

Dossier projet : ISN

Les Fabuleuses aventures du Pizzaiolo



SOMMAIRE

I) présentation du projet

1. Cahier des charges
2. Moyens mis en œuvre
3. Structure du projet
4. Répartition du travail

II) Réalisation personnelle

1. La programmation orientée objet
2. Analyse du code

III) Résultat final

1. Interface de base
2. Différentes interactions avec de décor

IV) Bilan personnel et perspectives

1. Perspectives
2. Bilan personnel

I) présentation du projet

Introduction :

Notre équipe de Pizzaïolo s'est lancée dans un projet en apparence basique (créer un jeu de plateformes) mais qui nécessite de nombreuses connaissances en programmation orientée objet. Nous sommes fiers de vous présenter notre jeu intitulé "Les Fabuleuses aventures du Pizzaïolo" dans lequel notre vedette incontournable du monde de la gastronomie va se lancer à la recherche de la pizza parfaite.

Pourquoi un tel projet ?

Depuis la création de notre groupe, l'idée de créer un jeu de plateformes était presque évidente et après plusieurs jours de réflexion, nous avons arrêté notre réflexion sur le choix d'un jeu en 2D, inspiré de la célèbre saga "Mario".

Nous avons choisi ce projet pour différentes raisons. Premièrement, les jeux Mario ayant bercé nos enfances, c'est l'esprit plein de nostalgie que nous avons décidé de nous replonger dans cet univers de fantaisie. De plus, c'était une occasion pour nous d'étendre nos connaissances en programmation d'une manière ludique.

La charge de travail nécessaire à la création de ce jeu ne nous a pas découragés. D'ailleurs, nous sommes fiers du résultat final. En effet, 5 mois auparavant, nous ne pensions pas être capables de réaliser un tel projet. Ce jeu représente pour nous un apport à la fois personnel et professionnel car sa réalisation n'a pas été aisée, ce qui a permis de renforcer la cohésion au sein du groupe.

La problématique était donc pour nous de créer une interface graphique pouvant accueillir un personnage, se déplaçant de plateformes en plateformes tout en évitant les obstacles

1. Cahier des charges

- Le personnage est un pizzaïolo de couleur rouge et grise.
- Les plateformes sont dans les airs.
- La victoire : récupérer une pizza sans tomber ni toucher l'ennemi

- La défaite :
 - à chaque fois que le pizaiolo tombe dans le vide (entre deux plateformes)
 - à chaque fois que le pizaiolo touche un obstacle (un ennemi)
- Utiliser la Programmation orientée Objet
- Créer différents types de plateformes :
 - des plateformes piège qui tombent lorsque le personnage marche dessus
 - des plateformes mobiles qui se déplacent verticalement
- Possibilité de recommencer le niveau lors d'une chute

2. Moyens mis en œuvre

Notre programme a été entièrement codé en Python. C'est effectivement le seul langage de programmation que nous avons appris cette année, et il s'avère qu'il était tout à fait adéquate pour réaliser notre projet de jeu de plateforme.

Néanmoins, subsistait un problème, car il n'était pas possible de créer une interface graphique en utilisant uniquement Python. Il nous fallait alors trouver un plugin ou un logiciel nous permettant de créer une interface graphique à partir de Python.

C'est pour cela, que nous nous sommes directement tournés vers Pygame qui, comme son nom l'indique, nous permet de créer des jeux en Python en permettant la programmation d'une interface graphique.

Pour en faire une présentation rapide : Pygame est une librairie de commandes pour python spécialisée dans la création d'interface graphique et de jeu



Ensuite, il nous fallait un logiciel de support pour coder et lancer notre programme. Nous avons commencé avec Thonny, mais nous nous sommes rendu compte qu'il était vraiment trop basique. Nous avons besoin d'un logiciel plus complet et c'est pour cela que nous nous sommes rapidement tournés vers Pycharm.

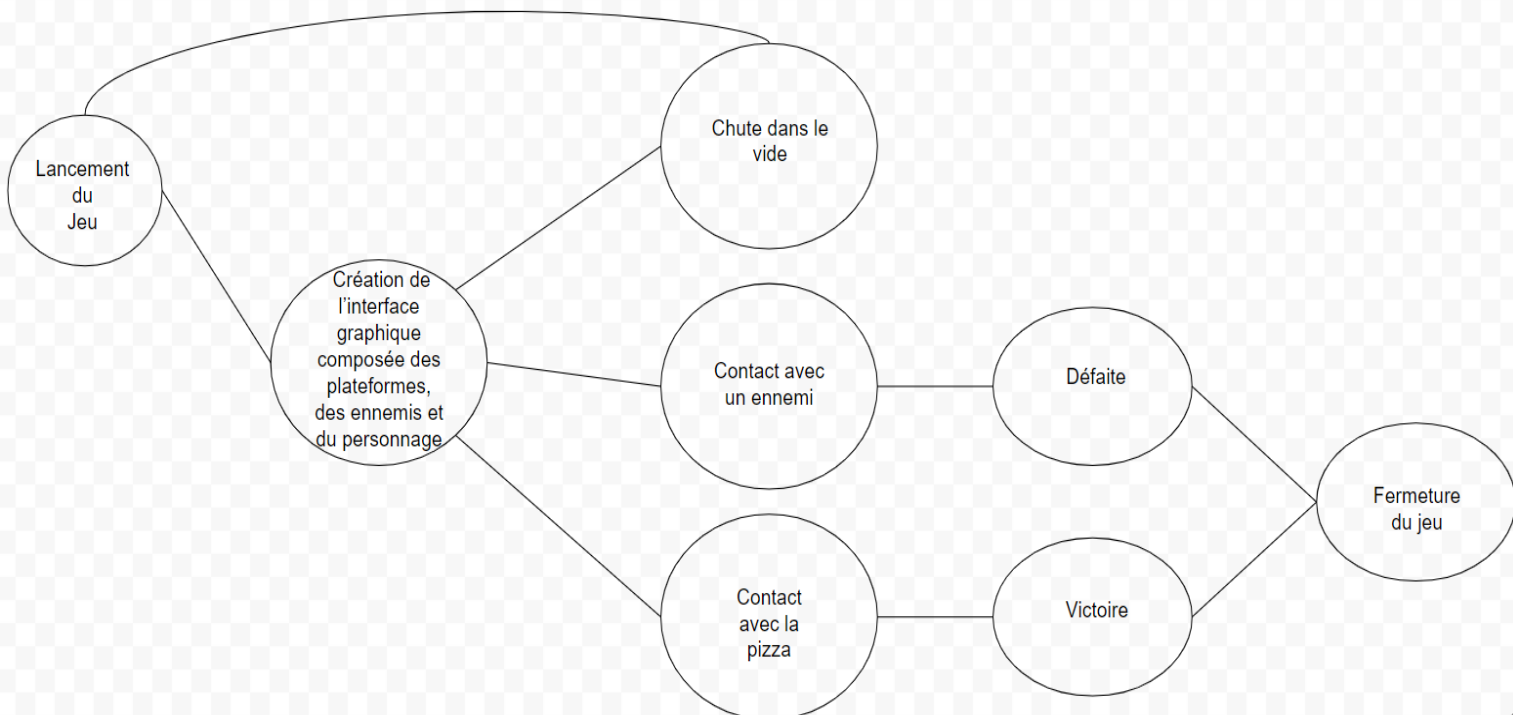


Pycharm est un environnement de développement coloré, permettant un meilleur travail visuel pour différencier les lignes et se retrouver plus facilement dans le code.

Un site a été utilisé pour la réalisation de notre projet, le site Piskel que l'on a utilisé pour le design de nos personnages et de notre plateforme.



3. Structure du projet



4. Répartition du travail

	Avancée des Tâches		
	Antoine	Bastien	Briac
10/01/2019	Choix du sujet : un jeu de plateformes dans lequel un personnage doit franchir différents obstacles afin d'atteindre une pizza		
20/01/2019	Comprendre le fonctionnement d'un jeu sous pygame à l'aide de forums et tutoriels. Néanmoins, cela n'est pas suffisant, ce qui nous pousse à découvrir par nous-mêmes les potentialités de création sous Pygame.		
24/01/2019	Création de l'interface graphique de base dans laquelle le personnage est immobile. Création des personnages et des plateformes à l'aide du site design Piskel.com Répartition du travail: -Création d'une interface dans laquelle on peut faire bouger le personnage -Création des plateformes -création des ennemis et des hitboxes associées		
31/01/2019	Mise en commun des premières lignes de code et découverte des fonctions de base de Pygame		
07/02/2019	recherche d'un objectif de fin de jeu : Design de la pizza		
Vacances de février	Création des différents moyens de mobilité : Implémentation du mouvement et du saut dans le jeu		
14/03/2019	Mise en place de dynamisme additionnel : Implémentation de l'animation de marche.		
03/04/2019	Mise en place de la programmation orientée objet (création des classes) pour ranger le code.		
11/04/2019	Ajout de beaucoup de contenus et de nouvelles classes : Mise en place de la gravité, création des plateformes, des ennemis et des hitboxes. -> le jeu prend forme. Début de la création d'une page d'accueil		

	Recherche et compréhension des différents outils de Tkinter à mes camarades
18/04/2019	Mise en place de l'objectif : Ajout de la princesse (et d'un fond d'écran) Design de l'architecture du début du niveau
09/05/2019	design de l'architecture de la fin du niveau Mise en place de deux fins possibles au jeu : La mort suite à la chute ou en touchant les ennemis, ou alors la victoire en attrapant la pizza. Problème majeur rencontré : Tkinter incompatible avec pygame
12/05	Ajout de commentaires afin d'expliquer le rôle des fonctions et changement du nom de certaines variables

II) Réalisation personnelle

1) La Programmation orientée objet (POO)

Lors de ce projet, j'ai travaillé sur l'affichage du personnage à l'écran. Il a d'abord fallu que je comprenne comment fonctionne l'interface graphique avec Pygame (c'est une bibliothèque de données qui facilite le développement de jeux vidéo temps réel avec le langage de programmation Python).

Cependant, nous nous sommes rapidement rendu compte que la programmation « classique » était limitée car le code comporte trop de redondances. Nous avons donc trouvé un moyen nous permettant de réduire le nombre de lignes du code et donc les erreurs d'inattention.

C'est ainsi que je me suis initié à la Programmation Orientée Objet. La programmation orientée objet (POO) permet de créer des entités (objets) que l'on peut manipuler. Elle impose des structures solides et claires. Les objets peuvent interagir entre eux, cela facilite grandement la compréhension du code.

Au sein de la programmation orientée objet, on peut distinguer les classes. Elles regroupent des fonctions et des attributs qui définissent un objet.

2) Analyse du code

```
class ennemie(object):  
    def __init__(self, x, y, height, width):  
        self.x = x  
        self.y = y  
        self.height = height  
        self.width = width  
        self.vel = 10  
        self.hitbox = (self.x + 20, self.y + 10, 60, 90)  
        pygame.draw.rect(surface, white, self.hitbox, 2)
```

Ici, dans la classe « ennemie », la première fonction contient l'attribut « self » qui permet d'associer des caractéristiques à « x », « y », « height », « width » (ces variables représentent la position de l'objet dans l'espace grâce à « x » et « y » mais aussi les dimensions de ce dernier avec « height » et « width »). Autrement dit, avec la programmation orientée objet, on assigne à un objet virtuel des caractéristiques précises ; tout comme dans la vie de tous les jours où on associe à un objet tel qu'une balle, une certaine taille, un poids donné, etc.

Le but de la POO est donc d'éviter de créer des boucles interminables afin de faire apparaître un objet sur une interface. De plus, elle permet d'éviter l'apparition de bugs à la limite des intervalles.

Ainsi, l'idée initiale était de déplacer le personnage dans la fenêtre afin qu'il puisse se déplacer de plateforme en plateforme. Néanmoins, un problème majeur a fait surface : la taille de notre jeu était limitée par celle de nos écrans respectifs.

C'est là que la programmation orientée objet a été utile. En effet, au lieu de déplacer le personnage, il nous a fallu faire bouger de « décor » afin de ne pas être limité par la taille de l'écran. La POO nous a donc permis de donner l'impression que l'écran suit le personnage au cours de ses déplacements comme une caméra.


```
125 class plateforme(object):
126     def __init__(self, x, y, height, width): #Fonction d'initialisation au sein de la classe permettant de définir les caractéristiques de la plateforme
127         self.x = x
128         self.y = y
129         self.height = height
130         self.width = width
131         self.vel = 10
132         self.hitbox = (self.x, self.y, 300, 10)
133
134     def Mouvementplateforme(self, surface): # Cette fonction permet de faire bouger les plateformes lorsqu'on appuie sur les touches
135         # L'attribut Surface représente les plateformes
136         if keys[pygame.K_RIGHT]:
137             self.x -= self.vel
138         elif keys[pygame.K_LEFT]:
139             self.x += self.vel
140         self.hitbox = (self.x, self.y, 300, 10)
141         surface.blit(platformimg, (self.x, self.y))
```

On peut voir ci-dessus la classe « plateforme » permettant le mouvement des plateformes. La fonction initiale, comme nous l'avons vu précédemment permet de définir les caractéristiques de base de l'objet « platform » à savoir son emplacement, sa taille, sa vitesse et sa hitbox (zone de collision).

La fonction « **Mouvementplateforme** » a comme attribut « surface » ce qui permet d'afficher les plateformes dans la fenêtre.

Si on appuie sur la flèche pour aller vers la droite, les plateformes se déplacent vers la gauche, donnant l'impression que le personnage va vers la droite. De même, si on appuie sur la flèche de gauche, les plateformes se déplacent vers la droite.

La fonction « **self.hitbox = (self.x, self.y, 300, 10)** » permet de définir les coordonnées et la taille de la hitbox des plateformes. Une hitbox est une forme invisible utilisée dans les jeux vidéo pour détecter les collisions entre objets en temps réel. Dans notre cas, elle a une forme rectangulaire.

La fonction « **Surface.blit(platformimg, (self.x, self.y))** » permet d'afficher l'image de la plateforme en fonction des coordonnées en abscisse et en ordonnée.

III) Résultat final

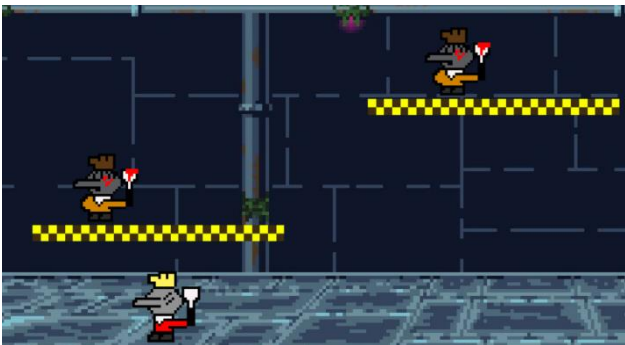
Voici notre réalisation finale :

1) Interface de base



Création de l'interface graphique composée de plateformes, ennemis et du personnage principal.

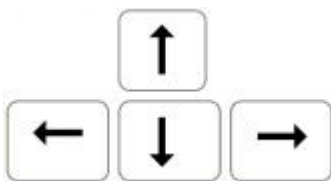
2) Différentes interactions avec le décor



En cas de chute, le personnage revient à sa position initiale.



Lors d'un contact avec un ennemi, c'est la défaite.



Lorsqu'on appuie sur les flèches de direction du clavier, les plateformes et les ennemis bougent, donnant l'impression que le personnage principal se déplace. La barre espace permet de sauter.



Lors d'un contact avec la pizza, c'est la victoire.

Notre projet répond au cahier des charges. En effet, on y trouve un personnage pouvant se déplacer de plateforme en plateforme. Le jeu recommence depuis le début lorsque le personnage tombe d'une plateforme.

On peut ajouter que le fait de faire bouger le décor à la place du personnage nous a permis de créer un grand niveau, composé de 16 plateformes. Une telle réalisation n'aurait pas pu être effectuée sans la programmation orientée objet.

De plus, nous avons ajouté certaines options qui n'étaient pas prévues à l'origine dans le cahier des charges. On a créé des plateformes « spéciales » la première se déplace verticalement dans la fenêtre et l'autre tombe lorsque le personnage marche dessus. Ces deux idées ont été appliquées grâce à un travail collaboratif en groupe.

Pour parvenir à la version finale de notre projet, nous sommes passés par plusieurs étapes dans notre code.

Nous avons d'abord réussi à afficher les plateformes, les ennemis ainsi que leurs hitbox respectives. Puis nous avons effectué plusieurs tests avant de pouvoir faire bouger le décor, donnant l'impression que le personnage est en mouvement.

De plus, grâce à GitHub, nous avons mis en commun nos avancées respectives. Cela a permis de structurer notre travail tout en gardant un œil sur les problèmes des camarades afin de s'entraider.

IV) Bilan personnel et perspectives

1) Perspectives

Au terme de ces six mois passés sur notre projet, nous pouvons être fiers de notre réalisation même si celle-ci n'est pas parfaite, nous avons "appris à apprendre" en découvrant les rudiments de la création d'un jeu.

Pour améliorer ce projet, je pense qu'il faudrait créer une fonction rassemblant les caractéristiques des plateformes et ennemis.

Cela nous aurait permis :

- de réduire la taille du code
- de pouvoir faire réapparaître le personnage à sa position initiale suite au contact avec un ennemi.

On peut ajouter que nous avons eu beaucoup de mal à créer le “cœur” du jeu en intégrant la programmation orientée objet. Néanmoins, cela nous a permis de découvrir une nouvelle manière de coder.

Par ailleurs, nous n’avons pas réussi à intégrer une page d’accueil à notre jeu car nous n’avons pas trouvé le moyen d’utiliser Tkinter et Pygame dans un même programme.

2) Bilan personnel

Ce projet m’a beaucoup apporté, cela m’a permis de découvrir le fonctionnement des interfaces graphiques lors d’applications concrètes.

De plus, étant donné que je me destine à faire des études d’ingénieur, le fait de m’être initié à la programmation orientée objet est donc un avant-goût de mes futurs cours d’informatique.

Par ailleurs, j’ai noté qu’une ambiance studieuse et de cohésion s’est installée immédiatement au sein de notre groupe de travail, bien que nous n’ayons pas l’habitude de travailler ensemble.

Enfin, je pense qu’il m’a été tout à fait profitable d’expérimenter la pédagogie par projet, car elle est représentative du travail mené en entreprise. En effet, dans le monde professionnel, le travail d’équipe est privilégié afin de mettre en commun les connaissances de chacun.