Jetpack Escape

Marconnet Hugo - Dumas Yvan

Jetpack Escape

Le jeu:

Type: endless runner

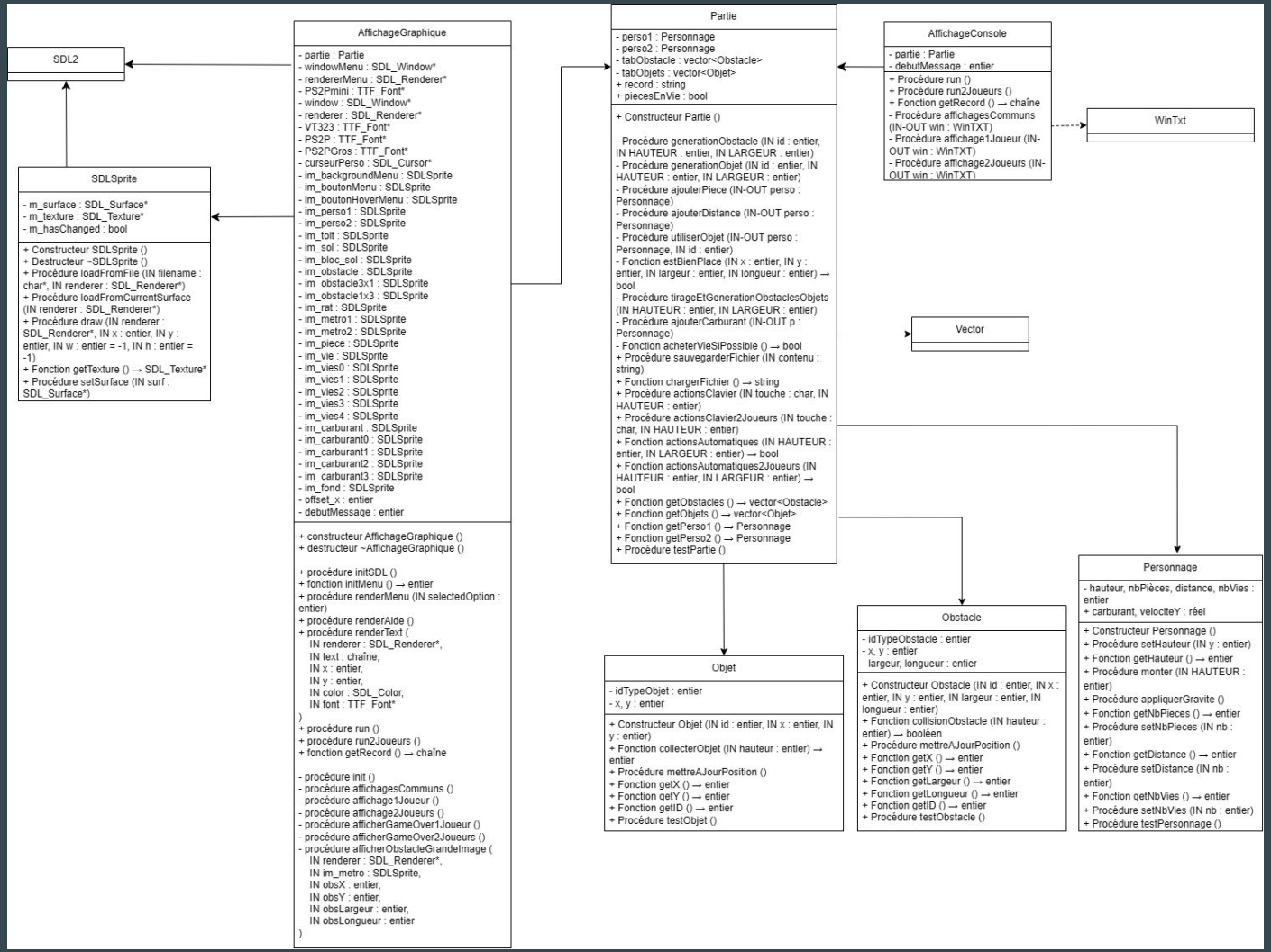
Objectif: parcourir le plus de distance

Graphismes: 2D (vue de côté)

Textures: pixel-art, style 32bits

Le jeu possède un mode un joueur et un mode deux joueurs

Diagramme des classes


SDL2
AffichageGraphique
- partie : Partie
- windowMenu : SDL_Window*
- rendererMenu : SDL_Renderer*
- PS2Pmini : TTF_Font*
- window : SDL_Window*
- renderer : SDL_Renderer*
- VT323 : TTF_Font*
- PS2P : TTF_Font*
- PS2PGros : TTF_Font*
- curseurPerso : SDL_Cursor*
- im_backgroundMenu : SDLSprite
- im_boutonMenu : SDLSprite
- im_boutonHoverMenu : SDLSprite
- im_perso1 : SDLSprite
- im_perso2 : SDLSprite
- im_toit : SDLSprite
- im_sol : SDLSprite
- im_bloc_sol : SDLSprite
- im_obstacle : SDLSprite
- im_obstacle3x1 : SDLSprite
- im_obstacle1x3 : SDLSprite
- im_rat : SDLSprite
- im_metro1 : SDLSprite
- im_metro2 : SDLSprite
- im_piece : SDLSprite
- im_vie : SDLSprite
- im_vies0 : SDLSprite
- im_vies1 : SDLSprite
- im_vies2 : SDLSprite
- im_vies3 : SDLSprite
- im_vies4 : SDLSprite
- im_carburant : SDLSprite
- im_carburant0 : SDLSprite
- im_carburant1 : SDLSprite
- im_carburant2 : SDLSprite
- im_carburant3 : SDLSprite
- im_fond : SDLSprite
- offset_x : entier
- debutMessage : entier
+ constructeur AffichageGraphique ()
+ destructeur ~AffichageGraphique ()
+ procédure initSDL ()
+ fonction initMenu () → entier
+ procédure renderMenu (IN selectedOption : entier)
+ procédure renderAide ()
+ procédure renderText (
IN renderer : SDL_Renderer*,
IN text : chaîne,
IN x : entier,
IN y : entier,
IN color : SDL_Color,
IN font : TTF_Font*
)
+ procédure run ()
+ procédure run2Joueurs ()
+ fonction getRecord () → chaîne
- procédure init ()
- procédure affichagesCommuns ()
- procédure affichage1Joueur ()
- procédure affichage2Joueurs ()
- procédure afficherGameOver1Joueur ()
- procédure afficherGameOver2Joueurs ()
- procédure afficherObstacleGrandeImage (
IN renderer : SDL_Renderer*,
IN im_metro : SDLSprite,
IN obsX : entier,
IN obsY : entier,
IN obsLargeur : entier,
IN obsLongueur : entier
)
SDLSprite
- m_surface : SDL_Surface*
- m_texture : SDL_Texture*
- m_hasChanged : bool
+ Constructeur SDLSprite ()
+ Destructeur ~SDLSprite ()
+ Procédure loadFromFile (IN filename : char*, IN renderer : SDL_Renderer*)
+ Procédure loadFromCurrentSurface (IN renderer : SDL_Renderer*)
+ Procédure draw (IN renderer : SDL_Renderer*, IN x : entier, IN y : entier, IN w : entier = -1, IN h : entier = -1)
+ Fonction getTexture () → SDL_Texture*
+ Procédure setSurface (IN surf : SDL_Surface*)
Partie
- perso1 : Personnage
- perso2 : Personnage
- tabObstacle : vector<Obstacle>
- tabObjets : vector<Objet>
+ record : string
+ piecesEnVie : bool
+ Constructeur Partie ()
- Procédure generationObstacle (IN id : entier, IN HAUTEUR : entier, IN LARGEUR : entier)
- Procédure generationObjet (IN id : entier, IN HAUTEUR : entier, IN LARGEUR : entier)
- Procédure ajouterPiece (IN-OUT perso : Personnage)
- Procédure ajouterDistance (IN-OUT perso : Personnage)
- Procédure utiliserObjet (IN-OUT perso : Personnage, IN id : entier)
- Fonction estBienPlace (IN x : entier, IN y : entier, IN largeur : entier, IN longueur : entier) → bool
- Procédure tirageEtGenerationObstaclesObjets (IN HAUTEUR : entier, IN LARGEUR : entier)
- Procédure ajouterCarburant (IN-OUT p : Personnage)
- Fonction acheterVieSiPossible () → bool
+ Procédure sauvegarderFichier (IN contenu : string)
+ Fonction chargerFichier () → string
+ Procédure actionsClavier (IN touche : char, IN HAUTEUR : entier)
+ Procédure actionsClavier2Joueurs (IN touche : char, IN HAUTEUR : entier)
+ Fonction actionsAutomatiques (IN HAUTEUR : entier, IN LARGEUR : entier) → bool
+ Fonction actionsAutomatiques2Joueurs (IN HAUTEUR : entier, IN LARGEUR : entier) → bool
+ Fonction getObstacles () → vector<Obstacle>
+ Fonction getObjets () → vector<Objet>
+ Fonction getPerso1 () → Personnage
+ Fonction getPerso2 () → Personnage
+ Procédure testPartie ()
AffichageConsole
- partie : Partie
- debutMessage : entier
+ Procédure run ()
+ Procédure run2Joueurs ()
+ Fonction getRecord () → chaîne
- Procédure affichagesCommuns (IN-OUT win : WinTXT)
- Procédure affichage1Joueur (IN-OUT win : WinTXT)
- Procédure affichage2Joueurs (IN-OUT win : WinTXT)
WinTxt
Vector
Objet
- idTypeObjet : entier
- x, y : entier
+ Constructeur Objet (IN id : entier, IN x : entier, IN y : entier)
+ Fonction collecterObjet (IN hauteur : entier) → entier
+ Procédure mettreAJourPosition ()
+ Fonction getX () → entier
+ Fonction getY () → entier
+ Fonction getID () → entier
+ Procédure testObjet ()
Obstacle
- idTypeObstacle : entier
- x, y : entier
- largeur, longueur : entier
+ Constructeur Obstacle (IN id : entier, IN x : entier, IN y : entier, IN largeur : entier, IN longueur : entier)
+ Fonction collisionObstacle (IN hauteur : entier) → booléen
+ Procédure mettreAJourPosition ()
+ Fonction getX () → entier
+ Fonction getY () → entier
+ Fonction getLargeur () → entier
+ Fonction getLongueur () → entier
+ Fonction getID () → entier
+ Procédure testObstacle ()
Personnage
- hauteur, nbPièces, distance, nbVies : entier
+ carburant, velociteY : réel
+ Constructeur Personnage ()
+ Procédure setHauteur (IN y : entier)
+ Fonction getHauteur () → entier
+ Procédure monter (IN HAUTEUR : entier)
+ Procédure appliquerGravite ()
+ Fonction getNbPieces () → entier
+ Procédure setNbPieces (IN nb : entier)
+ Fonction getDistance () → entier
+ Procédure setDistance (IN nb : entier)
+ Fonction getNbVies () → entier
+ Procédure setNbVies (IN nb : entier)
+ Procédure testPersonnage ()

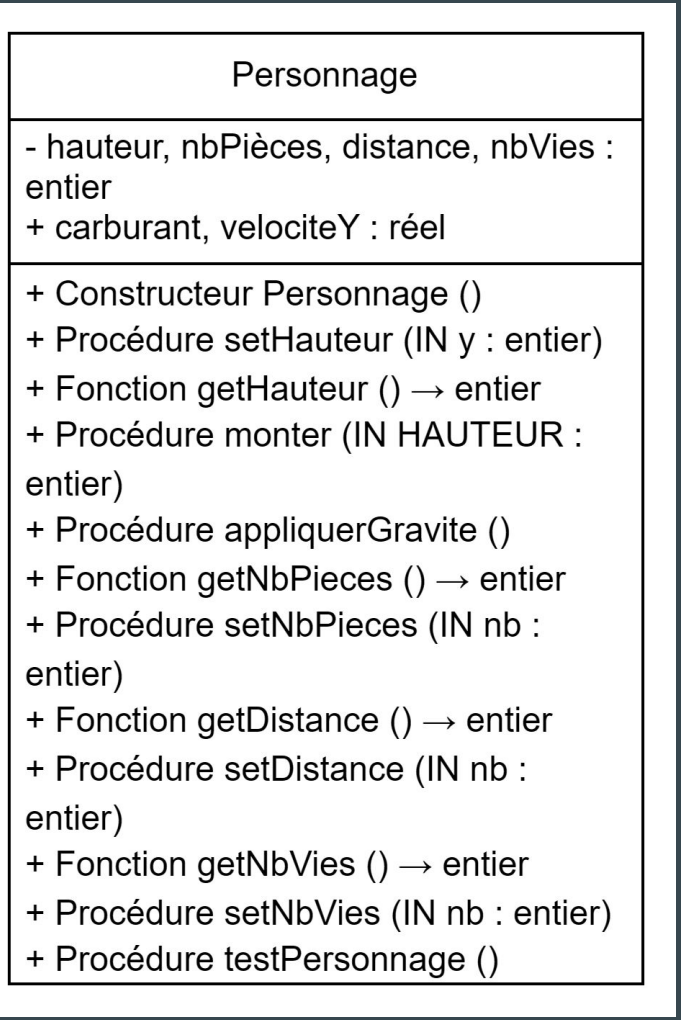
Classe Personnage

hauteur

velociteY

Personnage::monter()

Personnage::appliquerGravite



Classe Obstacles, et similitudes avec la classe Objet

idTypeObstacle

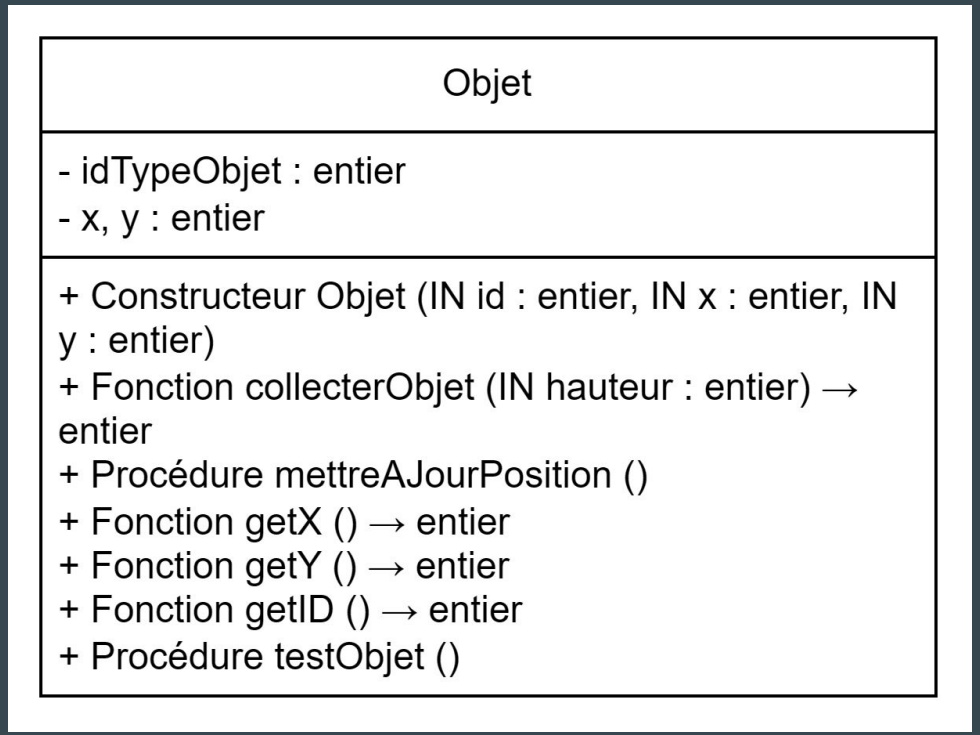
Obstacle::collisionObstacle(int hauteur)

Obstacle::mettreAJourPosition()

Objet::collecterObjet(int hauteur)

Obstacle
- idTypeObstacle : entier - x, y : entier - largeur, longueur : entier
+ Constructeur Obstacle (IN id : entier, IN x : entier, IN y : entier, IN largeur : entier, IN longueur : entier) + Fonction collisionObstacle (IN hauteur : entier) → booléen + Procédure mettreAJourPosition () + Fonction getX () → entier + Fonction getY () → entier + Fonction getLargeur () → entier + Fonction getLongueur () → entier + Fonction getID () → entier + Procédure testObstacle ()

Classe Obstacles, et similitudes avec la classe Objet



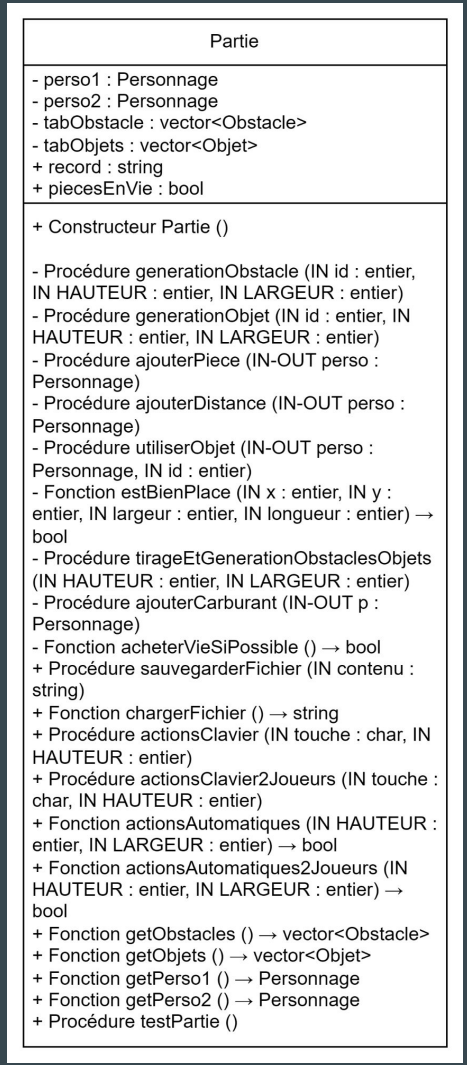
Classe Partie

joueur1

joueur2

Partie::actionsAutomatiques()

Partie::actionsClavier()



Conclusion

- Un projet concret pour mettre en pratique la gestion et le développement d'un jeu

- Exigences du cahier des charges respectées:
  - Déplacement du personnage, inertie
  - Obstacles et objets
  - Carburant limité
  - Mode deux joueurs

- Ce qui peut être amélioré:
  - Sons et effets visuels
  - variété des obstacles/objets