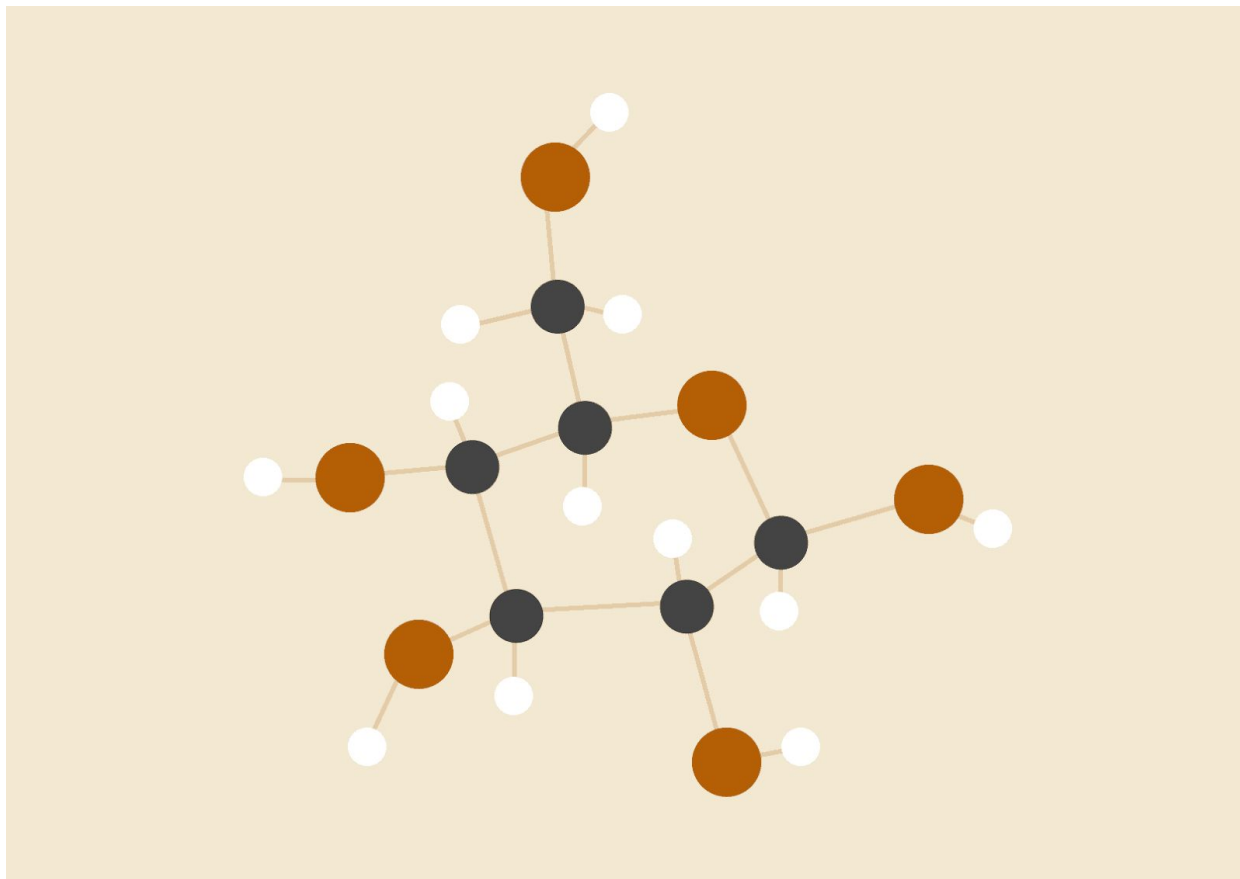


OOP_ARCADE

Documentation



Sitpi RAJENDRAN / Louis VANDENBOSSCHE / Ethan SZNAJDER

06/04/2020

{Epitech.} - Paris

INTRODUCTION

Dans le cadre de notre formation a Epitech, lors de notre deuxième année, nous avons un projet dénommé Arcade à effectuer. Ce projet en CPP, consiste à créer deux jeux, dans trois bibliothèques graphiques. Le tout avec un menu permettant de passer d'une bibliothèque à une autre.

Nous avons commencé à travailler sur la bibliothèque SDL, SFML et NCURSES. Ainsi que deux jeux : le Nibbler ainsi que le Centipèdes.

LE LAUNCHER

Il s'agit de la classe représentant le coeur du programme. Il s'agit du cerveau de notre jeu, celui qui contient toutes les informations concernant les classes à charger, les fonctions à appeler, ou encore le jeu à lancer.

void launch() :	Lance le jeu sélectionné
void quit():	Arrête le programme
void restart():	Relancer le jeu
int changeGame(Input):	Permet de changer le jeu
int changeGrp(Input):	Permet de changer la bibliothèque graphique
void menu():	Lance le menu

LE DDLOADER

Cette classe permet de charger une classe depuis

int loadLibrary(char*) :	Charge la bibliothèque
int closeLibrary():	Ferme la bibliothèque
T* getInstance(char *):	Créer une classe

ELEMENT

Il s'agit d'une classe contenant les informations d'un élément à afficher.

```
Color _color;  
Image _image;  
std::string _element;  
Position _position;
```

EXCEPTION

Cette classe est la classe Erreur, elle hérite de la classe std:exception

```
std::string _message;
```

ICOREGAME

Une classe récupérant et envoyant les informations du jeu au Core.

```
void SendInput(Input &input):  Envoie une entrée au jeu  
void restartGame():            Recommence le Jeu  
CoreGamer_t *getData();        Récupère la structure du Jeu
```

IDISPLAY

Il s'agit de la classe qui lie une bibliothèque graphique au Core du jeu.

void clear():	Efface l'affichage
void Refresh():	Rafraîchit l'affichage
void dispMap(std::vector<std::string>,Element)	Affiche la carte
void dispObject(std::vector<std::Element>)	Affiche tous les éléments
int menu(std::vector<std::string>)	Affiche le Menu
int getEvent()	Récupère les événements
std::string getName()	Récupère le nom
void setName(std::string)	Définit le nom
void getTimeDiff()	Récupère la différence de temps
void dispName(std::string)	Affiche le nom
void dispScore(int)	Affiche le score

IMAGE

Il s'agit de la classe comportant les différentes informations de l'image

std::string getFilename()	Récupère le nom d'un fichier
arcade::Position getPositionBeg()	Récupère Position début
arcade::Position getPositionEnd()	Récupère Position Fin
void setFilename(std::string)	Définit le nom du fichier
void setPositionBeg(arcade::Position)	Définit la position début
void setPositionEnd(arcade::Position)	Définit la position fin

LIB

Il s'agit de la classe comportant les différentes informations de l'image

std::vector<std::string>getGameLib() Récupère la bibliothèque du jeu

std::vector<std::string>getGraphLib() Récupère la bibliothèque graphique

POSITION

Il s'agit de la classe comportant les différentes informations concernant la position

void setPosition(int x, int y); Définit la position x et y

int getPositionX() const; Récupère la position x

int getPositionY() const; Récupère la position y

AJOUTER UN JEU

Il faut pour cela ajouter deux fonctions en extern "C" nommé comme ci-dessous afin que le Core puisse récupérer le jeu ainsi que la classe du jeu, qui doit hériter de ICoreGame.

```
extern "C" arcade::ICoreGame *entryPoint() {  
    return new arcade::Snake();  
}  
  
extern "C" void destroyGame(arcade::ICoreGame *game) {  
    delete dynamic_cast<arcade::Snake *>(game);  
}
```

AJOUTER UNE BIBLIOTHEQUE GRAPHIQUE

Il faut pour cela ajouter deux fonctions en extern “C” nommé comme ci-dessous afin que le Core puisse récupérer la librairie graphique ainsi que la classe de celle-ci, qui doit hériter de IDisplay.

```
extern "C" arcade::IDisplay *entryPoint() {  
    return new lib::Ncurses();  
}  
  
extern "C" void destroy(arcade::IDisplay *lib) {  
    delete lib;  
}
```