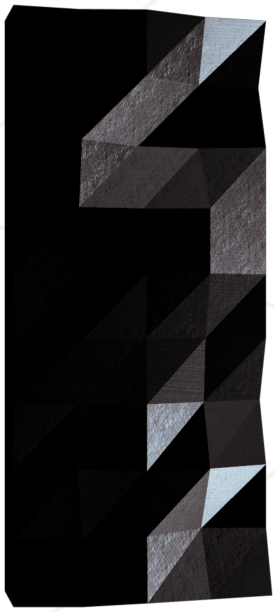

42run

Mini-projet d'Infographie

Summary: "In every good action movie, there is a scene when the hero steal the moto of the big bad guy, but then he must run away and survive the pursue. 42 Run, it s that scene. It s incredible."

Version:

Contents

Chapter I

Foreword

Great Scott! Marty, you're just not thinking fourth-dimensionally!!

Right, right. I have a real problem with that.

Chapter II

Subject

II.1 You were coding? Well, time to run now...

Did you enjoy temple run? Well, we are going to do the same thing. The pitch is a tad different: you have accidently touched Kwame's beloved bike, and so he is pissed. So you need to run! And fast. But careful now. . . many obstacles are in your way and you need to avoid them. How far will you get?

II.2 What you need to create

Your goal is to create a small program that will present an endless run (within the school walls) in 3D, while using the codes of temple run/temple run 2 gameplay. The program needs at minima to show the following elements:

- A set with a cool perspective.
- A set that moves forward to give an impression of movement.
- A randomly generated set using a limited number of 3D obstacles put together.
- A set inspired by the architectural elements of the school.
- A motionless character in depth that we can move laterally and jump.
- Obstacles to avoid, and to jump over, otherwise the game stops
- A distance meter

II.3 What you can or cannot do

The technical constraints are as follow:

- Choose the langage you want to use on this project
- Have a compilation mechanism and for the binary creation (some kinf of Makefile).
- The binary is called *42run* .

- Use OpenGL and use a MODERN Open GL: at the very least the 4.0 version with shaders, it s mandatory
- You can use any library you want, however you remain limited to use them for:
  - Load mesh and images
  - Compute your matrixes(glut, glfw, png, jpeg, ...)
  - Window management
- In this project, you are allowed to use all the libC (man 2) syscalls, as well as `malloc`, `free`, `perror`, `strerror`, `exit`, all the math lib functions (`-lm`), and all the `MinilibX` functions - or their equivalent in another graphic library. You'll have to recode a png or tga (or anything else you need) reader.
- You cannot use a library that does the gameplay (ie the work) for you.
- The game must be playable on the cluster's computers.

Chapter III

Bonus

Here are some ideas of possible bonuses:

- A particularly cool set (with a proper 3D, not like my shitty demo)
- Coins (or kittens) to be picked-up in addition to the obstacles to avoid
- Some Power-ups to be picked-up that give special powers
- Specific missions to accomplish
- Slide under some obstacles that are higher
- Trip while running
- Different characters with different skills
- All sorts of adds-on that exists in these type of games
- There is got to be more bonuses that you can implement

Good Luck!

Chapter IV

Demo

Figure IV.1: Marvin in the entrance

Figure IV.2: Marvin nearby the arena

Figure IV.3: Marvin must avoid obstacles