

Space Cow Game

LIFAP4 | Printemps 2018

Université Claude Bernard Lyon 1 Département Informatique

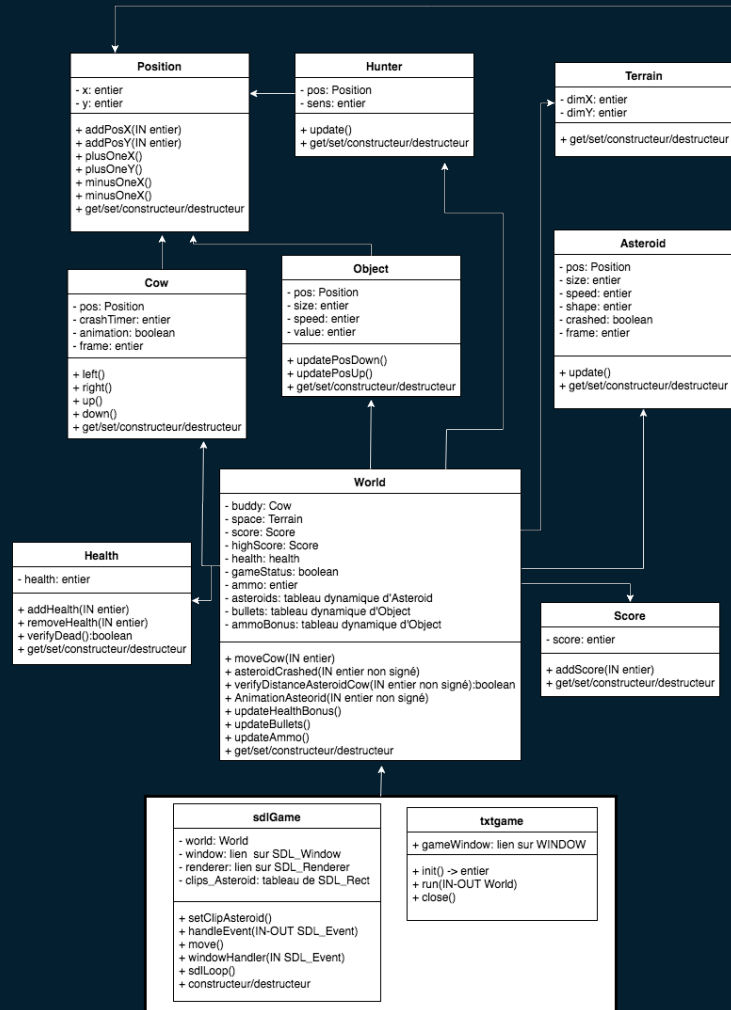


Ali BAKIR 11607221

| Majd ODEH 11608951

| Rodislav IVANOV 11511441

Diagramme des modules





Classe Cow

Elle représente le joueur et gère ses mouvements ainsi

Cow
<ul style="list-style-type: none">- pos: Position- crashTimer: entier- animation: boolean- frame: entier
<ul style="list-style-type: none">+ left()+ right()+ up()+ down()+ get/set/constructeur/destructeur



Asteroid
<ul style="list-style-type: none">- pos: Position- size: entier- speed: entier- shape: entier- crashed: boolean- frame: entier
<ul style="list-style-type: none">+ update()+ get/set/constructeur/destructeur

Classe Asteroid

Elle représente les obstacles qui tombent et que Buddy doit éviter

The background of the slide is a dark blue space scene. In the upper left, there is a small orange planet. Below it is a tiny blue planet. To the right of the blue planet is a large Jupiter-like planet with prominent horizontal bands. Further right is a reddish-orange planet. On the far right, there is a large, colorful geometric shape composed of several overlapping triangles in shades of purple, blue, green, and orange.

Classe World

Elle représente l'état actuel du jeu et regroupe les fonctions les plus importantes

World
<ul style="list-style-type: none">- buddy: Cow- space: Terrain- score: Score- highScore: Score- health: health- gameStatus: boolean- ammo: entier- asteroids: tableau dynamique d'Asteroid- bullets: tableau dynamique d'Object- ammoBonus: tableau dynamique d'Object
<ul style="list-style-type: none">+ moveCow(IN entier)+ asteroidCrashed(IN entier non signé)+ verifyDistanceAsteroidCow(IN entier non signé):boolean+ AnimationAsteorid(IN entier non signé)+ updateHealthBonus()+ updateBullets()+ updateAmmo()+ get/set/constructeur/destructeur



Classe sdlGame

Elle récupère les interactions du joueur, gère les différentes fenêtres, tourne la boucle d'événements

sdlGame

- world: World
- window: lien sur SDL_Window
- renderer: lien sur SDL_Renderer
- clips_Asteroid: tableau de SDL_Rect

- + setClipAsteroid()
- + handleEvent(IN-OUT SDL_Event)
- + move()
- + windowHandler(IN SDL_Event)
- + sdlLoop()
- + constructeur/destructeur

Conclusions!



Le processus de développement a été mené à bien dans les délais



Difficultés rencontrées:

- a. d'implémenter les bibliothèques externes (ex. Ncurses, SDL2...)
- b. Des OS différents



Si l'on disposait de plus de temps...

