# PROJET SUIKA-GAME

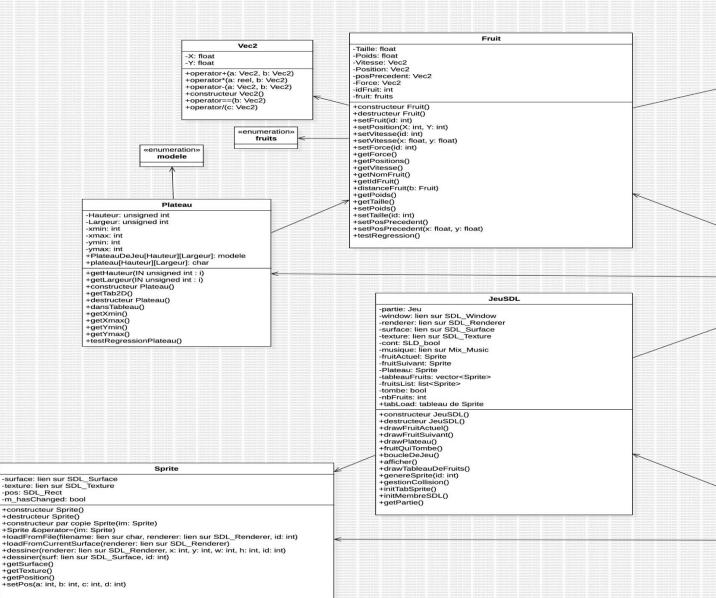


Ba Cheikh Bah Mamadou Yrius Marc

## PRINCIPE DU JEU







-capacite: unsigned int -taille\_utilisee: unsigned int -tab: Tableau de lien +aiouterElement(F: Lien) +constructeur Vector() +vider() +supprimerElement() +insererElement() +destructeur TableauDynamique() Jeu -Tableau\_fruit: Tableau Dynamique Fruit -Score: unsigned int Fruit suivant: Fruit -Fruit\_actuel: Fruit -p: Plateau -score1: unsigned int -score2: unsigned int -score3: unsigned int -origine: Vec2 +fusionner\_fruit(Tableau\_fruit: Vector) +est\_sortie(F: Fruit) +collision\_fruit(F: Fruit, E: Fruit) +increment\_score() +constructeur Jeu() +destructeur Jeu() +addFruit() +getFruitActuel() +getFruitSuivant() +getScore() +getTableauDeFruit() +generationFruit() +fruitQuiTombe() +sauvegarder() +recuperer() +FruitAddForce(F: Fruit) +collision\_Mur(F: Fruit)
+collision\_Entre\_Fruit(F1: Fruit, F2: Fruit) +maj\_vitesse\_position\_fruit(F: Fruit, G: float) +physique\_fruit() +setPosition(X: float, Y: float, a: int) +getPosition(a: int) +getTaille() +getVitesse(a: int) +getIdFruit(a: int) +getXmin() +getYmin() +getXmax() +getYmax() +getIdFruitActuel() +getIdFruitSuivant() +norme(A: Vec2) +genereFruit(id: int) +setPosition(x: float, y: float) +finDeJeu()

MenuSDL

-gFont: lien sur TTF\_Font

-score1: lien sur TTF Font

-score2: lien sur TTF\_Font

-score3: lien sur TTF\_Font

-posBouton: SDL\_Rect

-posArrPlan: SDL\_Rect

+constructeur MenuSDL()

+isFileEmpty(filename: string)

+destructeur MenuSDL()

-continu: SDL Bool

button: Sprite

+boucleMenu() +setBoutons() +afficher()

-cont: Sprite

-score: Sprite

-texture: lien sur SDL\_Texture

-surface: lien sur SDL\_Surface

window: lien sur SDL Window

-renderer: lien sur SDL Renderer

# -Partie: Jeu -plat: Plateau +afficherLeJeu() +initFruitActuel() +mise\_a\_jour\_position() +afficherScore() +finDeJeu()

**JeuModeConsole** 

## **MENU SDL**

```
oid MenuSDL::boucleMenu()
  int x,y;
   SDL Event event;
      while(SDL_PollEvent(&event))
           switch(event.type)
               case SDL QUIT:
                   continu = SDL_FALSE;
                   break;
               case SDL_MOUSEBUTTONUP:
               x = event.motion.x;
               y = event.motion.y;
                   bool fichier = fichierVide("./data/save.txt");
                   if (((x >= 120 \&\& x <= 600))
                               && (y \ge 263 \&\& y \le 333))
                           fermerMenu();
                           JeuSDL j;
                           j.boucleDeJeu();
                   if ((x >= 120 \&\& x <= 600)
                               && (y >= 366 && y <= 433) && !fichier)
                       fermerMenu();
                       JeuSDL j;
                       j.getPartie().recuperer();
                       j.boucleDeJeu();
               break;
      afficher();
      SDL RenderPresent(renderer);
```

```
bool fichier vide = fichierVide("./data/save.txt");
if(fichier vide)
    cont.loadFromFile("./data/menu/CONTINUE_f.png",renderer,2);
else
   cont.loadFromFile("./data/menu/CONTINUE.png",renderer,2);
cont.setPos(100,350,TAILLE SPRITE, TAILLE SPRITE);
score.loadFromFile("./data/menu/score.png",renderer,2);
score.setPos(10,10,TAILLE_SPRITE, TAILLE_SPRITE);
font = TTF_OpenFont("./data/utile/arial.ttf",25);
if(font==NULL) TTF_CloseFont(font);
    score1.setPos(85,49,30,30);
    score2.setPos(85,121,30,30);
    score3.setPos(85,190,30,30);
boucleMenu();
```

### **CLASSE GESTION DU JEU**

```
void Jeu::sauvegarder()
   string filename = "./data/save.txt";
   ofstream fichier(filename.c str());
   assert(fichier.is open());
   fichier << score << endl;
   fichier << fruit actuel.getIdFruit() << endl;</pre>
   fichier << fruit_suivant.getIdFruit() << endl;</pre>
   fichier << tableau_fruit.size() << endl;</pre>
   for (int i = 0; i < (int)tableau fruit.size(); i++)</pre>
       fichier << tableau fruit[i].getIdFruit() << endl;</pre>
       fichier << tableau fruit[i].getPosition().x << ' ' << tableau fruit[i].getPosition().y << endl;</pre>
void Jeu::recuperer()
   string filename = "./data/save.txt";
   ifstream fichier(filename.c_str());
   int fa, fs, size;
   fichier >> score >> fa >> fs >> size;
   fruit_actuel.setFruit(fa);
   fruit_suivant.setFruit(fs);
   int Id;
   float x, y;
   for (int i = 0; i < size; i++)
       fichier >> Id >> x >> y;
       tableau_fruit[i].setFruit(Id);
       tableau_fruit[i].setPosition(x, y);
```

```
void Jeu::fusionner_fruit()
   vector<Fruit> nouveaux_fruits;
   // Parcourir le tableau de fruits
   for (int i = 0; i < (int)tableau fruit.size(); ++i)</pre>
       // Parcourir à partir du fruit suivant
       for (int j = i + 1; j < (int)tableau_fruit.size(); ++j)</pre>
           // Vérifier si les fruits ont le même identifiant et si cet identifiant est inférieur à 11
           if (tableau_fruit[i].getIdFruit() == tableau_fruit[j].getIdFruit() && tableau_fruit[i].getIdFruit() < 11)</pre>
                // Calculer la position moyenne des deux fruits
                Vec2 temp = (tableau_fruit[i].getPosition() + tableau_fruit[j].getPosition()) / 2.0;
                // Créer un nouveau fruit avec l'identifiant suivant
               Fruit nouveau_fruit = genereFruit(tableau_fruit[i].getIdFruit() + 1);
                // Définir la position du nouveau fruit
                nouveau_fruit.setPosition(round(temp.x), round(temp.y));
                nouveaux fruits.push back(nouveau fruit);
                // Incrémenter le score
                increment score(tableau fruit[i].getIdFruit() * tableau fruit[j].getIdFruit());
                // Sortir de la boucle intérieure après avoir trouvé une paire de fruits à fusionner
                break;
    // Effacer le tableau de fruits actuel
   tableau_fruit.clear();
    // Ajouter les nouveaux fruits au tableau de fruits
   for (int i = 0; i < (int)nouveaux fruits.size(); ++i)</pre>
       tableau fruit.push back(nouveaux fruits[i]);
```

#### **JEU SDL & SPRITE**

```
Sprite::Sprite(const Sprite &im)
    *this = im;
Sprite &Sprite::operator=(const Sprite &im)
   surface = im.surface;
   texture = im.texture;
   return *this;
void Sprite::drawScore(SDL_Renderer *renderer,TTF_Font * font,SDL_Color color,int score)
   char buffer[20] = {'\0'};
   citoa(score,buffer,10);
   surface = TTF_RenderText_Solid(font,buffer,color);
   texture = SDL_CreateTextureFromSurface(renderer, surface);
   SDL_FreeSurface(surface);
   SDL_RenderCopy(renderer,texture,NULL,&pos);
```

```
void JeuSDL::initTabSprite()
   Plateau.loadFromFile("./data/utile/plateau suika.png", renderer, 0);
   Plateau.setPos(157, 401, TAILLE SPRITE, TAILLE SPRITE);
   tabLoad[0].loadFromFile("./data/fruits/circle0.png", renderer, 1);
   tabLoad[1].loadFromFile("./data/fruits/circle1.png", renderer, 1);
    tabLoad[2].loadFromFile("./data/fruits/circle2.png", renderer, 1);
   tabLoad[3].loadFromFile("./data/fruits/circle3.png", renderer, 1);
   tabLoad[4].loadFromFile("./data/fruits/circle4.png", renderer, 1);
   tabLoad[5].loadFromFile("./data/fruits/circle5.png", renderer, 1);
   tabLoad[6].loadFromFile("./data/fruits/circle6.png", renderer, 1);
   tabLoad[7].loadFromFile("./data/fruits/circle7.png", renderer, 1);
   tabLoad[8].loadFromFile("./data/fruits/circle8.png", renderer, 1);
   tabLoad[9].loadFromFile("./data/fruits/circle9.png", renderer, 1);
   tabLoad[10].loadFromFile("./data/fruits/circle10.png", renderer, 1);
// fabrique tout ce qui est necessaire pour le fruit actuel
void JeuSDL::drawFruitActuel()
   fruitActuel.setPos(partie.getPositionFruitActuel().x,partie.getPositionFruitActuel().y
                                ,TAILLE SPRITE, TAILLE SPRITE);
   SDL_RenderCopy(renderer, tabLoad[partie.getIdFruitActuel()-1].getTexture(), nullptr, &fruitActuel.getPosition());
void JeuSDL::drawFruitSuivant()
   fruitActuel.setPos(fruitSuivant.getPosition().x,fruitSuivant.getPosition().y
                                ,TAILLE SPRITE, TAILLE SPRITE);
   SDL RenderCopy(renderer, tabLoad[partie.getIdFruitSuivant()-1].getTexture(), nullptr, &fruitSuivant.getPosition());
```

CE QUE NOUS AVONS RETENU

CE QUE NOUS AVONS APPRIS

CE QUI NOUS A PLU DANS LE PROJET



MERCI DE VOTRE ATTENTION