Maxime Demers Boucher

Table des matières

[Les buts 1](#_Toc122004898)

[Liste des concepts 1](#_Toc122004899)

[Références 1](#_Toc122004900)

[Cartographie 1](#_Toc122004901)

[Tableau comparatif avec autre tech 2](#_Toc122004902)

[Journal des enjeux 2](#_Toc122004903)

[Conclusion 4](#_Toc122004904)

# Les buts

* Apprendre une nouveau langage
* Création d’un jeux de puissance 4
* Comprendre les avantages du langage

# Liste des concepts

Séparations des tâches

Compréhension facile

Changement des tours des joueurs

Victoire d’une joueur

Gravité dans la colonne ne peut pas placer un jeton au milieux du tableau

# Références

https://www.quora.com/What-are-the-advantages-of-the-Delphi-programming-language

<https://www.simplilearn.com/tutorials/cpp-tutorial/top-uses-of-c-plus-plus-programming>

<https://www.reddit.com/r/fsharp/comments/nus5rg/what_are_fs_advantages/>

# Cartographie

# Tableau comparatif avec autre tech

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Valeur: | 1 | 1 | 1 | 3 | 3 |  |
| Langage | Override fonction | Outils avancer | Librairie | Facile à lire | Apprendre(dans le temp limité) | Total |
| Delphi | 1 | 1 | 1 | 2 | 2 | 15 |
| F# | 0 | 0 | 1 | 3 | 3 | 19 |
| C++ | 1 | 1 | 1 | 2 | 2 | 15 |

# Journal des enjeux

* Changer la valeur d’une variable. (Ajouter mutable pour pouvoir changer une variable)

let mutable donne = ""

* Pour changer une valeur utilisé <-

        ligne<-ligne+1

* L’appelle a une fonction est comme celle d’une variable

//trouver la case vide la plus base d'une colonne

let TrouverY(colonneChoisie:int) =

    //-1 pour une ligne pleine

    valeurRetour<- -1

    ligne<-5

    while loopligne do

            if grilleJeu[ligne,colonneChoisie] = null

                then valeurRetour<-ligne

                     loopligne<-false

                else

                     //vérifie si la ligne est pleine

                     if ligne = 0

                     then loopligne<-false

                     else ligne<- ligne-1

    //reset les valeurs pour la prochaine utilisation

    loopligne<-true

    ligne<-0

    valeurRetour

* Try with au lieu de try catch

    try

        x <- Console.ReadLine() |> int

    with

    | :? SystemException as ex->

            printfn"La donnée fourné n'est pas un nombre"

            x <- -1

* Parce

        x <- Console.ReadLine() |> int

* Declaration d’un tableau 2D

Array2D.zeroCreate

* La verification si un joeur a gagner. Sauvegarde du dernier jeton inseré et vérification de chaque coté.

//vérifie la ligne

let VerifierLigne()=

    //jeton le plus éloignier sur la droite en bas

    donne<- grilleJeu[dernierJeton[1],dernierJeton[0]]

    distanceD<-0

    distanceG<-0

    //jeton le plus éloignier sur la droite

    while loopligne do

        if (dernierJeton[0]+distanceD) <= 6

            then

                if grilleJeu[dernierJeton[1],(dernierJeton[0]+distanceD)] = donne

                    then distanceD<- distanceD+1

                    else distanceD<- distanceD-1

                         loopligne<-false

            else distanceD<- distanceD-1

                 loopligne<-false

    loopligne<-true

    //jeton le plus éloignier sur gauche

    while loopligne do

        if (dernierJeton[0]-distanceG) > 0

            then

                if grilleJeu[dernierJeton[1],dernierJeton[0]-distanceG] = donne

                    then distanceG<- distanceG+1

                    else distanceG<- distanceG-1

                         loopligne<-false

            else distanceG<- distanceG-1

                 loopligne<-false

    loopligne<-true

    //vérifier ligne

    if (distanceD+distanceG+1) >=4

        then partieEncour<- false

             AfficherTableau()

             if joueur1

                then printfn "le joueur 0 a gagner"

                else printfn "le joueur X a gagner"

# Conclusion

F# est un langage facile à comprendre , facile à apprendre et agréable à lire. Le langage utilise les indentation pour se repérer se qui élimine les risques d’erreur de syntaxe. F# ne possède pas un grand avantage qui le rend unique de là vient la très basse popularité du langage malgré la lecture et écriture agréable.