



OTR – OVER THE RYTHM

Planning & Bilan

Auteurs : **MARTIN Justine, BOTTON Léa**
Date : **22 - 02 - 2018**
Filière : **SIO-1**

Résumé de votre projet :

Création d'un mini-jeu avec Monogame (Visual Studio 2017).



CONTENU

I.	DOCUMENTATION	2
A.	Singleton.....	2
B.	IScreen (Interface screen)	2
II.	EXEMPLES DE MODIFICATION POSSIBLE	3
A.	Modifier la taille des images	3
B.	Changer les touches pour jouer	3
C.	la musique	3
D.	Modifier l'image de fond.....	3
E.	Gerer une nouvelle colonne de jeu	4
F.	Modifier la position du texte sur l'écran de jeu	4

I. DOCUMENTATION

A. SINGLETON

```
public class OTRGame : Game
{
    //Le singleton empeche l'instanciation d'une classe (new MaClasse() ).
    //On utilise getInstance() pour récupérer la SEULE instance de la classe qui existe
    // ===== DEBUT SINGLETON =====
    private static OTRGame instance;

    public static OTRGame getInstance()
    {
        if(instance == null)
        {
            instance = new OTRGame();
        }
        return instance;
    }
    // ===== FIN SINGLETON =====
}
```

- ➔ On utilise ici le mot clé “static” dans la définition de la variable instance afin que celle-ci prenne la même valeur peu importe l’instance d’OTRGame.
- ➔ Le fait qu’elle soit “private” empêche une modification quelconque en cas d’héritage (inexistant ici).
- ➔ Le constructeur OTRGame, déclaré par la suite (*non visible sur la capture d’écran, cf : code source*), est donc privé et non public pour empêcher toute nouvelle instance.
- ➔ La fonction getInstance() est déclarée “static” afin que son utilisation ne nécessite pas une nouvelle instanciation.

Le reste de la classe OTRGame permet de gérer l’affichage et les différents écrans.

B. ISCREEN (INTERFACE SCREEN)

L’IScreen (classe) permet de gérer la logique des différents écrans indépendamment.

II. EXEMPLES DE MODIFICATION POSSIBLE

A. MODIFIER LA TAILLE DES IMAGES

```
public float scale = 1.0f;
```

En agissant sur la variable `scale`, ligne 22, de la classe `Sprite`, on modifiera la taille des images affichées. Le `f` accolé au nombre permet d'indiquer que ce dernier est flottant (donc pas un double) et évite ainsi les erreurs.

B. CHANGER LES TOUCHES POUR JOUER

Dans la classe `GameMap`, ligne 88 :

```
Keys[] inputs = new Keys[] { Keys.D, Keys.F, Keys.J, Keys.K };
```

Par exemple, si on veut jouer avec la touche `Q` à la place de la touche `D` ; on modifie `Keys.D` par `Keys.Q`.

On pensera à modifier cette même ligne dans la classe `SpriteLight`, ligne 23 afin que les lumières d'animation se déclenche bien à l'appui des touches.

Dans cette même classe, on pourra modifier l'effet d'animation en remplaçant l'image `light` importée, ligne 14.

```
texture = OTRGame.getInstance().Content.Load<Texture2D>("light");
```

C. LA MUSIQUE

```
music = OTRGame.getInstance().Content.Load<Song>("Tobu - Higher");  
MediaPlayer.IsRepeating = true;  
MediaPlayer.Play(music);
```

Dans la classe `GameScreen`, lignes 19/20/21, on peut modifier le fichier dans la variable `music`. On peut supprimer la répétition de la musique pendant le jeu (en la mettant en `false`), pour démarrer ou arrêter le son, il faut passer par la classe `MediaPlayer` en utilisant les méthodes `Play` ou `Stop`.

D. MODIFIER L'IMAGE DE FOND

Dans la classe `GameMap`, ligne 46, remplacer l'image (ou importer une nouvelle image sous le nom déjà donné !).

```
public GameMap() : base("map")
```

E. GERER UNE NOUVELLE COLONNE DE JEU

➔ Dans les classes `GameMap` et `SpriteLight`, ajouter la touche voulue (ici la touche Espace).

```
Keys[] inputs = new Keys[] {Keys.D, Keys.F, Keys.J, Keys.K, Keys.Space };
```

➔ Dans la classe `GameMap`, dans le constructeur, ligne 60, passer le nombre 4 à 5 (ou au nombre souhaité).

```
for (int i = 0; i < 4; i++)  
    lightSprites.Add(new SpriteLight(i));
```

➔ Dans la classe `GameMap`, ligne 198, rajouter le nombre 4 à la suite.

```
List<int> position = new List<int>() { 0, 1, 2, 3 };
```

Penser à modifier le design de l'image de fond pour que ce soit adapté ! ;)

F. MODIFIER LA POSITION DU TEXTE SUR L'ECRAN DE JEU

```
batch.DrawString(font, "Score : " + score, new Vector2(OTRGame.getInstance().Window.ClientBounds.Width - tailleScore.X - 50, 20), Color.White);
```

Dans la classe `GameMap`, ligne 232, changer la position de l'affiche du score (ici) ou du nombre d'erreurs.