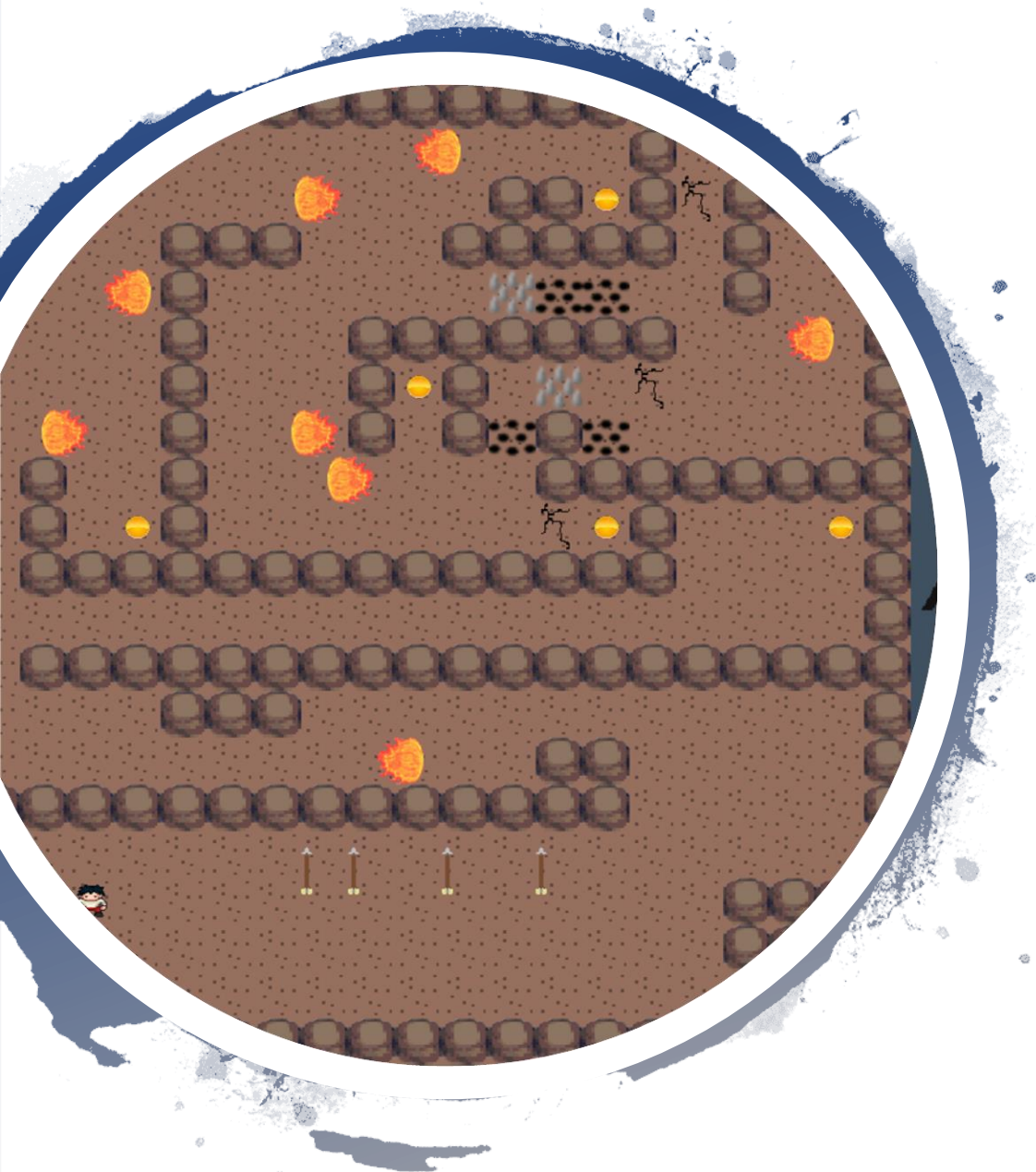


Student Struggle

Incarnez un étudiant en informatique dans son parcours du combattant pour obtenir la licence et devenir un développeur confirmé!

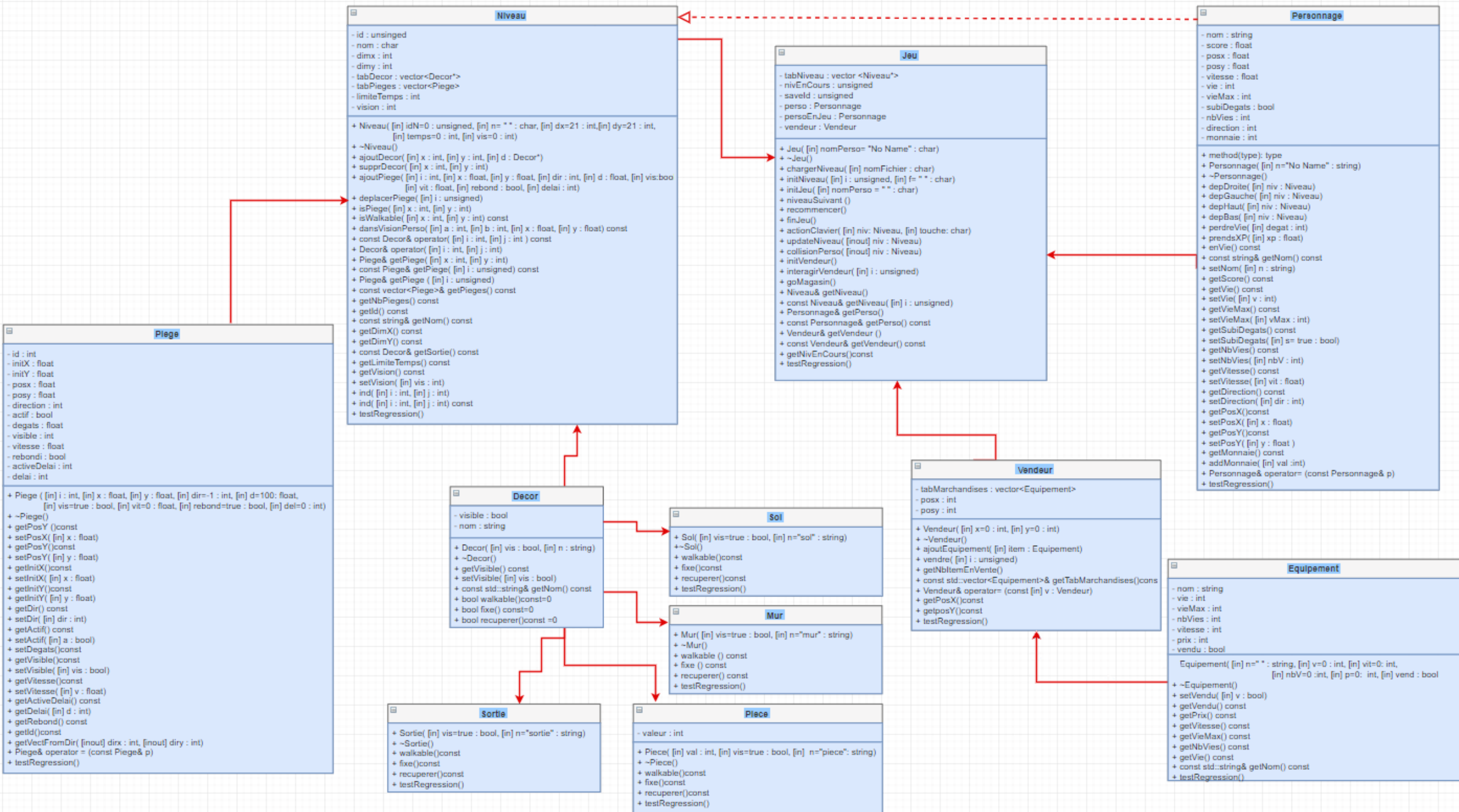
Vous devrez réussir les matières avec succès en franchissant les obstacles avec un minimum de dégâts, réussir à vous repérer dans un labyrinthe dans une obscurité quasi-totale, et ceci sans dépasser le temps imparti! Bien sûr, plus les dégâts sont importants moins votre score sera élevé...

Heureusement que le magasin est là pour vous proposer des équipements qui amélioreront vos caractéristiques et vous permettront de réussir!



Réalisé par :

- Goupy Gaspard
- Ben Abdallah Nour
- Brignone Jean
- Derdaki Akram



SDLjeu

```

- win : SDL_Window *
- ren : SDL_Renderer *
- police : TTF_Font *
- jeu : Jeu
- menu : Menu *
- shop : Shop *
- img_Perso : Image
- img_Sol : Image
- img_Piege : Image
- img_Mur : Image
- img_Sortie : Image
- img_Piece : Image
- img_Vendeur : Image
- img_HUD : Image
- son_piege : Mix_Chunk*
- son_piques : Mix_Chunk*
- son_feu : Mix_Chunk*
- son_sortie : Mix_Chunk*
- son_gauche : Mix_Chunk*
- son_droite : Mix_Chunk*
- son_fleche : Mix_Chunk*
- son_jeu : Mix_Chunk*
- son_menu : Mix_Chunk*
- AvecSon : bool
- sonJeuEnCours : bool
- sonMenuEnCours : bool
- winx : int
- winy : int
- centreurX : int
- centreurY : int
- TAILLE_SPRITE : int
- TAILLE_BORDURE : int
- framePerso : int
- perso_a_bouge : bool
- FPS_perso : Timer
- FPS_pieges : Timer
- horloge : Timer
- scoreJeu : Timer

+ SDLjeu( [in] nomPerso : char)
+ ~SDLjeu()
+ chargeTexte ()
+ chargeImages ()
+ SDL_exit ( [in] error : char)
+ jouerSon ( [in] s : Mix_Chunk, [in] channel : int, [in] loop : int)
+ overlap ( [in] box1 : SDL_Rect, [in] box2 : SDL_Rect, [in] img : Image)
+ isWalkable ( [in] niv : Niveau, [in] img : Image, [in] perso_x : int, [in] perso_y : int)
+ collisions ( [inout] niv : Niveau, [inout] perso : Personnage)
+ actionClavierPixel ( [inout] perso : Personnage, [in] niv : Niveau, [in] touche : char)
+ deplacerPieges ( [inout] niv : Niveau)
+ ecrireTexte ( [in] x : int, [in] y : int, [in] img : char, [in] bd : char, [in] r : int, [in] v : int, [in] b : int)
+ champsVisionPerso ( [in] perso : Personnage, [in] visionPerso : int)
+ afficherHUD ( [in] perso : Personnage)
+ afficherTempsRestant ( [in] niv : Niveau)
+ afficherPieges ( [in] niv : Niveau)
+ afficher ( [in] niv : Niveau, [in] perso : Personnage)
+ jouer()
+ finJeu()
+ score()
+ lireSoundTrack()

```

Image

```

- surface : SDL_Surface *
- texture : SDL_Texture *

+ Image()
+ ~Image()
+ chargerFichier ( [in] ren : SDL_Renderer *, [in] fichier : char*, [in] height : int, [in] weight : -1 : int)
+ charger ( [in] ren : SDL_Renderer *)
+ afficher ( [in] ren : SDL_Renderer *, [in] x : int, [in] y : int)
+ SDL_Texture* getTexture ()const
+ setSurface ( [in] surf : SDL_Surface *)
+ getW()const
+ setW( [in] weight : int)
+ getH()const
+ setH( [in] height : int)

```

Timer

```

- startTicks : int
- pausedTicks : int
- paused : bool
- started : bool

+ Timer()
+ start()
+ stop()
+ pause()
+ unpause()
+ get_ticks()
+ is_started ()
+ is_paused ()
+ get_s()

```

Shop

```

- ren : SDL_Renderer*
- winx : int
- winy : int
- TAILLE_SPRITE : int
- actif : bool
- img_Fond : Image
- img_Items : Image
- vendu : int
- bob : Vendeur
- jeu : Jeu*
- nbItems : int
- rect_item : SDL_Rect*
- rect_fond : SDL_Rect

+ Shop( [in] j : Jeu, [in] r : SDL_Renderer*, [in] x : int, [in] y : int, [in] t : int, [in] a : bool)
+ ~Shop()
+ vendre ( [in] x : int, [in] y : int)
+ afficher()
+ dansMagasin()
+ ouvrir()
+ getActif()const
+ setActif( [in] a : bool)
+ testRegression()

```

Menu

```

- menuPrincipal : bool
- option : bool
- son : bool
- TAILLE_SPRITE : int
- TAILLE_BORDURE : int
- ren : SDL_Renderer *
- img_MenuBoutons : Image
- img_FinJeuBoutons : Image
- img_options : Image
- img_credits : Image
- img_son : Image
- img_retour : Image
- img_commandes : Image
- hover0 : int
- hover1 : int
- hover2 : int

+ Menu ( [in] r : SDL_renderer*, [in] T_B : int, [in] T_S : int)
+ ~Menu()
+ chargeImage()
+ afficherMenu()
+ afficherHover( [in] sourisX : int, [in] sourisY : int)
+ traitementClic ( [inout] niv : Niveau, [in] clicX : int, [in] clicY : int, [inout] stop : bool)
+ traitementClicFinJeu ( [in] clicX : int, [in] clicY : int, [inout] stop : bool)
+ getSon()const
+ getOption()const
+ setMenuPrincipale( [in] m : bool )

```

TXTjeu

```

+ jouer( [inout] j : Jeu)

```

TXTwin

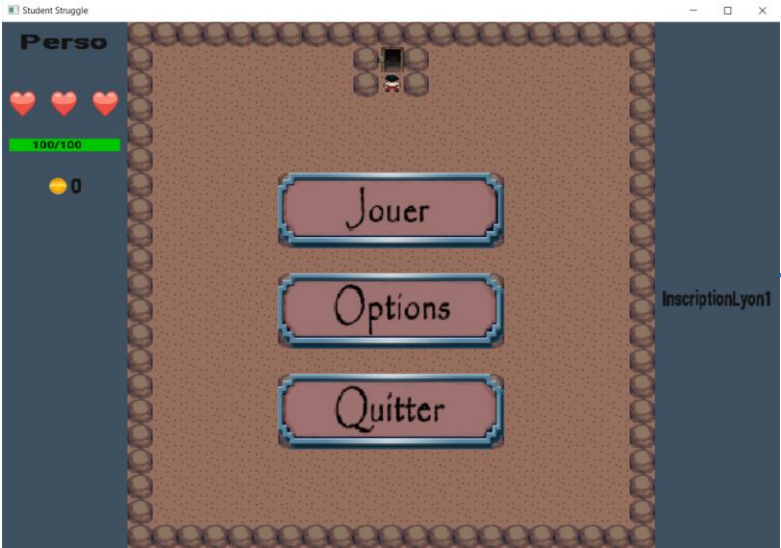
```

- dimx : int
- dimy : int
- win : char*

+ TXTwin( [in] dx : int, [in] dy : int)
+ ~TXTwin()
+ clear()
+ print( [in] x : int, [in] y : int, [in] c : char)
+ print( [in] x : int, [in] y : int, [in] c : char)
+ draw()
+ pause()
+ getCh()

```

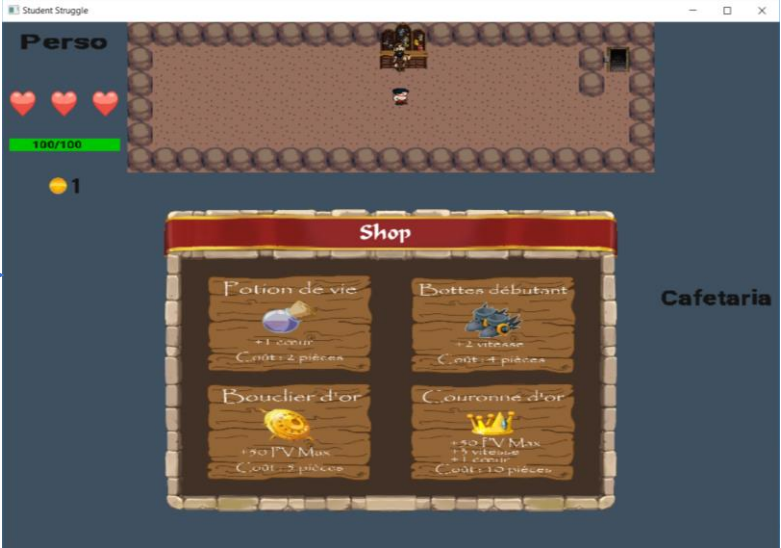

Niveau 1



Niveau 2



Niveau 0



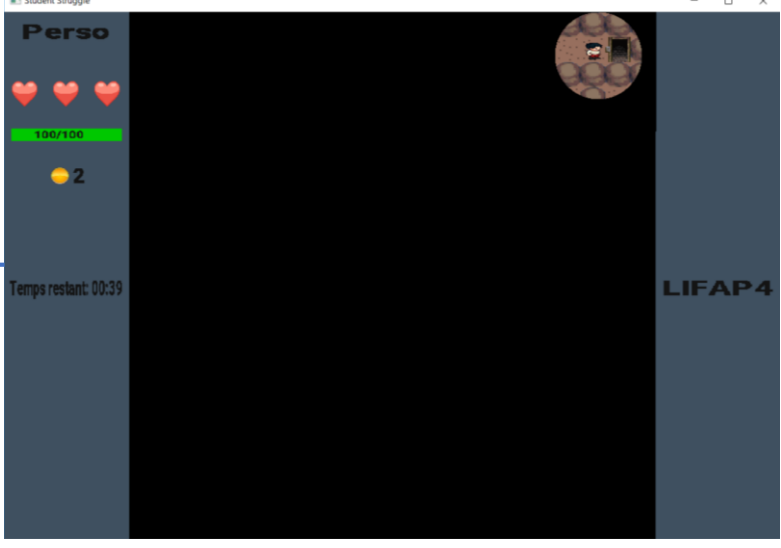
Niveau 4



Niveau 0



Niveau 3



Gestion des collisions

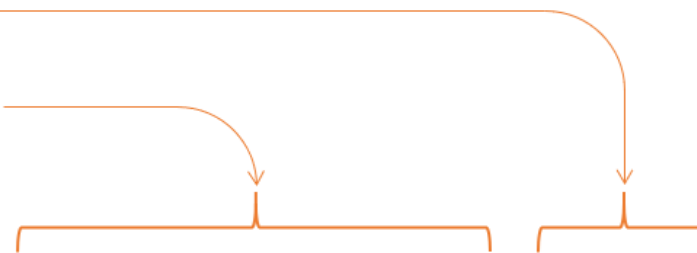
- overlap : true si 2 rectangles se superposent
- isWalkable : true si un sprite peut se déplacer à une position dans un niveau
- collisions : collisions du personnage avec le niveau



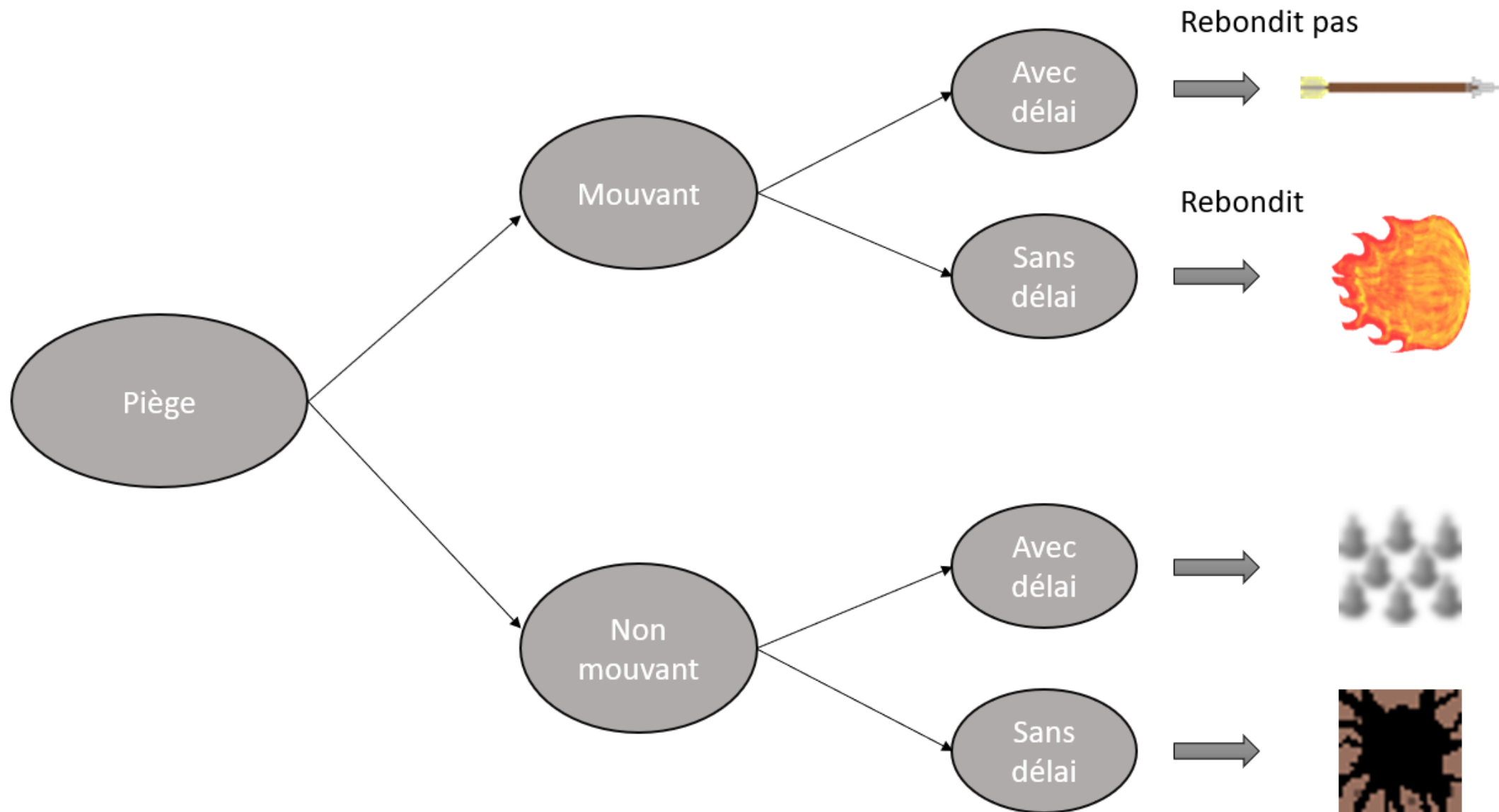
Vue 2.5D : Effet de profondeur

Secondes depuis le dernier tour de boucle

*Vitesse du perso * Taille d'un sprite*



Déplacement du personnage en pixels = (vitesse en pixels / secondes) * Delta temps





Conclusion

Nous aurions aimé ajouter

- Piège roue de feu
- IA dans le labyrinthe
- Niveau énigmes

