

L'Antagoniste

Auteurs :

DJESSY-ALBERTO KITA

COME KOUAME

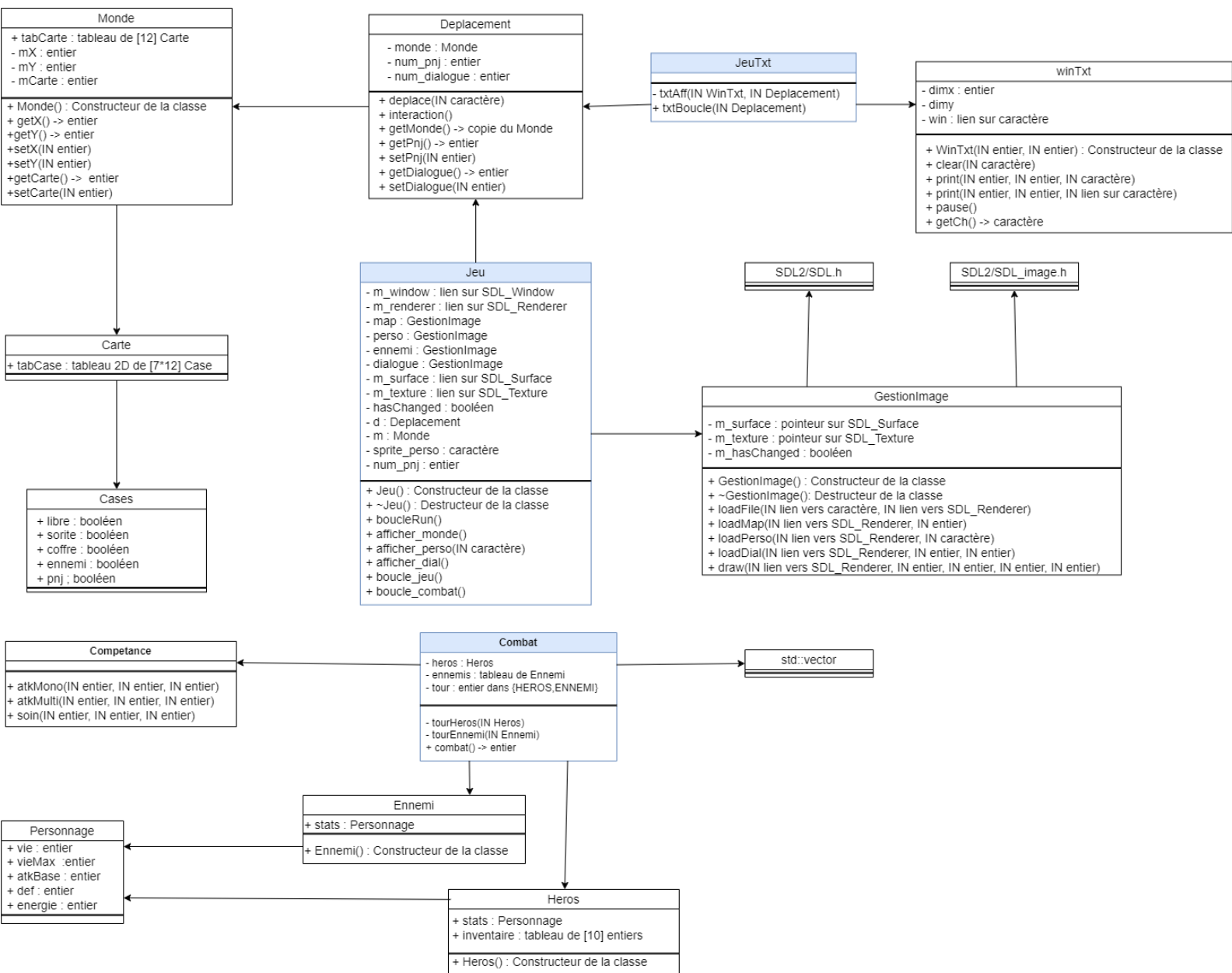
LUCAS DUCHAMP

Présentation :

Notre projet est un jeu RPG avec des combats en tour par tour. Il se déplace sur une carte découpée en cases avec des caractéristiques différentes.



Diagramme des classes UML :



Classe Déplacement : On gère ici les déplacements du joueur et tout ce qui est relatif à sa position c'est-à-dire l'ouverture de coffre, initier un dialogue avec un PNJ ainsi que lancer un combat

```
21 void Deplacement::interaction() {
22     int pos_x=monde.getX();
23     int pos_y=monde.getY();
24     int carteX=monde.getCarte();
25     int newx=pos_x;
26     int newy=pos_y;
27     int pos_s;
28     for (int i=-1;i<2;i=i+2) {
29         if ( (monde.tabCarte[carteX].tabCase[newx=pos_x+i][newy=pos_y].pnj) || (monde.tabCarte[carteX].tabCase[newx=pos_x][newy=pos_y+i].pnj) ) {
30             pos_s=newx+newy*100;
31             switch(carteX)
32             {
33                 case 1 : if (pos_s==5) { // pos_x=5,pos_y=0 -> 5;
34                     num_pnj=11;
35                     if (num_dialogue==1000) {num_dialogue=0;}
36                     else {num_dialogue++;}
37                     break;
38                 } else if (pos_s==704) { // pos_x=4,pos_y=7 -> 704;
39                     num_pnj=12;
40                     if (num_dialogue==1000) {num_dialogue=0;}
41                     else {num_dialogue++;}
42                 }
43             }
44             break;
45         }
46     }
47 }
```

Classe GestionImage : On gère ici l'initialisation et l'affichage de toutes les images ainsi que déterminé quelle image affichée selon l'action par exemple lancé le dialogue numéro 0 du 1er PNJ de la carte 5 ensuite le numéro 1 puis 2.

```
113 void GestionImage::draw(SDL_Renderer *renderer, int x, int y, int w = -1, int h = -1)
114 {
115     SDL_Rect pos;
116     pos.x = x;
117     pos.y = y;
118     pos.w = (w < 0) ? m_surface->w : w;
119     pos.h = (h < 0) ? m_surface->h : h;
120     if (m_hasChanged)
121     {
122         SDL_UpdateTexture(m_texture, nullptr, m_surface->pixels, m_surface->pitch);
123         m_hasChanged = false;
124     }
125     SDL_RenderCopy(renderer, m_texture, nullptr, &pos);
126 }
127
```

Classe Combat :

```
void Combat::combat(int action){
    int total;
    cout<<"Il y a "<<ennemis.size()<<" ennemis"<<endl;
    for(long unsigned int j=0;j<ennemis.size();j++){
        cout<<"Ennemi "<<j+1<<" : "<<ennemis[j].stats.vie<<endl;
    }
    if(tour == MOI){
        cout<<"Votre tour"<<endl;
        menu(action);
    }
    else{
        cout<<"Tour ennemi"<<endl;
        for(long unsigned int y=0;y<ennemis.size();y++){
            if(garde) heros.stats.vie -= ennemis[y].stats.atkBase*0.75;
            else heros.stats.vie -= ennemis[y].stats.atkBase;
        }
        cout<<"Il te reste "<<heros.stats.vie<<" PV"<<endl;
    }

    if(tour == MOI && est_valide) tour = ENNEMI;
    else{
        tour = MOI;
        est_valide = false;
    }
}
```

```
void Combat::selecteurEnnemi(long unsigned int pos, vector<Ennemi> &ennemis){
    Competence comp;
    switch (pos)
    {
        case 1:
            if(etat == COMP){
                comp.atkMono(ennemis[0].stats.vie,heros.stats.energie,heros.stats.atkBase);
                cout<<"L'ennemi 1 a "<<ennemis[0].stats.vie<<" pv"<<endl;
            }
            else{
                ennemis[0].stats.vie -=30;
                cout<<"L'ennemi 1 a "<<ennemis[0].stats.vie<<" pv"<<endl;
            }
            est_valide = true;
            break;
        case 2:
            if(pos>ennemis.size()){
                cout<<"Choisissez un autre ennemi";
            }
            else{
                if(etat == COMP){
                    comp.atkMono(ennemis[1].stats.vie,heros.stats.energie,heros.stats.atkBase);
                    cout<<"L'ennemi 2 a "<<ennemis[1].stats.vie<<" pv"<<endl;
                }
                else{
                    ennemis[1].stats.vie -=10;
                    cout<<"L'ennemi 2 a "<<ennemis[1].stats.vie<<" pv"<<endl;
                }
                est_valide = true;
                break;
            }
            break;
    }
}
```

Diagramme de Gantt :

Taches/Dates	05/03	12/03	19/03	26/03	02/04	09/04	16/04
Personnage			x				
Heros				x			
Ennemi				x			
Case	x						
Carte	x						
Monde	x	x					
Menu					x	x	
Inventaire						x	
Combat					x	x	x
Competence					x	x	
GestionImage			x	x	x		
Deplacement		x	x		x	x	
Jeu		x	x		x		
WinTxt	x	x					

Conclusion :

Quel difficultés rencontrées :

- Lier les modules entre eux
- Affichage avec SDL
- Fuite de mémoire
- Le combat

Qu'est-ce qu'on aurait fait avec plus de temps :

- Format du monde sans cases
- Finaliser et intégrer le combat au jeu
- Compléter l'affichage du mode texte et graphique pour le combat