Rapport : Projet puissance 4

Structures de données:

On a défini un tableau **tabCase** de 42 bytes initialisées à 0 pour représenter la grille du jeu et pour pouvoir la parcourir facilement.

UpdateRecord (Hussein et Hamza):

La fonction ***UpdateRecord*** prend en paramètres le numéro de la colonne choisi par l'un des joueurs (1-7 stocké dans **$v0**), elle soustrait ensuite -1 de ce input pour le transformer en notation de tableau (0-6) pour pouvoir parcourir le tableau de cases **tabCase** puis la sous-function ***checkifpleine*** commence par la 1ere case de la colonne et elle vas ensuite tester si l'input du joueur est bien entre 0 et 6 inclus (entre 1 et 7 inclus pour le joueur) sinon elle saute vers ***HorsGrille*****.**
La sous fonction ***PreCheckifpleine*** permet de passer à la case en haut en ajoutant 7 à **$v0.**

Sous fonctions de UpdateRecord:

- ***Checkifpleine*****:** Cette fonction parcourt la colonne ou le joueur a choisi de placer le jeton pour tester si une place est disponible en testant à chaque fois si **$v0** n'est pas supérieur à 41(indice de la dernière case) et si la case n'est pas vide sinon on stock le numéro du joueur dans cette case.
- ***colonePleine*****:** Cette fonction affiche le message de colonne pleine et saute vers **rejouer** qui repasse le tour au même joueur pour lui permettre de choisir une autre colonne.
- ***HorsGrille*****:** Cette fonction affiche le message ***"Choisissez un nombre entre 1 et 7 (inclus)"***

```
Tour du joueur 1 : 8
Choisissez un nombre entre 1 et 7 (inclus)
```

WinCheck (Hamza et Hussein):

Cette fonction vérifie les 5 cas de victoire. Dans chacun de ces cas on commence toujours par initialiser un compteur et stocker le numéro de la case choisie dans le registre **$t2**. Les 5 cas sont:

- **Vérification de la ligne horizontale:**
  Cette fonction commence par vérifier si le jeton n'est pas tout à gauche ou tout à droite de la ligne grâce à une opération mathématique pour savoir s'il est possible de vérifier les cases dans les deux côtés de ce jeton. Pour aller d'une case à une autre, on ajoute +1 ou -1 a l'indice de la case. On incrémente le compteur **$t4** a chaque fois qu'elle trouve une case avec la même valeur que le numéro du joueurs actuel et si **$t4** égale à 4, saute vers ***PlayerWon***.
- ***Vérification de la ligne vertical:***
  Même principe que la sous-fonction précédente mais cette fois le test se fait verticalement et on a juste besoin de vérifier les cases en dessous du jeton en décrémentant **$t2** par -7.

- ***Vérification de la diagonale avant :***
  Ici on commence par vérifier si le jeton n'est pas soit tout en haut à droite ou tout en bas à gauche pour savoir si on saute vers ***verifieBG*** ou ***verifieHD*****.**
- ***Vérification de la diagonale arrière:***

Ici on regarde si le jeton n'est ni en haut à gauche, ni en bas à à droite pour savoir si on saute vers ***verifieBD*** ou ***verifieHG*.**

- ***Vérification d'égalité:***
  Qui vérifie juste si la grille est remplie et qu'aucun des joueurs n'a gagné ensuite saute vers ***GameTie*.**

GamteTie - PlayerWon:

Les deux fonctions affichent un message sur l'écran dépendant du résultat de la partie.

- ***GameTie (Hussein)***:
  Le programme vérifie d'abord la condition du remplissage de la grille. Si il y a égalité donc le programme saute vers ***GameTie*** puis affiche le message ***"Il y a égalité !"***
- ***PlayerWon (Hamza)***:
  La fonction compare la dernière valeur de **$a0** qui est le numéro du joueur avec le joueur numéro 1 $t5 si les deux sont pas égaux donc joueur 2 gagne la partie et on affiche ***"Le joueur 2 à gagner !"*** Sinon le programme affiche forcément que le joueur 1 a gagné.

Relancement de partie (Hussein et Hamza):

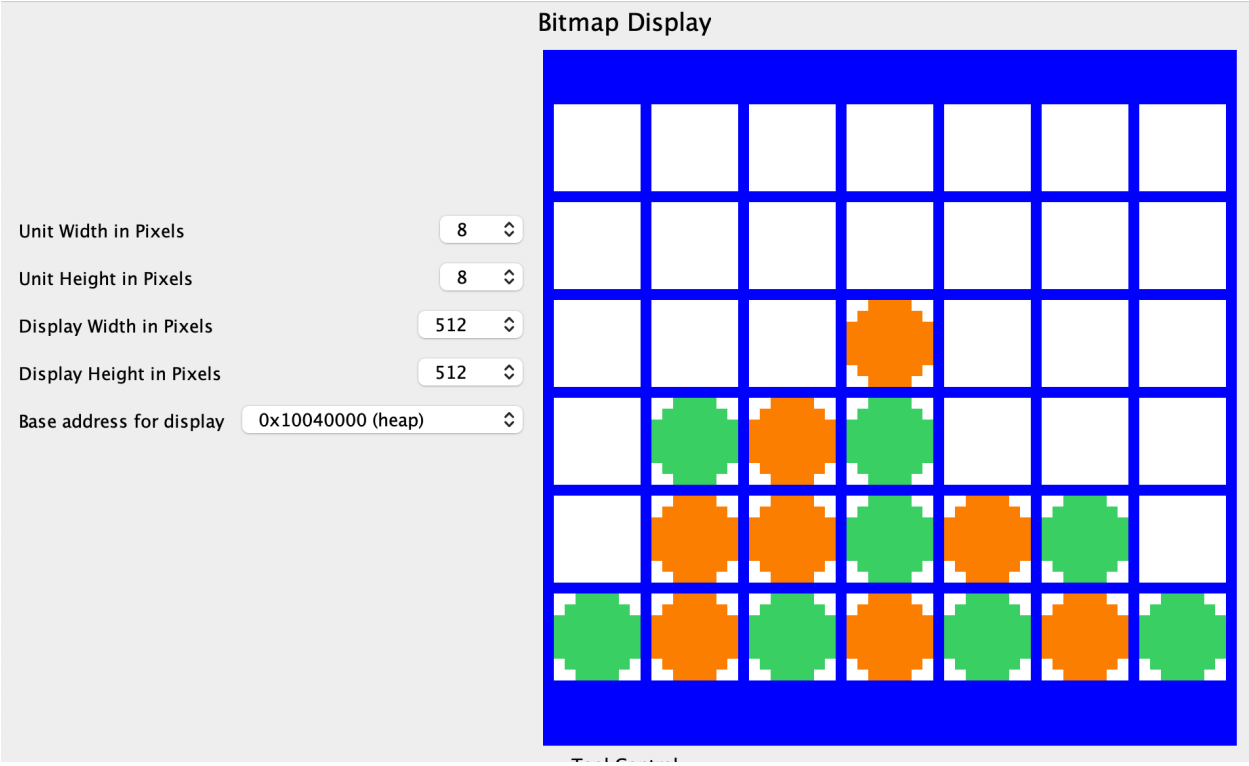
Cette fonction suit la fonction ***PlayerWon***.
Son rôle est d'afficher à l'écran ***"Voulez vous rejouer? (1): oui, (0): non."***
Et prend comme input un int (1 - 0) qui détermine si l'utilisateur veut relancer une partie ou pas.
Si oui, la fonction réinitialise chaque case du tableaux ***tabCase*** a 0 pour supprimer les anciens placement des jetons de la partie précédente. Puis saute à la fonction ***Inite*** pour le lancement de la nouvelle partie.

```
Tour du joueur 1 : 1
Le joueur 1 à gagné !

Voulez vous rejouer? (1): oui, (0): non.
```

fonctionnalités ajoutées (Hussein et Hamza):

Dans le tableau prédéfini **Colors** on a ajouté d'autres couleurs et on a modifié le programme tel que les numéros des joueurs ne sont plus juste 1 ou 2 mais correspondent à la couleur choisie par chacun au début du jeu (donc de 1 à 6) .



Difficultés rencontrées:

Une petite difficulté qu'on a rencontré était pour la fonction bonus de laisser les joueurs choisir une couleur, c'est de garder les valeurs de la couleur qui sont associées au joueur à travers tous l'exécution du programme.