



Licence Informatique

Rapport du Projet

Méthodologie de la programmation L1

Auteur :
Loïc LOKO

Numéro étudiant :
15607684

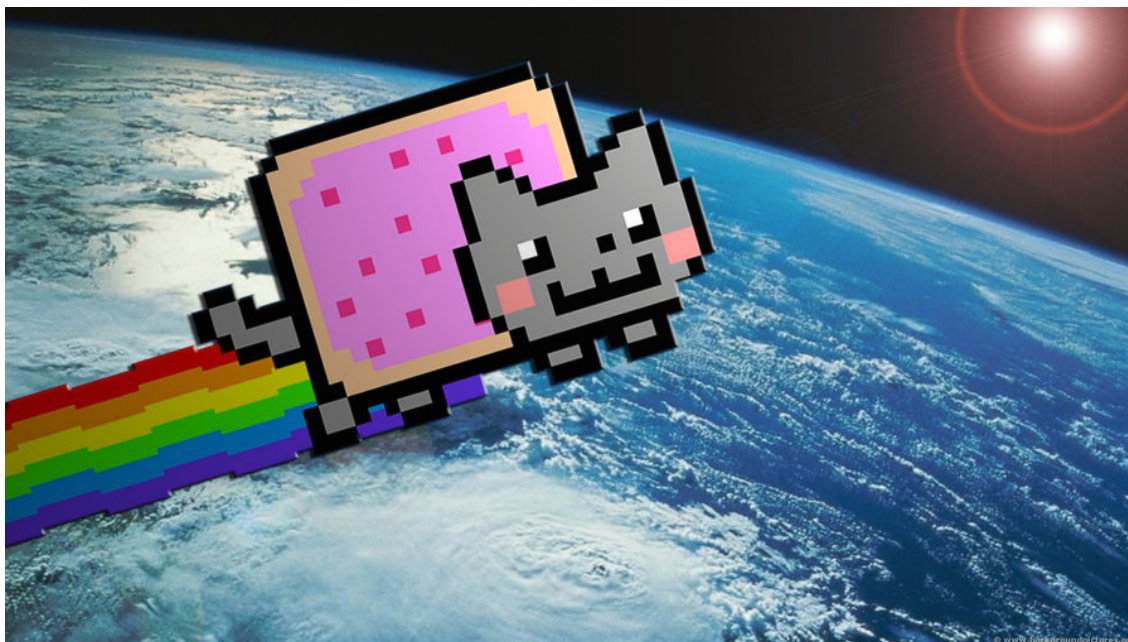


Table des matières

1	Odysée Ludique	3
1.1	Notre jeu Nyancat !	3
1.1.1	Comment un passionné à décider de s'improviser créateur de jeux	3
1.2	Cahier de charges	4
2	Architecture logicielle	8
2.0.1	Cours	10

Chapitre 1

Odysée Ludique

1.1 Notre jeu Nyancat !

1.1.1 Comment un passionné à décider de s'improviser créateur de jeux

à propos de Nyancat

Nyan cat est une animation d'un chat, réalisée par Christopher Torres sous le pseudonyme « PRguitarman », il l'a crée pour l'optique du « mignon ». L'utilisateur de Youtube SraJ00n, a combiné cette animation du chat avec la version du son Nyanyanyanyanyanyanya !.

Le but du jeu

La compréhension du jeu est limpide, il en est de même pour la manière de jouer. Nous dirigeons horizontalement un personnage dans l'espace. Le joueur avance tout seul, nous devons juste le faire monter et descendre. Le côté austère du jeu pour ce qu'il en est de la compréhension de la maniabilité, ne le définit pas totalement. En effet ce jeu n'est pas laxiste, pas exempté d'obstacles, car tout en nous faisant voyager dans un univers spatiale, il nous montre le côté périlleux de cet univers. En général, il s'agira donc le voler dans l'espace en évitant des comètes. Il sera vraiment casse-gueule souvent mais continuons sans plus tardé, Je vous présente une capture d'écran du jeu qu'on se propose de créer : « *Nyanyanyanyanyanyanya !.* »

Core gameplay

Lorsque vous démarrez le jeu Nyancat, vous arrivez sur une page d'accueil, en suivant les instructions vous rentrerez dans le jeu, comment y jouer ? C'est simple, la touche haut vous permet de monter, la touche bas de descendre, vous pouvez maintenir une des touches pour monter plus vite, il y a des limites sur l'écran à ne pas franchir comme celle du score, si c'est le cas vous êtes Game Over Daamn, vous devez éviter les obstacles, mais attention diriger le personnage devient très difficile au bout d'un moment ! wOOt

Gameplays secondaires

Vous voulez vous échapper d'une collision probable ? Utilisez la touche K pour une téléportation rapide vers le bas et seulement vers le bas. L'étoile est votre amie, lors de vos conquêtes spatiales elle vous aidera à augmenter votre score, comme jadis l'amitié qui régnait entre Nyancat et cette étoile. Appuyez la touche espace pour mettre le jeu en pause durant 5s, il vous faudra donc pressez plusieurs fois 6 fois d'affilé par exemple afin d'avoir une pause de 30s.

Jusqu'où irons nous ?

Ne pas s'étendre énormément, juste assurer au moins le minium de ce qu'on veut véhiculer à travers ce jeu .

1.2 Cahier de charges

Ce que le jeu pourra faire

- Jeu 2D
- Son
- Collisions avec les glaçons
- Le background doit tourner en boucle
- Le but du jeu est de faire le meilleur score possible, soit le meilleur temps / ou, la partie sera terminée quand Nyancat aura atteint son ami Nyan (j'hésite entre ces deux choix)
- Rapidité de Nyan :3
- Personnage avançant tout seul

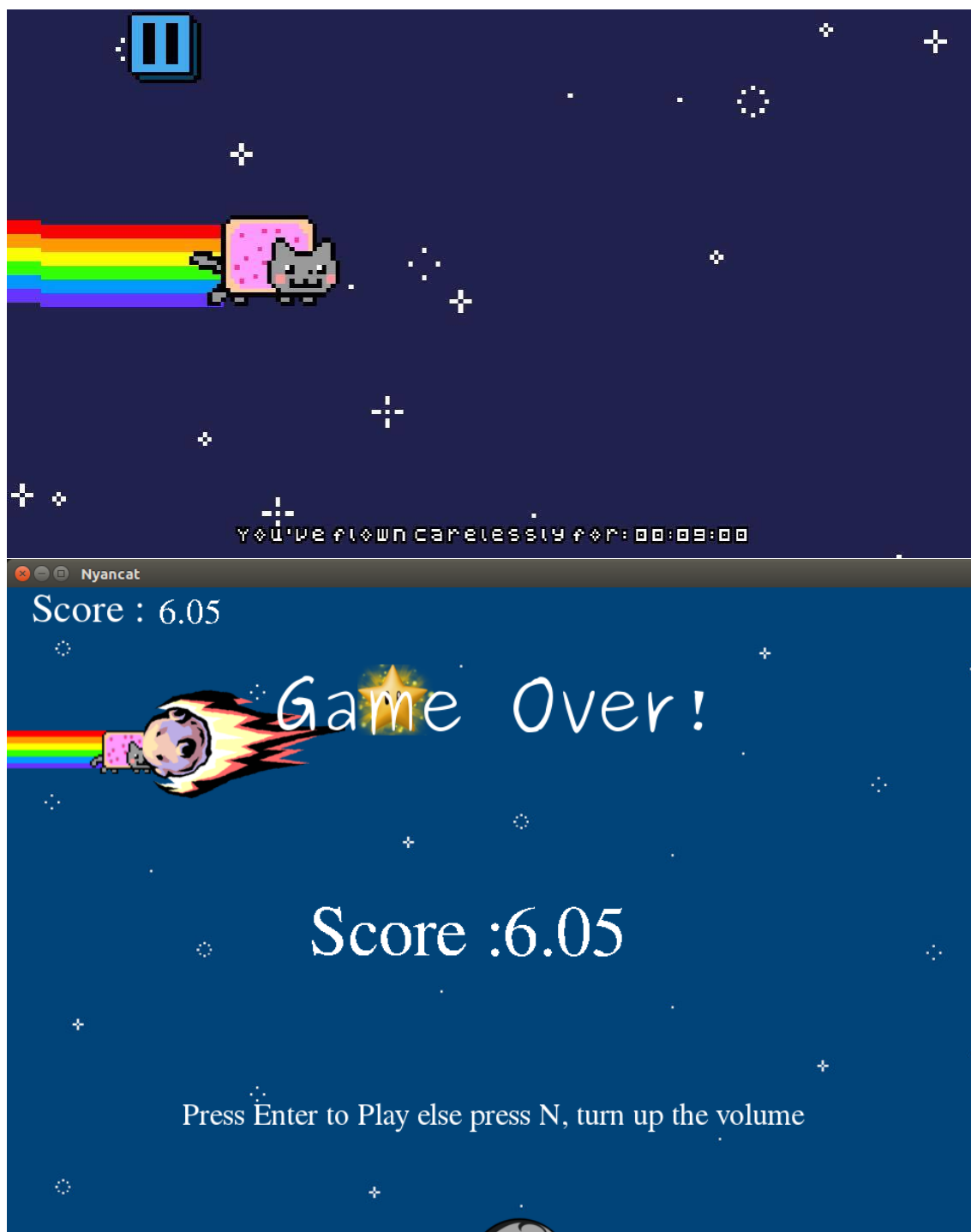


FIGURE 1.1 – Screens du personnage et d'un des obstacles

Ce que le jeu ne pourra pas faire

- Généralement, lorsque le personnage Nyancat se déplace, l'arc-en-ciel qu'il laisse derrière lui devrait systématiquement bouger (faire des mouvements serpent) quand il monte ou descend.
- La fonction pause normale
- Les étoiles sur l'image de fond ne scintilleront pas
- Jeu trop basique

Difficultés rencontrées

- Définir une fonction qui affichera le nouveau score si l'on a dépassé le précédent (Mauvaise organisation de mon temps)
- Utiliser pygame sans les rectangles me mettait dans l'allégresse, heureusement que je me suis très vite rendu compte que j'en aurais besoin, j'ai donc eu du mal pour les collisions.
- UnboundLocalError : local variable 'y' referenced before assignment, souvent même lorsque j'avais défini déjà une fonction au début du code, ma boucle principale m'affichait cette erreur avec le "y", il fallait donc que je l'insère dans ma boucle principale, je ne sais pas à quoi est due cela.
- Faire un menu astuce ou expliquer comment jouer, j'avais un problème avec mon Menu général, normalement ce menu astuce doit se retrouver après le Menu général, mais quand je le mettais après celui-ci il ne fonctionnait pas, tandis que quand je le mettais avant mon menu il fonctionnait. Je me suis rendu compte plus tard pourquoi il y avait ce bug, mais je ne l'ai pas réglé. Faire défiler l'image de fond et le personnage simultanément
- le son principale, crashait, j'étais donc obligé de le mettre dans la fonction quickfail

UnboundLocalError : local variable 'y' referenced before assignment

Grilles des contraintes

Critères	Tests/Réponses
Tout le monde perd ?	Oui
Quel est l'objectif du jeu à atteindre ?	Meilleur score
Les règles sont écrites ?	Non
Tout le monde perd ?	Oui
Durée du jeu	Illimité/Sauf si Game Over
Qualité esthétique du jeu	2D
Aspect pratique	Le code
Le jeu procure du plaisir à jouer	Oui/Non
Le jeu fonctionne ?	Oui

Chapitre 2

Architecture logicielle

Pour le moment je met toutes mes classes et parties le même fichier, les autres fichiers sont juste mes différentes modifications du fichier original, en vue d'éviter tout problème. Je compte créer d'autres fichiers quand je maîtriserai bien pygame.

J'utilise un commentaire de texte, je précise que je suis sous Python2, en représailles, les caractères comme les accents, ne sont pas lisibles. Pour remédier à ce problème j'ai mis au préalable de mon code coding : utf-8 qui me permettra de lire tous les caractères.

- En plus de `import pygame` et `time`, j'utilise le code `from pygame.locals`, qui nous permettra d'importer les constantes.
- `Convert` et `convert_alpha`, permettront de convertir notre image au bon format.
- J'utiliserai aussi une boucle dans l'optique d'ouvrir et fermer la fenêtre, la variable `QUIT` est un événement qui provient des locaux de pygame.
- `Keydown`, soit l'événement quand la touche est pressée, me sera utile dans les déplacements du personnage
- `Blit` sera utilisée, je définirai aussi la fonction/couleur `White` en RGB
- En cherchant dans la bibliothèque de pygame, j'ai pu importer les polices de mon choix grâce à `pygame.font.SysFont`
- Je définis mes différentes images ainsi que leurs rectangles
- J'importe les sons avec `pygame.mixer.sound`
- j'ai défini une image pour mon menu, sur laquelle j'ai écrit avec la fonction `message` que j'ai créé qui utilise évidemment la fonction `Police`

- les fonctions obs et collisions sont propices aux collisions sur le jeu, dans collision, l'idée est la suivante, si le haut de mon personnage est supérieur au bas de mon obstacle etc... dans tous ces cas, il n'y a pas collisions.
- La fonction QuickFail, je l'utilise pour mon GameOver,
- Je définis une fonction pause qui contient juste un compte à rebours de 5s avant que le jeu ne reprenne là où il s'était arrêté.
- Dans la fonction message juste en bas, l'idée est la suivante " Tant que QuickFail est égal à None le temps e fige" le jeu va se figer et va afficher GameOver
- La fonction scored affiche le message "Score :"
- La fonction GameOver utilise message qui je rappel à la fonction clock.tick(), et importe aussi le son du game over et dit que le son principale se stop.
- La fonction Menu est une boucle, si j'appuie sur Entrée l'image importée et le texte écrit partent et le jeu commence.
- Dans ma boucle principale, j'ai plusieurs variables, celles que nous sont méconnaissables sont les variables xboule etc..., utilisées pour que mes obstacles défilent sur l'écran de manière aléatoire.
- if tk >= 0 : offset = (offset+speed)... fenetre.blit(fond,(offset,0)) fenetre.blit(fond,(offset-width,0)) Cette fonction est utilisée pour le déplacement en boucle de mon fond.
- Les variables" variable variable2, score"je les utilise pour mon score, j'affiche une variable juste en haut à coté du message score, et l'autre variable est toujours en cours, sauf qu'elle s'affiche seulement quand je met, elle a donc la fonction d'afficher le score que fais.
- Si il y a collision avec mon étoile, mon score augmente de 1.01, tant que je suis sur l'étoile précisément.
- j'ai applique la fonction collision a mes obstacles
- j'affiche plusieurs fois mon étoile car je veux qu'elle scintille!!!!
- en fin de code est marqué : f nyanrect.bottom > fenetrehauteur +15 or nyanrect.top < -15 : fenetre.blit(score2,(300,300)) gameOver() Si le haut et le bas de mon personnage depasse la limite de l'écran il y a GameOver et le temps s'affiche
- En dessous de la boucle précédente, l'idée véhiculée dans ce code est " Chaque fois que le score atteint le nombre le jeu devient rapide"

Les outils

Pour créer ce jeu, j'ai fait appel au langage Python, ainsi que de la bibliothèque Pygame. ils ont l'avantage d'être gratuit, rapide à l'exécution, ce qui nous facilitera les choses pour structurer ce projet.

2.0.1 Cours

Dans le but de parfaire nos bases : La création de ce jeu a été possible grâce aux cours et aux indices de notre professeur Mr Pablo Rauzy, les différentes bases de pygames et la manière d'utiliser la réflexion de façon informatique et pragmatique découlent de son enseignement. Tous les remerciements au Professeur, surtout pour avoir réussi à nous faire aimer programmer !