



Mini-projet d'Infographie

42run

Résumé: “Dans tous les films d’aventure, il y a une scène dans laquelle le héros met finalement la main sur la moto du grand méchant, mais ensuite il doit déjouer une multitude de pièges afin d’en sortir vivant. 42 Run, c’est cette scène. Et c’est incroyable.”

Table des matières

I	Préambule	2
II	Sujet	4
II.1	Vous codiez ? et bien courez maintenant.	4
II.2	Ce qu'il faut réaliser	4
II.3	Ce que vous pouvez ou pas faire	4
III	Bonus	6
IV	Démo	7

Chapitre I

Préambule



Nom de Zeus ! Marty, tu ne réfléchis pas en 4 dimensions !



Oh là, moi, déjà avec la 3D je suis perdu, alors....

Chapitre II

Sujet

II.1 Vous codiez ? et bien courez maintenant.

Vous avez aimé Temple Run ? Et bien on va se faire la même. Le pitch est un poil différent : vous avez malencontreusement touché à la moto d'un gros baleze, et du coup, il est pas content. Bref vous devez courir ! et vite. Mais attention : de nombreux obstacles parsèment votre chemin, vous devez les éviter. Jusqu'où irez-vous ?

II.2 Ce qu'il faut réaliser

Votre objectif est de réaliser un petit programme qui affiche une course sans fin en 3D, en reprenant les codes du gameplay de Temple Run / Temple Run 2, et qui se déroule dans les locaux de l'école. Le programme doit à minima fournir les éléments suivants :

- Une vision du décor en perspective
- Le décor défile pour nous donner l'impression d'avancer
- Un décor généré aléatoirement en réutilisant un jeu fini d'objets 3D mis bout à bout
- Un décor qui reprend les éléments architecturaux et visuels de l'école
- Un personnage immobile en profondeur et que l'on peut déplacer latéralement et faire sauter
- Des obstacles à éviter et à sauter, sinon la partie s'arrête.
- Un compteur de distance

II.3 Ce que vous pouvez ou pas faire

Les contraintes techniques sont les suivantes :

- Coder avec le langage que vous voulez.
- Avoir un mécanisme de compilation et de création du binaire (une forme quelconque de Makefile).

- Le binaire s'appelle *42run* .
- Utiliser de l'OpenGL, et faire de l'OpenGL MODERNE : en version 4.0 minimum, avec des shaders obligatoirement.
- Vous pouvez utiliser des lib pour gerer le fenêtrage, pour loader des mesh et des images, pour faire vos calculs de matrices (glut, glfw, png, jpeg, ..). Vous pouvez utiliser votre libft, la lib math, la mlx avec son extension OpenGL.
- Vous ne pouvez pas utiliser de lib qui fait le gameplay (bref le boulot) à votre place.
- Le jeux doit être jouable sur les dump en salle machine.

Chapitre III

Bonus

Quelques bonus sont proposés :

- Un décor particulièrement chiadé (avec de la vraie 3d, pas comme dans ma démo moche)
- Des coins (ou des chatons) à ramasser en plus des obstacles à éviter
- Des power-up à attrapper qui donnent des pouvoirs particuliers
- Des missions à effectuer
- Glisser sous des obstacles en hauteur
- Trébucher
- Personnages différents avec skills différents
- Tout plein d'autres ajouts qui existent dans ce type de jeux
- Et il y a sûrement encore d'autres bonus que vous pouvez implémenter

Bon projet !

Chapitre IV

Démo



FIGURE IV.1 – Marvin dans l'entrée



FIGURE IV.2 – Marvin passe en salle de jeux vidéo



FIGURE IV.3 – Marvin doit éviter les obstacles en cluster