



Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page [Préparer votre participation](#).

Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à info@trophees-nsi.fr.

NOM DU PROJET : Infinity jump

> PRÉSENTATION GÉNÉRALE :

Nous souhaitons créer un jeu vidéo dans lequel un personnage situé dans un gouffre devrait sauter de plateformes en plateformes qui tomberaient du ciel afin de ne pas chuter dans le vide. Le but du jeu étant de monter le plus haut possible et donc de tenir longtemps, en accumulant des points, jusqu'à atteindre le sommet. Nous avons pensé à un tel jeu vidéo en discutant de nos expériences personnelles au sein de notre groupe. L'intérêt est de faire un jeu agréable, beau, fluide, drôle et donc addictif !

> ORGANISATION DU TRAVAIL :

Notre groupe est composé de 2 personnes : Faustine et Maëlys. Faustine s'est occupée des décors et des déplacements du personnage tandis que Maëlys a géré les plateformes et leur déplacement. Nous avons ensuite travaillé sur les collisions entre le personnage, les plateformes et les murs.

Nous avons mis en commun toutes nos avancées le mardi pendant nos heures de permanence et nous en avons profité pour faire progresser le projet. Notre professeur a également privilégié des heures de cours pour l'élaboration de nos projets. Nous avons discuté du projet via la plateforme Instagram et nous nous sommes partagé le code par mail. Nous avons aussi utilisé le site kanboard pour pouvoir facilement observer l'avancée du jeu et ce qu'il nous restait à faire.

LES ÉTAPES DU PROJET :

Étape 1 : La réflexion. Que pourrait-on créer ? De quoi s'inspirer ? C'est ainsi qu'en discutant, nous pensons à ce jeu vidéo.

Étape 2 : La répartition des tâches. Découper la programmation en différentes parties afin de se les répartir et de mesurer le travail à accomplir.

Étape 3 : Programmer les plateformes et leur déplacement. Elles tomberont du ciel avec un écart restant toujours atteignable par le personnage.

Étape 4 : Programmer le personnage. Ce sera un oiseau, il pourra se déplacer de gauche à droite ainsi que sauter pour atteindre les plateformes. S'il n'atteint pas une plateforme, il tombe dans le vide.

Étape 5 : Gérer les collisions avec les parois.

Étape 6 : Dessiner les décors.

Étape 7 : Afficher un score.

Étape 8 : Créer une image de victoire et de défaite.

Étape 9 : Créer un « code triche ».

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

Nous avons d'abord terminé la version la plus minimaliste de notre projet, c'est à dire avec le strict minimum pour que ce soit jouable. Puis nous avons apporté des modifications comme le score et les décors. Nous avons pensé à de nouvelles fonctionnalités que nous pourrions programmer comme des pierres qui tombent aléatoirement du ciel, afin d'améliorer le jeu.

Afin de vérifier l'absence de bugs nous avons testé notre jeu tout au long de l'élaboration en essayant de chercher les bugs mais aussi en jouant de manière « normale ». L'interface Python nous a également aidé puisqu'elle nous prévient de certaines erreurs comme les erreurs de syntaxe, les fautes frappes...

Nous avons rencontré de nombreuses difficultés, notamment concernant le saut du personnage mais aussi la collision avec les plateformes en déplacement. Grâce à des recherches, l'aide de notre professeur de NSI ainsi que de nombreux tests nous avons réussi à modifier notre programme de manière à éliminer ces problèmes.

> OUVERTURE :

Pour améliorer notre jeu, il est possible d'ajouter des fonctionnalités. Certaines sont déjà prévues, comme la mise en place de roches qui tombent du ciel ou encore faire tomber les plateformes à l'infini et aléatoirement tout en permettant toujours de pouvoir sauter d'une plateforme à celle d'au-dessus. Nous pourrions également créer un système de vies, de manière à perdre une vie pour chaque roche touchée. La partie serait alors perdue lorsque l'oiseau touche 3 roches ou qu'il tombe dans le vide. En outre, nous pourrions faire baisser les plateformes de plus en plus vite, afin de complexifier le jeu au fur et à mesure de la partie.

Pour promouvoir notre jeu, nous pourrions inventer une histoire pour notre oiseau qui expliquerait comment il s'est retrouvé au fond d'un gouffre. Nous pourrions créer un compte Instagram dédié au jeu, le faire connaître (par exemple en faisant de la pub à partir de nos comptes personnels) et organiser un concours en offrant des récompenses au trois premiers qui arriveraient à terminer le jeu. Nous pourrions également participer à des événements de gaming et des salons pour présenter notre jeu ou encore proposer des versions exclusives de notre jeu à des dates spéciales comme l'anniversaire de création du jeu.

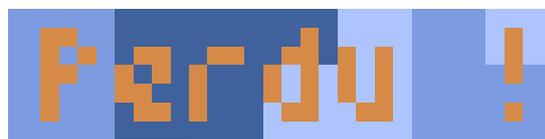
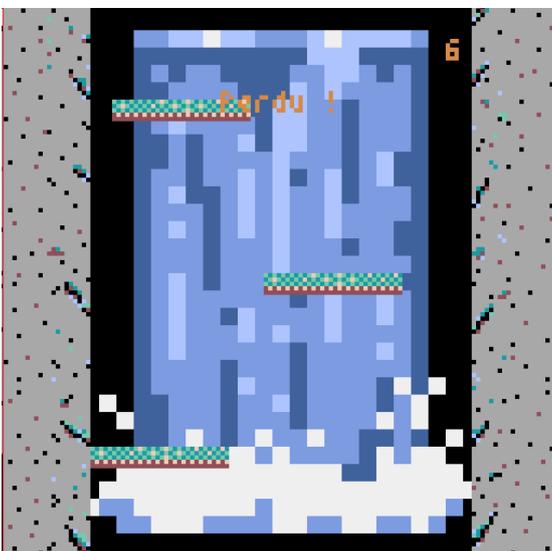
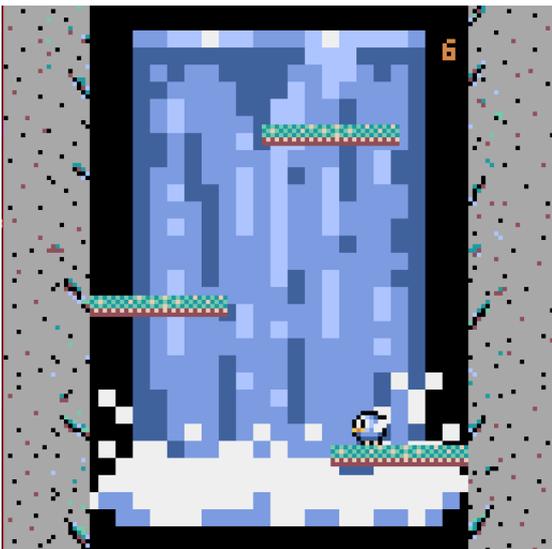
Si c'était à refaire, nous pensons que nous reprendrions la même idée de projet car il permet différentes approches pour de futures améliorations. En outre, il nous a permis d'apprendre de nombreux aspects et fonctionnalités du langage Python et de comment résoudre les problèmes rencontrés.

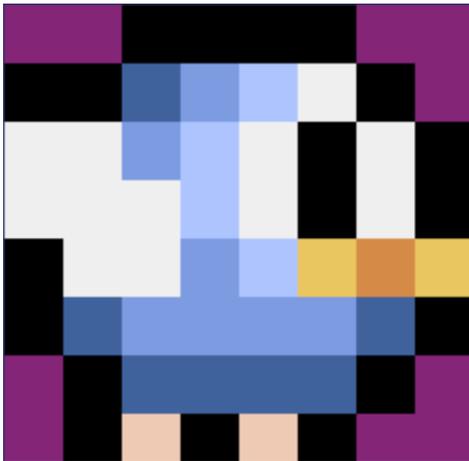
> DOCUMENTATION

Pour l'utilisation de notre jeu nous utilisons le logiciel « edupyster » ainsi que le module « Pyxel » de Python. Notre jeu est donc entièrement programmé en Python. Pour les décors, ils sont stockés dans un fichier .pyxres situé dans le même dossier que notre programme.

Peu de touches sont nécessaires pour l'utilisation de notre jeu. Pour déplacer le personnage il suffit d'appuyer sur la flèche de gauche ou de droite suivant la direction voulue. La touche espace permet de sauter, mais attention, la direction du saut sera choisie en fonction du dernier déplacement du personnage. Si le personnage se dirige vers la gauche il sautera à gauche, et inversement pour la droite. La direction est facilement identifiable grâce au visuel du personnage (voir image à la fin). La touche « a » permet de tricher en relançant le jeu en cas de défaite.

Images du jeu et d'une partie du code :





```

1 import pyxel
2
3 # taille de la fenetre 128x128 pixels
4 taille=128
5 pyxel.init(taille, taille, title="Infinity jump")
6 liste_PF=[[20,120],[78,90],[40,60],[20,30],[50,0],[78,-30],[20,-60],[60,-90],
7 perso_x=liste_PF[0][0]+16
8 perso_y=liste_PF[0][1]-8
9 perso_x0 = 0
10 perso_y0 = 0
11 x0=0
12 y0=0
13 repos = True
14 i=1
15 gagné=True
16 score = 0
17 longueur=32
18 #chargement du gamePack
19 pyxel.load("14.pyxres")
20
21 def draw():
22     global perso_x, perso_y, gagné, repos
23     pyxel.cls(0)
24     #Blitter les murs
25     pyxel.blit(0,0,1,6,0,20,taille,2)
26     pyxel.blit(taille-20,0,1,48,0,20,taille,2)
27     #Blitter la cascade
28     pyxel.blit(20,6,2,14,0,88,116,2)
29     #Blitter le personnage en fonction de son orientation
30     if i == 1:
31         pyxel.blit(perso_x, perso_y,0,0,16,8,8,2)
32     if i == 2:
33         pyxel.blit(perso_x, perso_y,0,0,64,8,8,2)
34     #Plateformes
35     for plateforme in liste_PF:
36         pyxel.blit(plateforme[0], plateforme[1],0,0,72,longueur,5,2)
37     #Score
38     pyxel.text(102,8,str(score),9)

```