Endless Arena

Un jeu de plateforme développé avec python

Henri de Pesquidoux – Olivier Moreau

Table des matières

Principe

Le jeu à se joue à deux, sur un seul clavier. Les joueurs sont l'un contre l'autre, et le premier arrivé à trois points gagne.

Les joueurs contrôlent chacun un personnage pouvant avancer vers la droite ou la gauche, faire un saut simple et double, et frapper l'autre personnage. Ils sont dans un écran comprenant des plateformes mouvantes comme seul support.

Un joueur peut gagner un point en frappant l'autre personnage, et en perdre un en sortant de l'écran par les côtés.

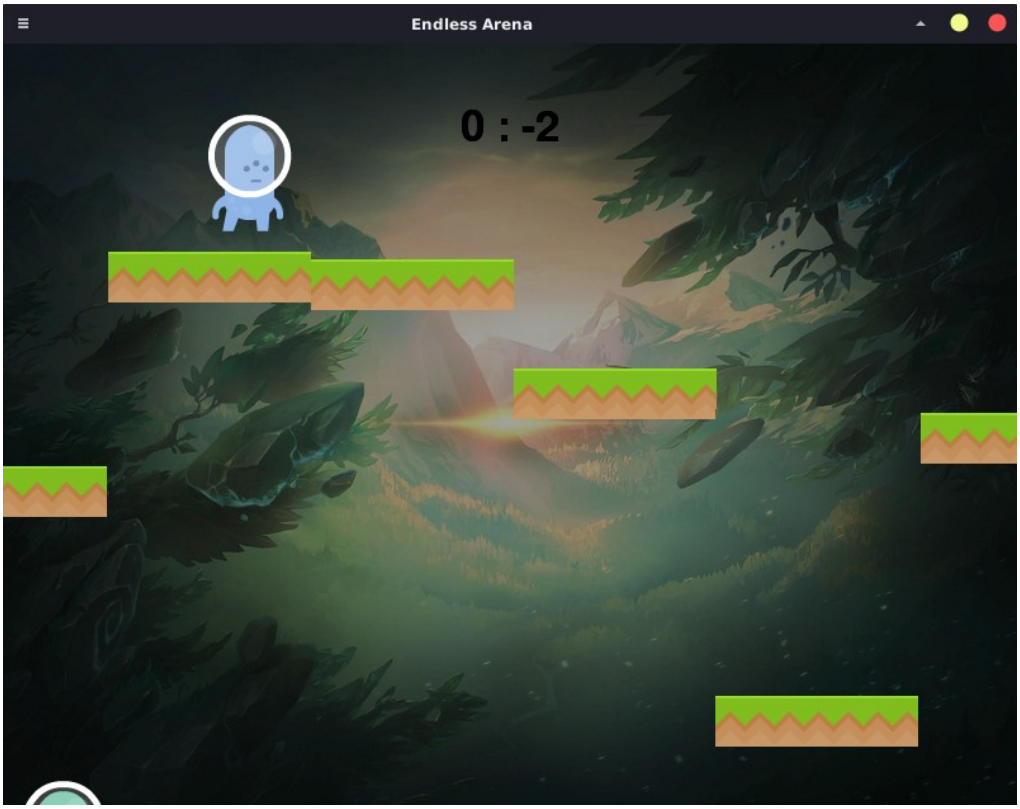

Illustration 1: Illustration du principe avec une capture d'écran

Solutions technologiques

Langage de programmation et moteur de jeu

Nous avons utilisé le langage python, en version 3.6, ainsi que le moteur de jeu pygame, un wrapper sdl python, spécialisé dans la création de jeux.

Ces choix proviennent de plusieurs critères :

- Leur large utilisation nous assurait une documentation consistante ;
- Leur stabilité nous permettait d'éviter au maximum l'apparition de bugs ne provenant pas de notre code ;
- Python est bien connu pour sa facilité d'utilisation et sa puissance, nous permettant de ne pas perdre de temps avec une gestion poussée de la mémoire, comme un autre choix potentiel, Allegro et C++, nous y aurait forcé ;
- Nous avions déjà étudié python en cours d'ISN, nous permettant d'éviter à avoir à apprendre l'utilisation d'un langage nouveau, et de nous focaliser sur l'apprentissage de pygame.

Système de gestion des versions

Nous avons utilisé le système de gestion de version Git, et le service de forge (serveur git) BitBucket.

Cela nous a permis une gestion relativement facile des versions, ainsi qu'un historique clair des progrès, et une mise à jour facile sur chaque poste de travail utilisé.

Nous avons choisi ce système de gestion car il est extrêmement utilisé, d'où l'existence de services d'hébergement de code se basant sur ce système, nous assurant encore une fois une large documentation, et une grande stabilité.

Programme

Programme général

Le diagramme suivant explique le fonctionnement du programme.

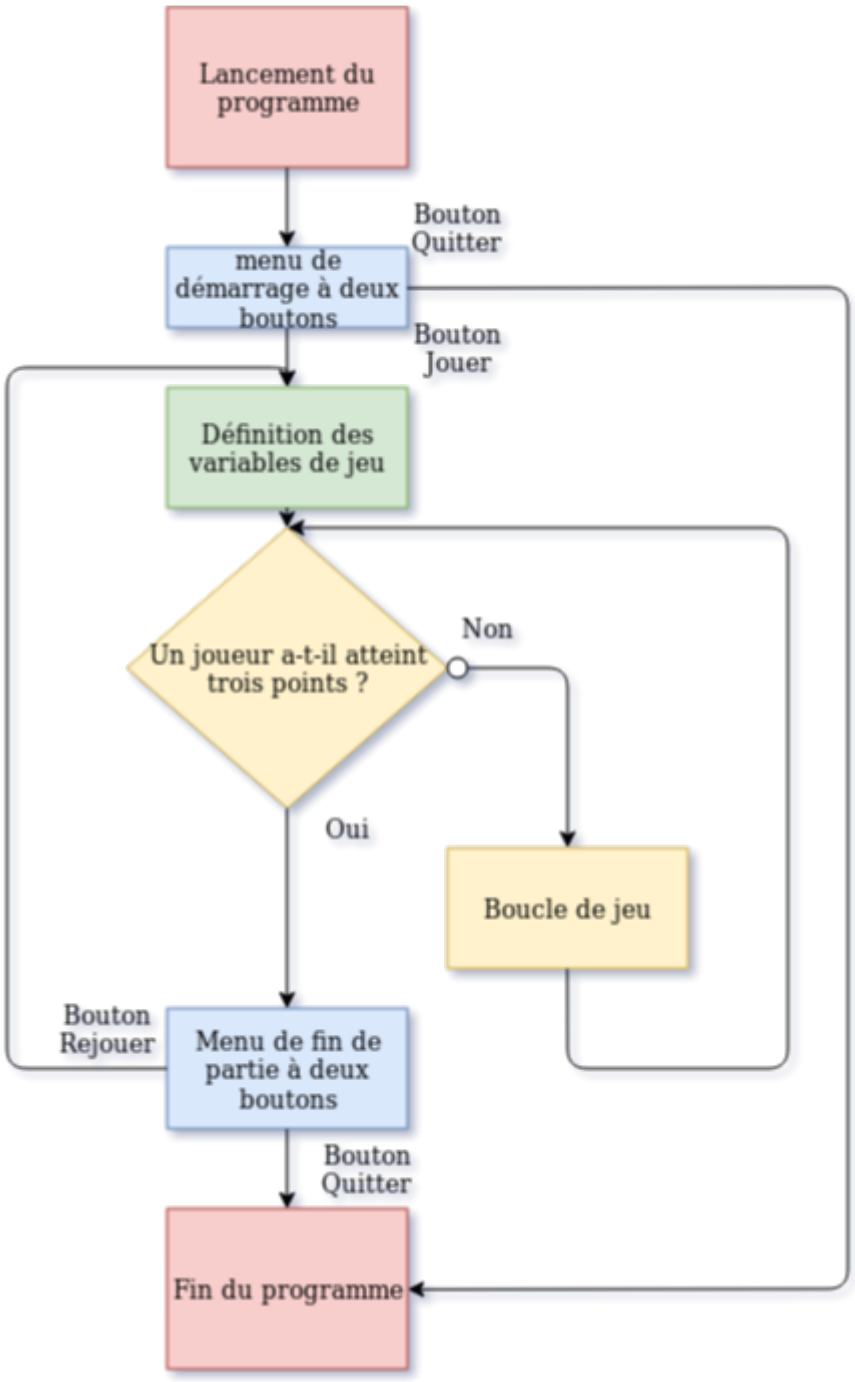

Illustration 2: Algorigramme du programme

Boucle de jeu

Voici une sortie standard du programme, limitée à une itération de la boucle de jeu.

```
[0]: Itération de la boucle
    [1]: Gestion du déplacement des plateformes
    [1]: Gestion des chutes
    [1]: Mise à jour du score (0:2)
    [1]: Mise à jour de l'affichage
```

Illustration 3: Sortie standard boucle de jeu

Sa compréhension à été simplifiée au maximum, et il n'y a que peu de choses à rajouter à la sortie, permettant de s'assurer du fonctionnement des différentes parties du programme.

Étudions chaque partie plus en particulier:

Gestion des entrées

On utilise la fonction de pygame.event.get(), qui renvoie une liste contenant les divers événements apparus durant la boucle, c'est-à-dire ici les touches enfoncées, avec le type d'événement KEYDOWN.

Ensuite :

- si la touche enfoncée est a ou d, on change la direction du personnage 1 ;
- si c'est j ou l, on change la direction du personnage 2 ;
- si c'est une touche de saut (z et i), on vérifie que le personnage n'a pas déjà fait un double saut, puis on le fait sauter, c'est à dire que l'on lui donne une vitesse verticale vers le haut de valeur 5 pixels par frame (il y a 60 frames par seconde, c'est donc une vitesse de 300 pixels par frame) ;
- si c'est une touche d'attaque qui est enfoncée, on vérifie que le personnage attaquant est tourné vers l'autre, se trouve à la même hauteur, et suffisament proche horizontalement.

  Si c'est le cas, on augmente de 1 le score du joueur attaquant, et on renvoie le personnage attaqué (et touché) aux coordonnées de départ.

Gestion du déplacement des plateformes

La gestion des plateformes est assez simple :

Pour chaque plateforme, on vérifie si elle est sortie de l'écran.

Si c'est le cas, on la renvoie à droite avec une hauteur aléatoire.

Sinon, on la fait avancer vers la gauche d'un nombre prédéfini de pixels (variable de jeu vitessePlateforme).

Gestion des chutes

Ensuite, on regarde si les joueurs sont sur une plateforme, en comparant les coordonnées de chaque joueur avec celles de chaque plateforme.

Si le joueur est soutenu, on met à zéro son compteur de saut pour permettre de nouveau un double saut, et on met sa vitesse verticale à 0. De plus, on le déplace avec les plateformes.

Si il n'est pas soutenu, on le fait tomber de sa vitesse verticale, et on diminue celle-ci de 0,1 pixel par frame.

Mise à jour du score

Le score est affiché par un panneau. On utilise le module « font » pour écrire ce score.

Mise à jour de l'affichage

Enfin, on met à jour l'affichage avec les fonctions display.blit() et display.update.

Annexes

Code

Le programme est trop long pour être mis ici en entier, mais vous pouvez le voir ici : https://tinyurl.com/ISNJanson

https://github.com/Makoto242/EndlessArenaMirror

Bibliographie

Vous trouverez ci-dessous la liste des ressources utilisées pour le projet :

- https://www.pygame.org/docs/ : la documentation officielle de pygame.
- https://pythonprogramming.net/pygame-python-3-part-1-intro/ : un tutoriel très bien fait sur l'utilisation de pygame.