

REUNION CONCEPTION N°2 12/03

12 mars 2025 / 16:00

PARTICIPANTS

Gabriel DIYAN Mathis PACCOUD Raphaël ROULLET Thomas RAMILLON

ORDRE DU JOUR

PARTICIPANTS	1
ORDRE DU JOUR	1
NOTES	1
Review de la TODO	1
Fin du cahier des charges	2
Mise en place de la structure du projet	3
Structures de données à utiliser	3
Répartition des tâches	3
TÂCHES	5
ORDRE DU JOUR DE LA SEMAINE PROCHAINE	5
Prochaine réunion : mercredi 19 mars à 16h	5

NOTES

Review de la TODO

Certaines personnes du groupe ont pu tester l'option. Thomas RAMILLON a fait une documentation à lire pour le mercredi 19/03.

Etat de l'art : Mathis PACCOUD a fait le début de l'état de l'art et s'est renseigné sur les algorithmes suivants:

- min-max

- alpha-beta

Mais ces algorithmes ne sont pas exactement adaptés au jeu il faudra donc les modifier pour pouvoir les utiliser. On commencera par pathfinding assez rudimentaire qui calcule tous les chemins avec une profondeur faible puis on pourra l'améliorer avec de l'élagage.

- deep learning (reinforcement / CNN)

Plus compliqué à implémenter surtout en C.

Étude des fonctionnalités du jeu : Il y a un document dans prise de note qui répertorie toutes les fonctionnalités du jeu. Raphaël ROULLET a trouvé un dossier de sprite en .obj . On classera les obstacles en 3 niveaux de difficulté d'implémentation, on commencera par implémenter le niveau 1 puis le 2 et le 3.

On doit faire attention au fait qu'il doit toujours avoir un chemin possible pour le joueur.

Le jeu n'accélère pas mais a une vitesse variable sur certaines portions.

On peut trouver plusieurs versions du jeu partout sur le net avec chacun sa version du menu de skin , des options et des paramètres.

Pipeline : Le pipeline fonctionne sans problème : il ne faudra pas oublier d'ajouter des tests afin qu'elle serve à quelque chose.

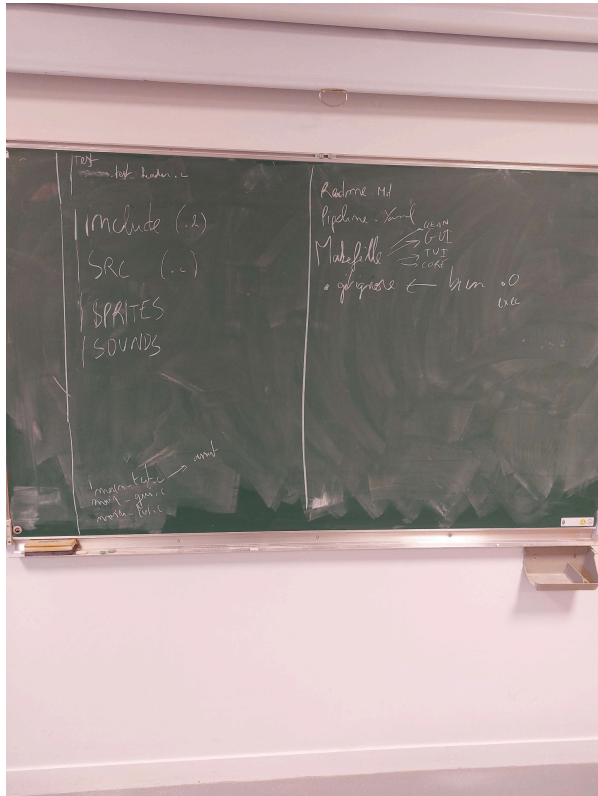
Jeu textuel : s'actualise seul sans forcément que le joueur ait à presser une touche.

Fin du cahier des charges

Le cahier des charges se trouve sur le drive, on a juste changé quelques options. Pour l'affichage des textures en 3D : on pourra utiliser les VOXEL.

Bibliothèque pour gérer les voxel SDL : <https://github.com/woxels/Woxel>

Mise en place de la structure du projet



On pourra rajouter un dossier lib si besoin.

Structures de données à utiliser

La gestion de la grille se fait avec des coordonnées flottantes ou entières ? On garde le fonctionnement de la grille. La position du poulet peut être flottante, de même pour les véhicules. On calcule à chaque itération sur quelle case est la voiture (une ou deux cases en fonction de la position de la voiture). On commence par une grille classique (tableau à 2 dimensions) et on pourra peut être implémenter 2 positions (grille pour pathfinding , etc..) et une position précise.

Répartition des tâches

La structure du jeu (plateau , poussin , voiture, position) doit être continuée pour la prochaine fois.

C'est le terrain qui bouge : le joueur reste toujours au même endroit et la grille se "déplace" par rapport au joueur.

- Game
 - Board

- grid.obstacle Obstacle **
 - grid.player Player *
 - grid.Game Game*
- Score : int
- status : int
- Object :
 - Player
 - skin skin *
 - alive bool
 - direction int
 - h_position : float
 - is_jumping bool
 - Obstacle
 - id : int
 - type : char
 - model : char
 - h_position : float
 - velocity : float
 - ground
 - type : char
- Skin : - name char *
 - model_path char *

Menu :

- buttons Button *
- name char *

Button :

- text char *
- position_x float
- position_y float
- height float

- width float

Gérer les fichiers du gitlab (créer les .gitignore, makefile ,pipeline)

Trouver une librairie pour afficher la grille.

Coder options : ajouter des éléments à la grille grille / retirer un élément à la grille.

TÂCHES

1. Lire la doc sur gitlab **Gabriel DIYAN** **Thomas RAMILLON**
Mathis PACCOUD **Raphaël ROULLET**
2. Mettre sous forme de Google sheets et trier le cahier des charges (mettre la priorité des éléments) + trier aussi par catégorie **Gabriel DIYAN**
3. Créer les structures du projet **Thomas RAMILLON**
4. Gérer les fichiers du gitlab **Gabriel DIYAN**
5. Trouver une lib pour affichage + actualisation
Raphaël ROULLET
6. Finir état de l'art **Mathis PACCOUD**
7. Coder fonctions associés à la structure :
 - a. Obstacle **Thomas RAMILLON**
 - b. Player **Raphaël ROULLET**
 - c. board **Mathis PACCOUD**
 - d. game **Gabriel DIYAN**
 - e. Ground **Thomas RAMILLON**

ORDRE DU JOUR DE LA SEMAINE PROCHAINE

- Review de la T0-D0.
- Discuter des choix d'implémentation.
- Revoir les potentiels problèmes liés aux structures.
- Fixer les prochaines étapes du projet.

Prochaine réunion : mercredi 19 mars à 16h