

## Jeux :

Faire une classe qui hérite de IGame avec au moins ces 2 méthodes :

```
virtual int launchGame(IGraphicLib *glib);
```

Cette fonction sera appelée une fois par l'arcade, c'est celle qui vous permettra de démarrer votre jeu et d'initialiser toutes vos variables.

Elle renvoie un int qui sera le score réalisé.

Cas spéciaux de return :

"-1" : n'enregistre pas le score et quitte le programme.

"-2" : n'enregistre pas le score et reviens au menu principale

Dans tout autre cas le score sera comparé avec les scores du leaderboard et sera enregistré s'il est supérieur à l'un d'eux.

```
virtual int gameLoop(IGraphicLib *glib);
```

C'est dans cette deuxième fonction que se trouvera votre boucle principale, elle sera appelée par launchGame et renvoie aussi le score réalisé.

Vous devrez en plus de ces deux fonction aussi ajouter ce morceau de code pour permettre à votre classe d'être reconnu comme un jeu:

```
extern "C"
{
    Game *MakeGame()
    {
        return new NomDeVotreClasse();
    }
}
```

Tous les "virtual" sont nécessaires pour le bon fonctionnement de votre librairie partagée.

## Librairie :

Faire une classe qui hérite de IGame avec au moins ces 10 méthodes :

```
virtual void init_lib();
```

Cette fonction s'exécute en premier et qu'une seule fois, ce qui vous permet d'initialiser tous vos attributs comme votre fenêtre par exemple.

```
virtual void exit_lib();
```

Cette fonction s'exécute en dernier et qu'une seule fois, ce qui vous permet de fermer votre fenêtre par exemple.

```
virtual void assetLoader(const std::string path);
```

Cette fonction est faite pour charger tous les assets d'un dossier, elle prend en paramètre le chemin vers un dossier.

```
virtual int keyPressed();
```

Cette fonction renvoie la touche du clavier appuyer.

```
virtual void printMap(std::vector<std::string> map);
```

Charge la map passé en paramètre où chaque string équivaut à une ligne et un caractère équivaut à un sprite.

```
virtual void printText(int x, int y, std::string str);
```

Charge une chaîne de caractère à la position (x, y).

```
virtual void printButton(int x, int y, std::string str);
```

Charge un bouton à la position (x, y) avec "str" comme texte à l'intérieur.

```
virtual void printSelectedButton(int x, int y, std::string);
```

Charge un bouton sélectionné à la position (x, y) avec “str” comme texte à l'intérieur.

```
virtual void clearWindow();
```

Charge un fond noir sur toutes la fenêtre.

```
virtual void refreshWindow();
```

Affiche tous les éléments chargés.

Vous devrez en plus de ces dix fonction aussi ajouter ce morceau de code pour permettre à votre classe d’être reconnu comme une lib graphique:

```
extern "C"
{
    Game *MakeGraphicLib()
    {
        return new NomDeVotreClasse();
    }
}
```

Tous les “virtual” sont nécessaires pour le bon fonctionnement de votre librairie partagée.