EPITECH.
LE FUTUR DE L'INFORMATIQUE
LE MEILLEUR DE L'INNOVATION

ARCADE

DOCUMENTATION

Classe IGame

Description

```cpp
/*
** EPITECH PROJECT, 2021
** B-OOP-400-NCE-4-1-arcade-marton.szuts
** File description:
** IGame
*/

#ifndef IGAME_HPP_
#define IGAME_HPP_

#include <string>
#include <vector>
#include <map>
#include "Key.hpp"

class IGame {
    public:
        virtual ~IGame(){};
        virtual void init() = 0;
        virtual bool update() = 0;

        virtual std::map<char, std::string> getTiles() const = 0;
        virtual std::vector<std::string> getMap() const = 0;
        virtual size_t getScore() const = 0;
        virtual size_t getBestScore() const = 0;
        virtual std::string getName() const = 0;

        virtual void setKeyPressed(Key k) = 0;
};

#endif /* !IGAME_HPP_ */
```

Les méthodes **Init** et **Update** sont les deux principales fonctions de l'interface IGame.

La fonction **init** a pour rôle d'initialiser les données nécessaires au bon fonctionnement de votre jeu.

La fonction **update** vas être appeler en boucle par le "core" de l'arcade, c'est ici que la map vas être mis à jour et où les conditions de victoire et de défaite sont vérifiées.

La fonction **getMap** vas permettre l'envoie à la lib Graphique de la map du jeu.

Les fonction **getScore**, **getBestScore** et **getName** sont là pour vous permettre d'implémenter un système de score et de nom de joueur.

La fonction **SetKeyPressed** met à jour la touche appuyer par le joueur, les touches possibles sont dictées par l'enum *KEY* disponible dans le fichier *Key.hpp*. Cette enum est nécessaire car les touches sont capturées par la lib Graphique et elles n'ont pas toutes la même convention au sujet des touche.

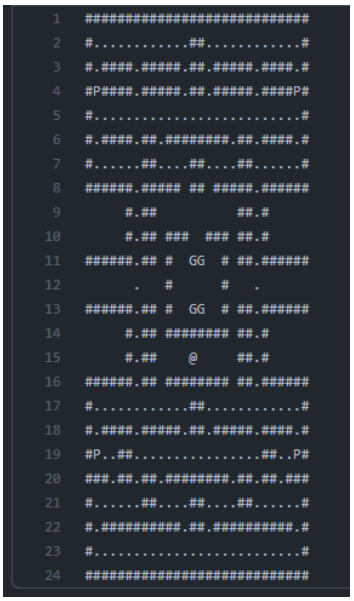
Pour finir la fonction **getTiles** permet à l'aide d'un fichier de config, d'avoir un rendu pour chaque jeu avec des sprite différents.

Maintenant que l'interface est expliquée on peut passer à sont implémentation dans un jeux.

Implémentation

La map :

Pour implémenter votre jeu avec notre interface vous devez d'abord avoir une map en caractère ascii ou chacun des caractères de la map sera interprété par la lib graphique et remplacé par une texture. Sur la droite un exemple de map de Pacman où chaque caractère représente un élément du jeu.



Les textures pour la lib graphique sont donnés par le jeu avec la fonction **getTiles**. La fonction va retourner une map qui assigne un caractère à une texture. Cette fonction s'appuie sur un fichier de configuration qui suit la convention suivante :

Chemin=[Caractère ascii]

Exemple :

```
db/assets/pacman.png=@
db/assets/ghost.png=G
db/assets/ghostWeak.png=w
db/assets/point.png=.
db/assets/gum.png=P
db/assets/wall.png=#
```

Note :

Pour les lib Graphique fournit avec le core, toutes les textures fournies par le jeu doivent suivre le format 32x32 sinon elles seront considérées comme une image et ne subirons aucune contrainte d'affichage.

Classe IGfx

Description

```cpp
/*
** EPITECH PROJECT, 2021
** arcade
** File description:
** lib graphique interface
*/

#ifndef IGFX_HPP
#define IGFX_HPP

#include <string>
#include "Key.hpp"
#include <map>
#include <vector>

class IGfx {
    public:
        virtual ~IGfx(){};
        virtual bool init(const std::string &map, std::map<char, std::string>) = 0;
        virtual Key getKeyPressed() = 0;
        virtual void drawText(size_t x, size_t y, std::string text, size_t size) = 0;

        virtual void draw(std::vector<std::string>) = 0;
        virtual void display() = 0;
        virtual void clear() = 0;
};

#endif
```

Init est une fonction qui sert pour initialiser les lib graphiques (la taille de la page …) ainsi que l'initialisation des textures selon le retour de la fonction **getTiles** de IGame

La fonction **drawText** permet d'afficher du Text sur le rendu final

getKeyPressed capture l'entrée du joueur pour ensuite que le core transmette la touche au IGame.

C'est dans la fonction **draw** que vous allez dessinée toutes la map renvoyer par **getMap** de IGame.

Les fonctions **display** et **clear** vont simplement faire appel aux fonctions qui vont **clear** le rendu est affichée le nouveau rendu dessiné juste avant.

Petite particularité dans les arguments de la fonction Init, vous pouvez remarquer qu'il prend deux arguments, la map du jeu et une map, de char et string.

La map du jeu vous permet d'initialiser la taille de la fenêtre selon la taille de la map du jeu

La map de char et de string, est la map qui permet d'attribuer les caractères de la map du jeu au Texture.

Implémentation

L'implémentation d'une lib Graphique est très libre, vous pouvez utiliser le minimum de paramètre et quand même avoir un rendu avec un simple affichage de map.