PROJET imagiNABle



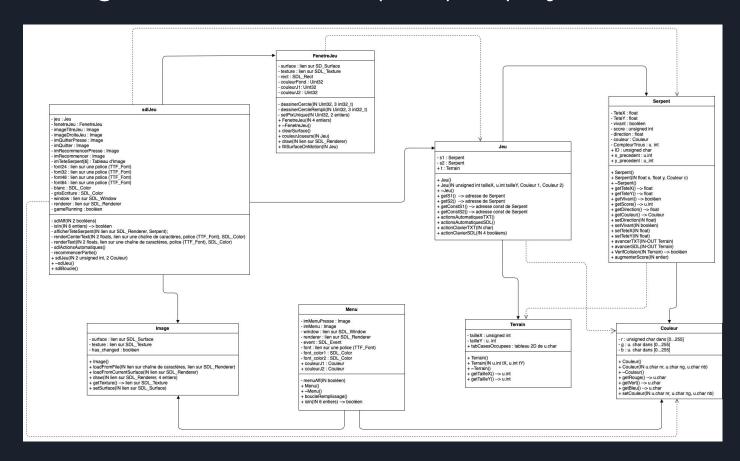
CURVE FEVER, le jeu de serpent original jouable en :

- Version graphique (fenêtre SDL2)

UE LIFAP4 Semestre de printemps 2021

BRIGUET Benoît LETO Aymeric PURICELLI Nathan

Diagramme des classes (UML) du projet



CLASSE Serpent

Gérer un serpent et ses informations :

- Couleur
- Score
- Déplacement
- Collisions

Base du projet, utilisée par la classe Jeu par exemple (voir slide suivante)

```
@author : Aymeric Leto, Benoît Briguet, Nathan Puricelli
18 using namespace std;
20 class Serpent
        @brief La position de la tête du serpent, TeteX dans la hauteur et TeteY dans la largeur du terrain de jeu.
        float TeteX, TeteY;
        @brief Booléens pour indiquer si le serpent est toujours en vie
        bool vivant;
        @brief Le score du joueur du serpent.
        unsigned int score;
        @brief La direction du serpent
        float direction:
        @brief La couleur du serpent
        @briet La couleur du serpent
        Tioat direction;
```

@brief Module gérant les serpents.

CLASSE Jeu

Gérer les données du jeu :

- Terrain
- Serpents
- Accesseurs, mutateurs...

Utilisée pour l'affichage graphique SDL2, par la classe FenetreJeu par exemple (voir slide suivante)

```
Obrief Module gérant les informations du jeu
    @author : Aymeric Leto, Benoît Briguet, Nathan Puricelli
 8 @date : Mars 2021
11 #ifndef JEU_H
12 #define JEU H
14 #include "Serpent.h"
#include "Terrain.h"
16 class Jeu
18 private:
          Obrief Permier serpent de la partie (celui du joueur 1)
          Serpent s1:
          Obrief Second serpent de la partie (celui du joueur 2).
          Serpent s2;
30 public :
          Obrief Terrain de leu de la partie.
          Terrain t:
          Obrief Constructeur de la classe, lance une partie avec 2 serpents et 1 terrain de jeu.
          Jeu();
          Obzief Constructeur de la classe, lance une parlie avec 2 serpents et 1 terrain de jeu. Obzief Constructeur de la classe, lance nue battie avec 5 serbeuts et 1 terrain de jen.
```

CLASSE FenetreJeu

Gérer la fenêtre, l'image sdl du plateau de jeu pendant la partie

Base de la version graphique SDL2, utilisée par la classe sdlJeu par exemple (voir slide suivante)

```
16 #include <SDL2/SDL ttf.h>
   #include <SDL2/SDL image.h>
22 private:
       SDL Surface* surface:
       SDL Texture* texture;
       SDL Rect rect:
       int origin_x, origin_y;
       Uint32 couleurFond;
       Uint32 couleurJ1;
       Uint32 couleurJ2:
       void dessinerCercle(Uint32 color, int32_t centreX, int32_t centreY, int32_t
      void dessinerCercleRempli(Uint32 color, int32 t centreX, int32 t centreY,
       void setPixUnique(Uint32 color, int x, int y);
       FenetreJeu(int x, int v, int w = 1122, int h = 630); //x et v coordonnées en
       ~FenetreJeu();
       void couleurJoueurs(const Jeu &j); // doit etre appelée avant de lancer la
      void draw(SDL_Renderer* renderer);
      void fillSurfaceOnMotion(const Jeu &i);
      void fillSurfaceOnMotion(const Jeu &j);
      void draw(SDL_Renderer* renderer);
      void couleurJoueurs(const Jeu &j); // doit etre appelée avant de lancer la
```



CLASSE sdlJeu

Coeur de la version graphique du jeu, sdlJeu :

- Contient les images utilisées, les couleurs, les polices d'écriture...
- Fonctionne autour d'une boucle qui récupère les actions des joueurs et appelle les fonctions de calculs et d'affichage.

```
#include <SDL2/SDL_ttf.h>
 #include <SDL2/SDL image.h>
class sdlJeu
public:
    sdlJeu(unsigned int tailleX, unsigned int tailleY, Couleur couleur1, Couleur
    ~sdlJeu();
    void sdlBoucle();
private:
    bool gameRunning;
    FenetreJeu fenetreJeu;
    Image imageTitreJeu:
    Image imageDroiteJeu;
    Image imQuitterPresse:
    Image imQuitter;
     Image imRecommencerPresse:
    Image imRecommencer;
    Image imTeteSerpent[6];
    TTF Font * font24;
    TTF Font * font32:
    TTF Font * font48;
    TTF_Font * font64;
    SDL Color blanc:
    SDL_Color grisEcriture;
    SDL Window * window:
    SDL_Renderer * renderer;
    void sdlAff(bool boutonRecommencer, bool boutonQuitter);
    bool isIn(int x, int y, int w, int h, int souris_x, int souris_y);
    void afficherTeteSerpent(SDL_Renderer* renderer, Serpent S);
    void renderCenterText(float p_x, float p_y, const char* p_text, TTF_Font* font,
         SDL_Color textColor);
    void renderText(float p_x, float p_y, const char* p_text, TTF_Font* font,
    void surfaceToTexture();
    void surfaceToTexture();
         SDL_Color textColor);
    void renderText(float p_x, float p_y, const char* p_text, TTF_Font* font,
    void renderCenterText(float p_x, float p_y, const char* p_text, TTF_Font* font,
    void afficherTeteSerpent(SDL_Renderer* renderer, Serpent S);
    bool isIn(int x, int y, int w, int h, int souris_x, int souris_y);
    void sdiAff(bool boutonRecommencer, bool boutonQuitter);
```

CONCLUSION

PROJET imagiNABle

Jeu fluide, jouable à 2 en local

Ajout d'un menu par rapport au cahier des charges

Bonus manquants

DIFFICULTÉ RENCONTRÉE:

Passage de le version TXT à la version SDL

AVEC PLUS DE TEMPS...:

- Implémentation des bonus
- Amélioration graphique
- Ajout de son
- Mise en ligne du jeu

Place à la démonstration!

