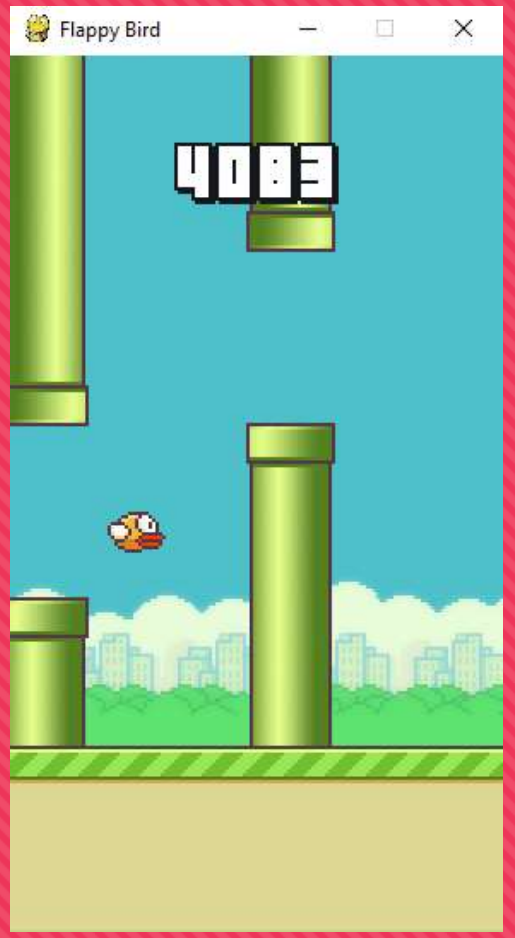
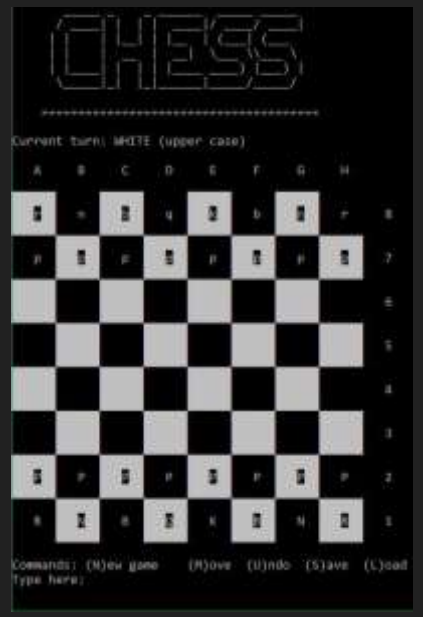
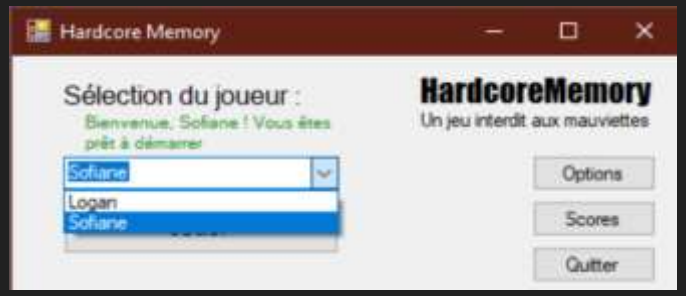
Notions requises :
- Bases de la programmation



Introduction aux librairies graphiques

Ni des applications consoles, ni des winforms : apprenons à réaliser des applications graphiques simplement !





Différentes manières de créer des applications

- Applications consoles
- Formulaires et interfaces natives
- **Applications multimédia**
- Applications web

Différents niveaux d'abstractions

Ecran

- Manipulation directe des pixels de la fenêtre

API Multimédias

- OpenGL
- SDL

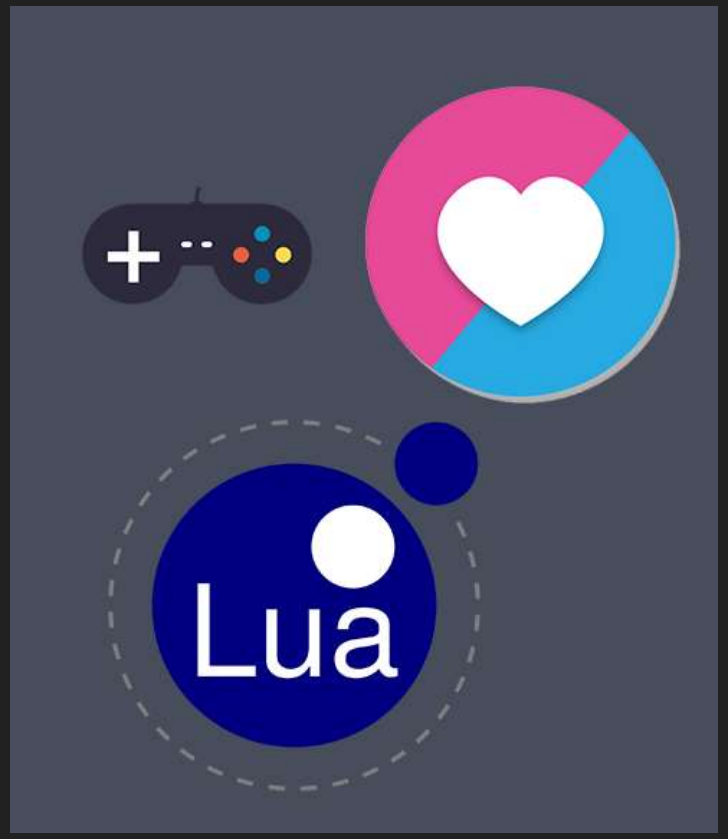
Librairies et Frameworks

- [Py] PyGame
- [JS] **<canvas>** html5
- [C++] SFML
- (framework) [Lua] Love2D
- (framework) [JS] ImpactJS, PixiJS …

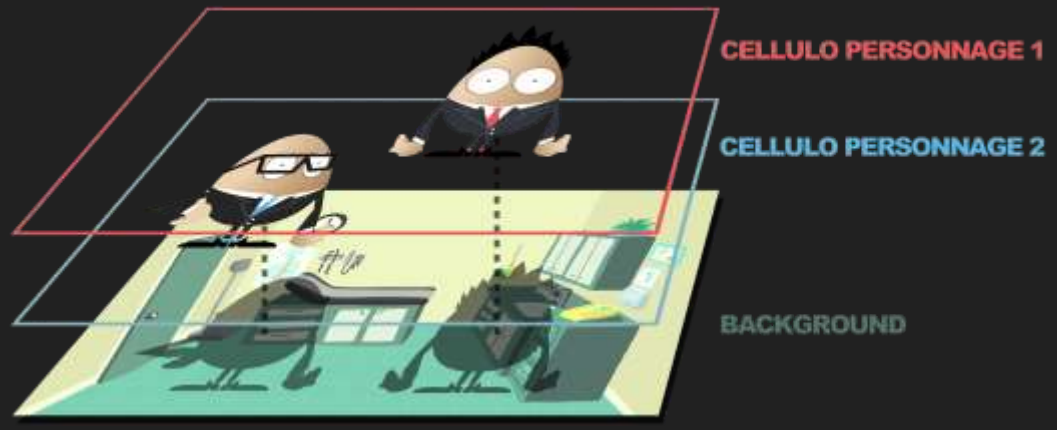
+ moteurs de jeu (Unity, Unreal, Godot…)

- À chaque langage sa librairie !
- Framework vs Librairie
- Les notions théoriques qu'on va voir dans les ateliers devrait fonctionner sur n'importe quelle librairie graphique présentée ici

Love2D



- C'est un Framework écrit en C++, où vous pouvez créer des jeux ou interfaces 2D avec le langage LUA
- Populaire car très simple d'utilisation (utilisé gamejams); permet de comprendre comment une application graphique fonctionne
- Portable : 1 code = Windows, GNU/Linux, Mac, Android…
- Un exemple de jeu créé en moins d'un mois : https://github.com/LoganTann/vocaloidWinterGame



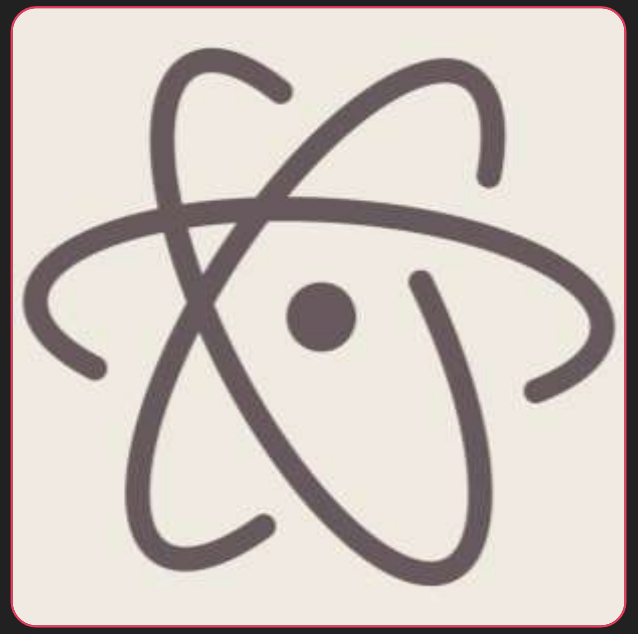
Un concept qui s'inspire du dessin animé

- Pour animer des images, il suffit de les faire défiler rapidement
- Pour créer les frames des dessins animés, on utilisais du celluloid (calques) empilées
- Les librairies graphiques reprennent ce concept, à la différence que l'on pourra programmer l'affichage et prendre en compte des évènements (clic ou mouvement de souris par exemple)

Un concept qui s'inspire du dessin animé

- Plusieurs « objets », qui ont tous une position sur l'écran
- Empilés sur un ordre précis (ce qui est en avant est empilé au dessus de ce qu'il y a en arrière)
- Il se passe un certain temps entre l'affichage de deux frames (24fps = 1/24s → Δt)
- Ce processus d'affichage est répété à l'infini, jusqu'à un évènement de fin (ex : bobine épuisée)

Outils et ressources pour coder



- Installer Love2D https://love2d.org/
- Installer un éditeur de code
  - Préféré : Atom + extension Love2D-IDE
- Un guide rapide sur le langage Lua : https://devhints.io/lua
- Le wiki Love2D pour avoir la documentation de la librairie et des tutoriels : https://love2d.org/wiki/Main_Page

Code de l'atelier

Objectif : Comprendre la structure générale d'un framework 2D, les bases du LUA et l'API Love2D

- Afficher du texte et gérer la couleur de fond
- Afficher le temps écoulé -> delta time
- Gestion des évènements uniques -> cookie clicker
- Afficher une image
- Gestion des évènements continus -> faire avancer le camion

Aller plus loin

- Jeter un œil aux librairies PyGame, SFML ou les canvas html5
- Plus de puissance et de productivité : les moteurs de jeu (type Unity)