



V
S



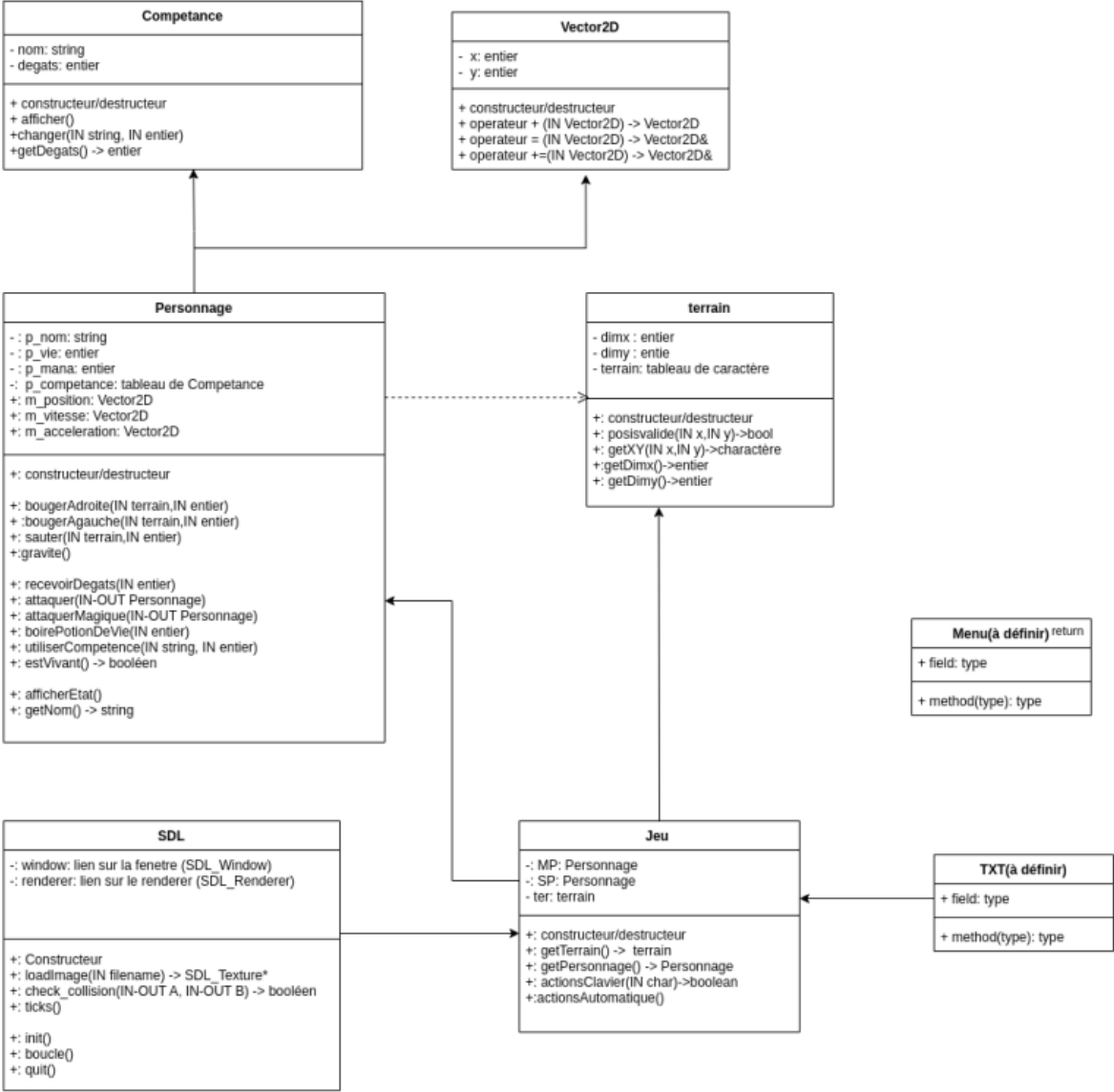
Anime Fighter

ZEMMOURI Adnane p1812386

ELFILALI Ayoub p1811588

HASSAN BABIKER Wail p1802896

Diagramme UML :



CLASSE:

1. Jeu
2. Personnage
3. JeuSDL

```
11 class Jeu{
12     private:
13
14
15
16     public:
17
18
19     Personnage MP;
20     Personnage SP;
21     terrain ter1;
22     Tuiles tuile[24];
23     Menu menu;
24
25
26     void actionsAutomatique();//evenements automatiques
27
28     void MPclavierDown(int touche); //gestion des evenements liés au clavié
29     void MPclavierUp(int touche);
30
31     void SPclavierDown(int touche);
32     void SPclavierUp(int touche);
33
34     void actionsMenu(int clic); //gestion des événements liés à la souris
35
36     //fct concernant le deroulement de la partie
37     void initPartie();
38     void updatePartie(float deltaTime);
39     void quitPartie();
40
41     void init();//initialisation du jeu
42     void update(); //update
43     void quit(); //terminaison
44
45 };
16 class JeuSDL{
17     private:
18         SDL_Window *window;
19         SDL_Renderer *renderer;
20
21         Jeu J;
22
23
24
25     public:
26         JeuSDL();
27
28
29         SDL_Texture* loadImage(const char* filename);
30
31         void KeyboardEventsLuffy(bool &isSprite);
32         void KeyboardEventsZoro(bool &isSprite);
33         void clavierUP(SDL_Event events);
34
35         void init();
36         void boucleAcceuil();
37         void quit();
38
39
40
41 };
18
```

```
11 class Personnage
12 {
13     public:
14
15         //constructeurs et destructeur
16         Personnage(); //constructeur
17         ~Personnage(); //destructeur
18
19
20         //mouvements
21         void bougerAdroite(float xd,terrain& t);
22         void bougerAgauche(float xg,terrain& t);
23         void sauter(float yh,terrain& t);
24         void sauterAdroite(float& t);
25         void sauterAgauche();
26
27         //Attaques et Compétences
28         void recevoirDegats(int nbDegats);
29         void attaquer(Personnage& cible);
30         void attaqueUltime(Personnage& cible);
31
32         //Vie et etat
33         void boirePotionDeVie(int quantitePotion);
34         bool estVivant() const;
35         void afficherEtat() const;
36         int getVie();
37         void setVie(int vie);
38
39         //test Regression
40         void testRegression();
41         //bool isColl(const Personnage &MP, const Personnage &SP);
42
43
44         Physic phy;
45         Texture tex;
46
47     private:
48
49         int m_vie;
50         int m_mania;
51         Competence *m_competence;
52
53
54 };
```

Illustration de la fenêtre SDL du jeu :

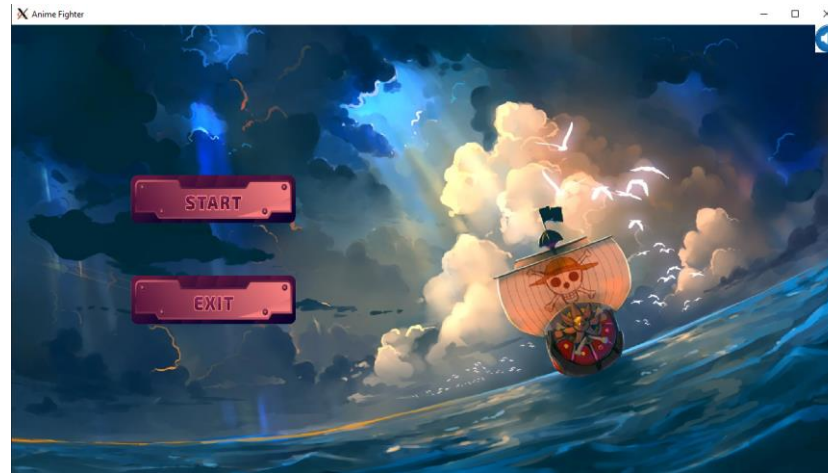
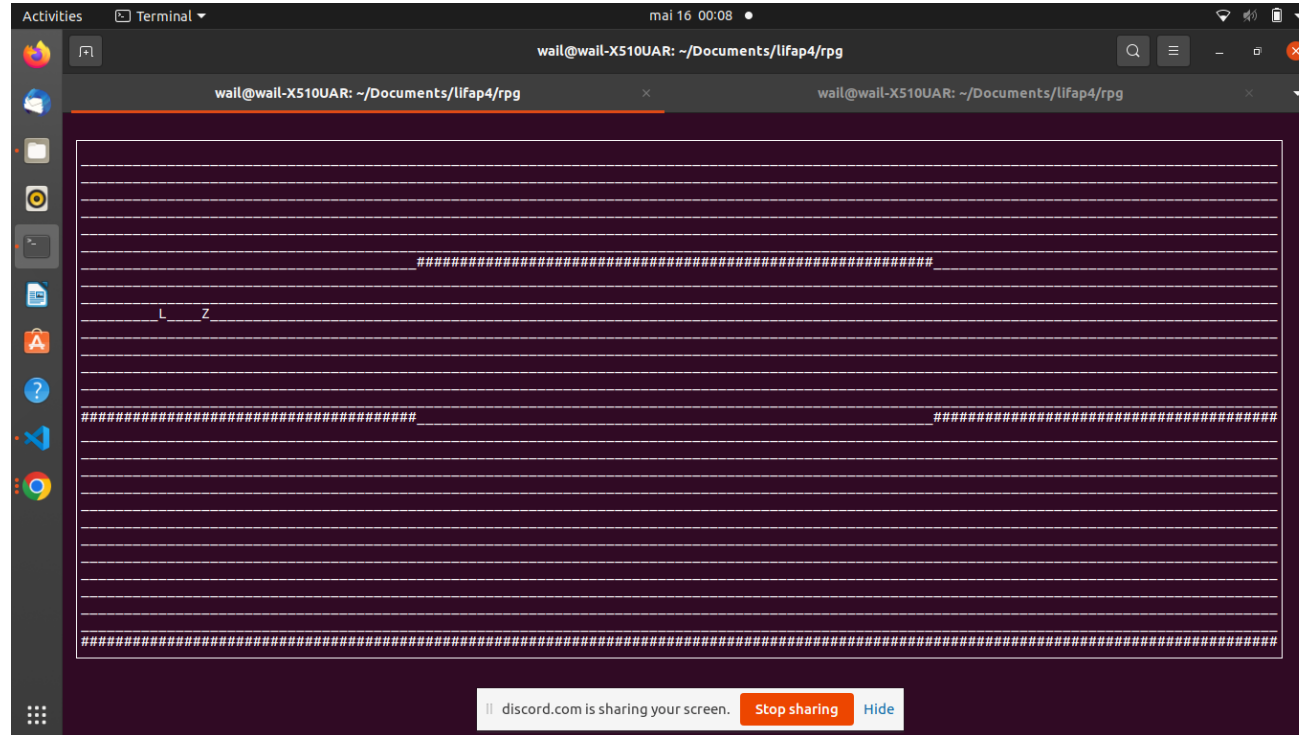


Illustration de la fenêtre du jeu en version texte :



« L » et « Z » désignent les 2 joueurs, le « # » les plateformes.