

Documentation Arcade (Cyprien Ricque, Axel Renard, Emilien Delevoye) :

Créer un nouveau jeu :

Pour implémenter un nouveau jeu, il est nécessaire de créer une nouvelle classe qui hérite de IGameModule :

```
class IGameModule {
public:
    IGameModule() = default;
    virtual ~IGameModule() = default;

    virtual void init() = 0;
    virtual void run(char input, float smoothness) = 0;

    virtual std::vector<std::vector<color_t>> getFramebuffer() const = 0;
    virtual std::vector<text_t> getTextures() const = 0;
    virtual std::vector<std::pair<std::string, float>> getSoundsStart() const = 0;
    virtual std::vector<std::string> getSoundsStop() const = 0;

    virtual int getScore() const = 0;
    virtual bool isOver() const = 0;
};
```

Constructeur de IGameModule :

- Initialisation des assets

Destructeur de IGameModule :

- Destruction des assets

Fonction init() de IGameModule :

- Initialisation des variables locales (score...)

Cette fonction est appelée à chaque lancement du jeu et ne retourne rien. L'appel de cette fonction va réinitialiser le jeu.

Fonction run() de IGameModule :

- char input : la valeur ascii correspondante à la touche du clavier enfoncée
- float smoothness : le nombre de pixels du framebuffer qui sont nécessaire pour afficher un pixel en fonction de la lib graphique.

Cette fonction est appelée à chaque boucle du jeu, c'est elle qui calcule les positions des éléments et qui les insère dans le framebuffer. Cette fonction ne retourne rien.

Fonction `getFramebuffer` de `IGameModule` :

Ce getter permet de récupérer le framebuffer créé par la fonction `run()`. Cette fonction retourne un vector contenant un autre vector contenant une structure `color_t`.

Fonction `getTexts()` de `IGameModule` :

Ce getter permet de récupérer les textes générés par la fonction `run()`. Cette fonction retourne un vector de structure `text_t`.

Fonction `getSoundsStart()` de `IGameModule` :

Ce getter permet de récupérer les strings des noms des musiques en train d'être jouées ainsi que leur volume. Cette fonction retourne un vector contenant un pair d'une string et d'un float.

Fonction `getSoundsStop` de `IGameModule` :

Ce getter permet de récupérer les strings des noms de musiques qui doivent être arrêtés. Cette fonction retourne un vecteur de strings.

Fonction `getScore` de `IGameModule` :

Ce getter permet de récupérer le score. Cette fonction retourne un int.

Fonction `isOver` de `IGameModule` :

Ce getter permet de récupérer un booléen qui représente la fin du jeu. Cette fonction retourne un booléen. Si ce booléen vaut true, la fonction `run()` doit cesser d'être appelée.

Créer une nouvelle lib graphique :

Pour implémenter une nouvelle lib graphique, il est nécessaire de créer une classe qui hérite de IGraphicalModule :

```
class IGraphicalModule {
public:
    virtual ~IGraphicalModule() = default;

    /* Display */
    virtual void createWindow(int x, int y, int &ac, char **av) = 0;
    virtual void closeWindow() = 0;
    virtual void display() = 0;
    virtual void drawFramebuffer(std::vector<std::vector<color_t>>) = 0;
    virtual void drawText(text_t) = 0;
    virtual size_t getSmoothness() const = 0;

    /* Input */
    virtual char getInput() const = 0;

    /* Sound */
    virtual void stopSound(std::string sound) = 0;
    virtual void playSound(std::string sound) = 0;
    virtual void setVolume(std::string sound, float volume) = 0;
    virtual float getVolume(std::string sound) const = 0;
    virtual void stopAllSounds() = 0;
};
```

Destructeur de IGraphicalModule :

- Initialisation des variables locales

Fonction createWindow() de IGraphicalModule :

- int x : hauteur de la fenêtre
- int y : largeur de la fenêtre
- int &ac : nombre d'arguments
- char **av : arguments du programme

Cette fonction crée la window en fonction de la taille demandée.

Fonction closeWindow() de IGraphicalModule :

Cette fonction ferme la window qui a été précédemment ouverte.

Fonction display() de IGraphicalModule :

Cette fonction affiche le framebuffer et les textes définis par le jeu ou le menu.

Fonction `drawFrameBuffer()` de `IGraphicalModule` :

Cette fonction prend en paramètre un framebuffer et elle le configure pour l'afficher correctement dans la window avec la fonction `display()`.

Fonction `drawText` de `IGraphicalModule` :

Cette fonction prend en paramètre une structure `text_t` et elle configure le texte pour l'afficher dans la window.

Fonction `getSmoothness()` de `IGraphicalModule` :

Cette fonction retourne un `size_t` contenant le niveau de smoothness demandé.

Fonction `getInput()` de `IGraphicalModule` :

Cette fonction retourne la valeur ASCII de la touche du clavier qui est enfoncée.

Fonction `stopSound()` de `IGraphicalModule` :

Cette fonction arrête le son qui est actuellement joué par la fenêtre, elle prend en paramètre une string qui contient le nom du son.

Fonction `playSound()` de `IGraphicalModule` :

Cette fonction joue le son dont le nom est passé en paramètre (le paramètre est une string).

Fonction `setVolume()` de `IGraphicalModule` :

Cette fonction définit le volume d'un son. Elle prend en paramètre une string qui contient le nom du son dont le volume doit être défini, son deuxième paramètre est un float qui représente le volume à définir.

Fonction `getVolume()` de `IGraphicalModule` :

Ce getter retourne le volume actuel d'un son. Elle prend en paramètre une string qui contient le nom du son.

Fonction `stopAllSounds` de `IGraphicalModule` :

Cette fonction arrête tous les sons qui sont actuellement joués.