

Rapport puyo projet

Fait tout seul par Rui Cesista.

J'ai fait le mode simulation et le mode IA, et le mode solo.

```
//constructeur
```

```
Grille();
```

```
Grille(std::vector<std::vector<puyoType>>tableau);
```

```
Mode();
```

```
Mode(std::vector<std::vector<puyoType>>tableau)
```

```
/** Change les types puyotypes en caractères
```

```
 * @param p un type puyo
```

```
 * @return un caractères
```

```
 */
```

Je suis sûr que ça marche

```
char PuyoToChar(puyoType p)
```

```
/** Change un caractère en un type puyo si le caractère fait partie de cela
```

```
 * @param un char p
```

```
 * @return un puyotype
```

```
 */
```

Je suis sûr que ça marche

```
puyoType CharToPuyo(char p)
```

```
/** Montre la place des puyo dans le terminal
```

```
 */
```

Je suis sûr que ça marche

```
void montrepuyo();
```

```
/** Calcule le score cumulé
```

```
 * \param un entier a, un entier combo, un entier puyo, un entier groupe
```

```
 * \return return a sinon return 999999 si a supérieur à ce nombre
```

```
 */
```

Affiche un mauvais score pour certain mode simulation

```
int scoreplus(int a, int combo, int b, int groupe);
```

```
/** \brief compte les puyo et les enleve quand il y a assez côte à côte et définit le score en plus
```

```
 */
```

Quelque bug mais je sais pas où car la méthode pour enlever les puyo est la règle 2 et c'est marqué défaite lorsque je teste avec le méga combo test du mode Simulation

```
void compte_puyo();
```

```
/** \ enlève les puyos dont les positions sont mis dans un tableau
```

```
 * \param lav un tableau2D
```

```
 */
```

```
void del(std::vector<std::vector<int>>lav);
```

```
/** compte les puyos si ils sont égale au puyo d'origine et rajoute leur coordonné dans un tableau
```

```
 * @param un entier x, un entier y et une type puyoType c
```

```
 */
```

```
void Count(int x, int y, puyoType c);
```

```
//quelque test
```

```
void testplacement1();
```

```
//quelque test
```

```
void testplacement2();
```

```
/** Fait tomber les puyos dans le bas du tableau et renvoie false si ils sont tous tombés
```

```
 * \return false si les puyos sont tous tombés
```

```
 */
```

Je suis sûr que ça marche car mon jeu fonctionne

```

bool jeugravite();
/** test si le haut du tableau est rempli
 * @return true si c'est vrai sinon false
 */
Je suis sûr que ça marche car j'ai tester avec le mode solo
bool IsGameOver();
/** créer la prochaine piece qui va tomber et la placer
 * @param un entier n un entier w
 * @param un caractère p un caractère z
 */
Plutôt confiant car il passe les mode simulation
void createnextpiece(int n,int w,char p,char z);
/** écrit dans le fichier le tableau des puyos en char
 * \param abc un fichier ofstream
 */
confiant
void retransforme(std::ofstream& abc);
/** \ Affiche le score dans le terminal
ça marche
void montre_score();
/** test si le haut du tableau est rempli
 * @return true si c'est vrai sinon false
 */
ça marche
void game_over(std::ofstream& abc);
/**Fait le mode simulation et renvoie le résultat dans le fichier
ça marche à part le dernier test
void Mode_Simulation(std::ifstream& abc);
/* fait le mode IA et renvoie le résultat dans un fichier
void Mode_IA(std::ifstream& abc);
/** Fait le mode solo et renvoyer le résultat dans un fichier
ca marche mais faut avant finir toute les pièces ou avoir une défaite pour avoir un fichier de sortie
void Mode_Solo(std::ifstream& abc);

```

Alors j'ai eu du mal à implanter la gravité car mon fonction avait des probleme pour le faire donc j'ai du changer la méthode pour la gravité. Aussi il y a la méthode pour faire enlever les puyos en règle 2 il y avait pas mal de problème par rapport à l'effacement des puyos côte à côte lorsqu'ils sont plus de 3.

Il y a encore des problèmes au niveau de la règle 2 de puyo car le test 5 du mode simulation me marque défaite alors que quand je teste avec un tableau rempli de puyo tout les puyo qui sont du même type et adjacent disparaisse.

Aussi j'ai un problème niveau score il affiche un score qui n'est pas celui attendu alors que quand je test avec des nombres cela marche.