

REUNION CONCEPTION N°6 23/04

23 avril 2025 16h

PARTICIPANTS

Gabriel DIYAN Mathis PACCOUD Raphaël ROULLET Thomas RAMILLON

ORDRE DU JOUR

PARTICIPANTS	1
ORDRE DU JOUR	1
NOTES	1
Review de ce qui a été fait pendant les vacances d'avril	1
Nouvelles structures	1
Fichiers pour SDL	2
Ajout de l'ia	2
Branches du git	2
TÂCHES	3
ORDRE DU JOUR DE LA SEMAINE PROCHAINE	3
Prochaine réunion : Mardi 29 avril 11H	3

NOTES

Review de ce qui a été fait pendant les vacances d'avril

Nouvelles structures

De nouvelles structures ont été implémentées spécifiquement pour le GUI:

- `display`

Contient des infos sur la grille

- `cam`

Contient des infos sur la cam (sa position,)

- `colors`

Ensemble des couleurs utilisées dans SDL. Ce ne sont pas des macros car moins pratique.

- ``renderer``

Structure relative à SDL.

- ``debug``

Contient le statut des différents modes de debug, la vitesse du jeu, etc.

Nouveaux modules

L'implémentation du GUI est accompagnées de deux nouveaux modules (.c et .h):

- ``gui.c``

affichage de polygones

- ``tff.c``

affichage de texte, ombrages

La boucle de jeu et les appels de fonctions SDL se trouvent principalement dans ``main_gui.c`` (où la bibliothèque SDL est *inclue*)

A noter que le module ``obstacle`` dépend de la structure ``render``, et que certaines infos sont passées en arguments en cascade pour cette raison.

A propos de la future IA

Après discussion, le module IA devra contenir:

- une fonction finale qui donne le prochain coup à jouer en fonction de l'état du jeu (à voir à l'utilisation où elle sera appelée exactement dans la boucle de jeu)
- un structure ``IA`` pour stocker les paramètres et les tableaux pour mémorisation et hitbox
- plusieurs niveaux d'ia (plusieurs fonctions finales)
 - ne fait qu'avancer ou attendre (pour tester)
 - parcours en profondeur de l'arbre des coups
 - avec élagage
 - regarder les cases finales et les chemins qui y mènent plutôt que le prochain coup (comme aux échecs)
 - ...

Un problème a été soulevé à propos de la manière de calculer les prochains: l'animation de saut prend bien plus de temps que les frames internes, ce qui peut être gênant pour *prévoir* le mouvement des obstacles et agir en conséquence

Plutôt que le multithreading, très dangereux car pas évident en c, c'est la solution de la segmentation du temps qui a été retenue pour le moment: attendre prend pour l'ia autant de temps que sauter.

Branches du git

Discussion à propos de l'avenir des branches du dépôt git. Il y avait alors ``main`` qui contient le tui (terminé), ``dev``, qui avait été utilisée pour le développement du mode TUI mais est abandonné car contient des erreurs, et ``new_dev``, où est actuellement développé le mode GUI.

Finalement:

- la branche `main` reste définitivement pour le tui et on n'y touchera plus a priori;
- l'ia sera développée sur `new_dev` puisque le GUI est fonctionnel et ne recevra a *priori* plus de màj lourde;
- `dev` est définitivement abandonnée.

Il y a un README pour chacune des deux branches principales.

TÂCHES

- Menu in game gui (pause, rejouer) **Raphaël ROULLET**
- skins **Raphaël ROULLET**
- implémentation de l'ia 1: parcours en profondeur avec segmentation du temps **Thomas RAMILLON** **Mathis PACCLOUD**
- réécrire tous les tests **Gabriel DIYAN**
- recommander tout le code **Raphaël ROULLET**
- update le trello **Mathis PACCLOUD**

ORDRE DU JOUR DE LA SEMAINE PROCHAINE

- review de la todo

Prochaine réunion : Mardi 29 avril 11H