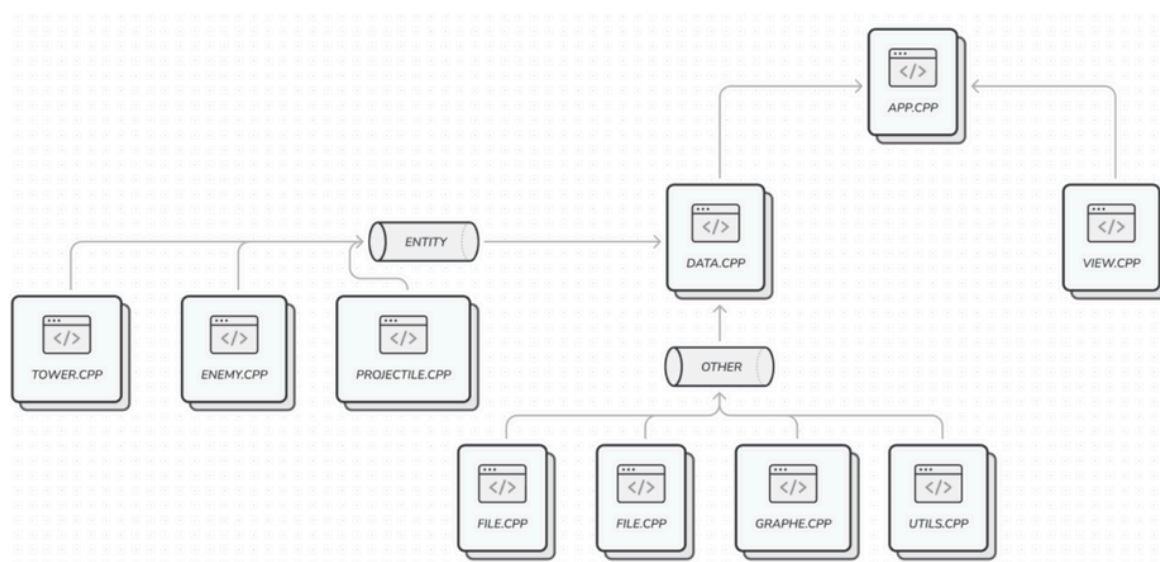
<https://github.com/ChristianJyL/ITD-ANTON-BONNOT>

ORGANISATION DU PROJET

Pour le développement de notre jeu **I Want to Talk to Your Manager**, nous avons opté pour une architecture inspirée du modèle de conception MVC (Model-View-Controller) pour structurer notre code de manière claire et modulable. Le jeu a été développé sur Windows.

```

v game
  App.cpp
  App.hpp
  data.cpp
  data.hpp
  view.cpp
  view.hpp
  > other
  
```



Voici une présentation détaillée de l'architecture du projet :

Design Pattern MVC

• Modèle (Model) :

- **Data.cpp** : Ce fichier contient toutes les données essentielles du jeu. C'est-à-dire les structures de données pour la carte, les ennemis, les tours, ainsi que certaines caractéristiques du jeu.

• Vue (View) :

- **View.cpp** : Ce fichier est responsable de l'affichage du jeu. Il contient toutes les fonctions de dessin nécessaires pour représenter les éléments du jeu.

• Contrôleur (Controller) :

- **App.cpp** : Ce fichier est le cerveau du programme. Il utilise les données de Data.cpp et les fonctions de dessin de View.cpp pour former le jeu. Le contrôleur reçoit les entrées utilisateur, les traite, et met à jour le modèle en conséquence.

Liste et descriptions

FONCTIONNALITÉS

Système de vies

Le manager possède une limite de 3 avertissements (vies affichées à droite). Si 3 clients arrivent au niveau du manager pour se plaindre, vous êtes viré.

Cartes avec plusieurs entrées

Il est possible d'avoir des cartes avec une ou plusieurs entrées. Dans le cas de plusieurs entrées, l'arrivée des clients se fera de manière alternée, ajoutant un niveau de défi supplémentaire dans la gestion de votre fast-food.

Types de tours

Il existe 3 types de tours correspondant aux chefs présentés en introduction. Chacun a une texture légèrement différente et peut être sélectionné à partir des "menus" au-dessus.

Types de Clients

Il existe deux types de clients :

- **Clients rapides** : Identifiables par leur couleur verte, ces clients se déplacent rapidement mais ont besoin de moins de nourriture pour être satisfaits.
- **Clients lents** : Identifiables par leur couleur rouge, ces clients se déplacent lentement mais ont besoin de plus de nourriture pour être satisfaits.

Vagues de Clients

Les clients arrivent sous forme de service (round affiché sur la droite). Pendant 20 secondes, un client arrive toutes les 2 secondes, suivi d'une pause de 7 secondes avant le début du prochain service. En cas de défaite, le nombre de round est indiqué sur l'écran de fin.

Progression du Jeu

Au fur et à mesure que les services défilent, les clients deviennent de plus en plus affamés, augmentant la quantité de nourriture nécessaire pour les satisfaire. En contrepartie, ces clients plus affamés rapportent également plus d'argent, vous permettant d'améliorer vos ressources et vos stratégies.

Entrées Utilisateurs

- **Sélection et Placement des Chefs** : Le joueur peut cliquer sur l'interface pour sélectionner un chef et le placer dans le restaurant.
- **Pause et Quitter** : Le joueur peut mettre la partie en pause ou quitter le jeu en cliquant sur la borne située à droite de l'écran. Le bouton de pause s'adapte selon l'état du jeu.
- **Raccourci Clavier** : En appuyant sur la touche d'espace, le joueur peut également mettre la partie en pause.

Informations graphiques lors du placement des tours

Un filtre vert indique les zones constructibles, la case survolée par la souris est délimitée par un contour rouge et la portée de la tour sélectionnée est indiquée par un carré bleu. De plus, la tour suit la souris lors du placement, comme si on "prenait" un cuisinier pour le déposer sur la carte.

Affichage des projectiles

On peut visualiser les tirs des tours sur les ennemis. Afin de mieux reconnaître l'origine des tirs, chaque type de tour a ses propres projectiles : pizza, bucket ou burger.

Bilan et conclusion

ANALYSE

CE QUI A BIEN FONCTIONNE :

- l'affichage globale des **textures**, notamment pour la carte
- Les interactions du joueur avec l'IHM

PROBLEMES RENCONTRES ET SOLUTIONS :

- **Gestion de la transparence** : les tours, projectiles et ennemis s'affichaient sur fond blanc. Il s'agit tout simplement d'adapter la fonction `loadTexture` et de modifier les paramètres d'`img::load` pour prendre en compte la couche *alpha*.

CE QU'ON FERAIT DIFFEREMMENT :

- **L'utilisation de GIT** : Dans un projet en équipe, l'utilisation des branches est quasiment indispensable, et c'est ce qui nous a fait défaut. Nous avons néanmoins réussi à nous débrouiller avec des merges.

CE QU'ON AURAIT AIME AJOUTER :

- **Du décors** autour des chemins en modifiant le fichier ITD et notre image.
- **L'animation des ennemis**, comme l'ajout de mouvements pour les jambes ou de l'orientation des personnages selon la direction dans laquelle ils marchent.
- **Une barre de vie pour les ennemis**, en superposant des rectangles suivant les personnages et dont la largeur s'adapte à leur variable *hp*.
- **L'affichage des caractéristiques des l'ennemis** au survol de ces derniers.

BILANS PERSONNELS :

- **Christian** : J'ai beaucoup aimé travailler sur ce projet, qui m'a rappelé un ancien projet en *Java*. Le thème du *tower defense* m'a particulièrement intéressé, et je suis triste de ne pas avoir eu assez de temps pour réaliser toutes les idées qu'on avait. Mon rôle principal était de travailler sur la partie algorithmique du projet, et j'ai apprécié l'utilisation concrète de l'un de nos TD (sur Dijkstra).
- **Théo** : M'étant principalement occupé de l'aspect graphique, ce projet m'a permis de concrétiser mes compétences de synthèse d'images et de mieux comprendre certaines notions vues en CM et en TD. Bien que certains points me paraissaient obscures au début du projet, ceux-ci me sont maintenant beaucoup plus clairs.

SOURCES DES ASSETS :

Toutes les textures ont été trouvées sur le site <https://opengameart.org/>

- Plancher : **DavidRStudios**
- Tapis : **Roars Games**
- Carrelage : **Screaming Brain Studios**
- Cuisiniers : **Umz**
- Clients : **NotAArtist**
- Manager : **gustavoasilveira**
- Nourriture : **KASTLE Knight**
- Borne : **retrogemn_**
- Pièce : **chillandcodegames**
- Cercle : **oglsdl**
- Dessin de fin : **Clker-Free-Vector-Images** de Pixabay
- Police interface : **Old computer St**
- Police de défaite : **Pricedown Black**