Projet Crossy road

Sacchetto Vladimir – Bellot Ian – Tregnago Hugo

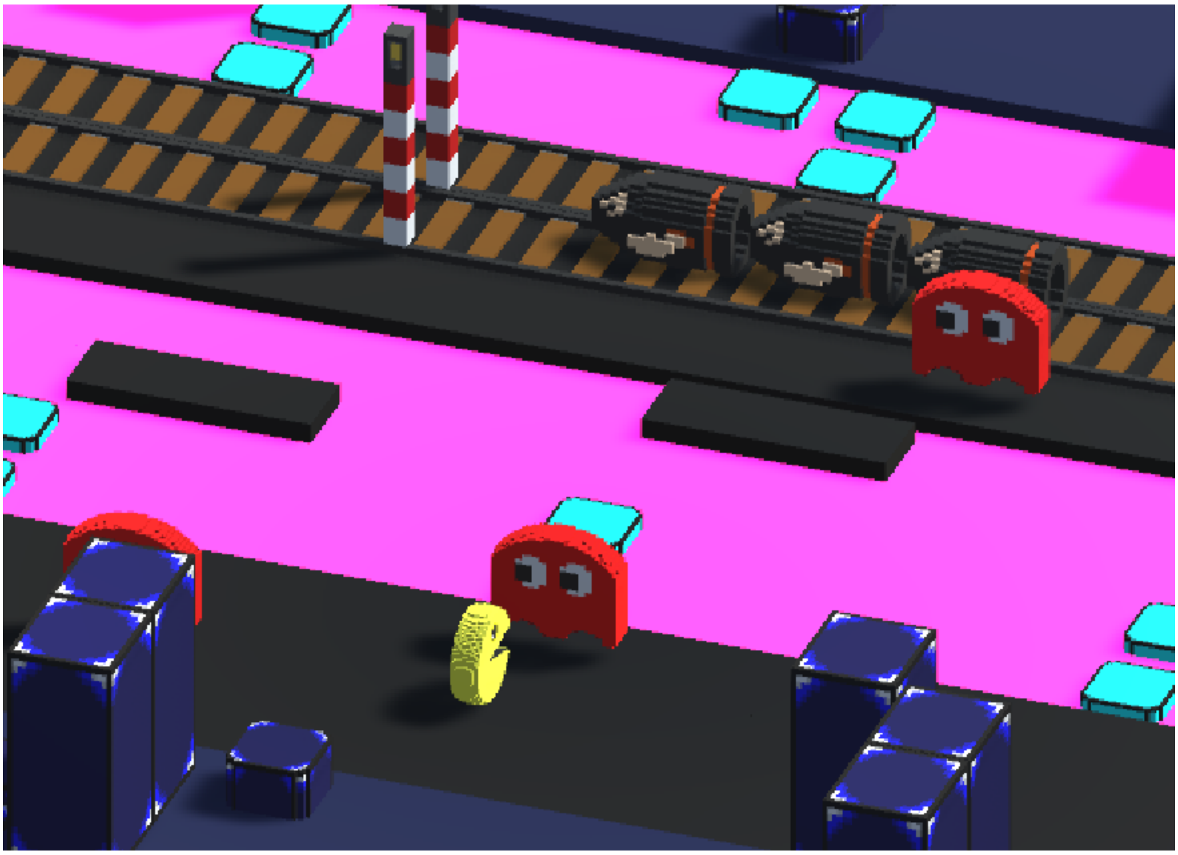


Table of contents

Introduction

Description of requirement

As part of an educational project, we have been asked to produce a video game based on the existing "Crossy Road" mobile game. The project is not intended to be commercialized, and has been created for purely educational purposes, to show two aspects of the computer engineering profession:

- object-oriented development
- Project management and general information systems methodology.

In addition to the basic Crossy Road gameplay with its score and time interface, we'll also be creating a score-saving system, an introduction menu, a pause menu and an option menu.

Educational objectives of the project

- **Master the fundamentals of programming :**
  Learn and strengthen students' programming skills

- **Object-oriented design :**
  Use object programming concepts wisely, structure your program properly

- **Use of UML analysis :**
  Use various analysis tools to better plan and manage your project as it develops.

- **Controlling source code and planning :**
  Use tools such as Git to version your code and create deliverables
- **Event and animation management** :
  Use events to control the character and menus, and integrate animations to enhance the user experience.

- **Game customization:**
  Students will be able to integrate new functionalities into the game to develop their creativity.

Scope and limitations

- **Technical constraints:**
  Developed in C# and Unity

- **Group constraints:**
  The project must be developed in a group. Our working group is made up of 3 people, who will have to divide the tasks between them.

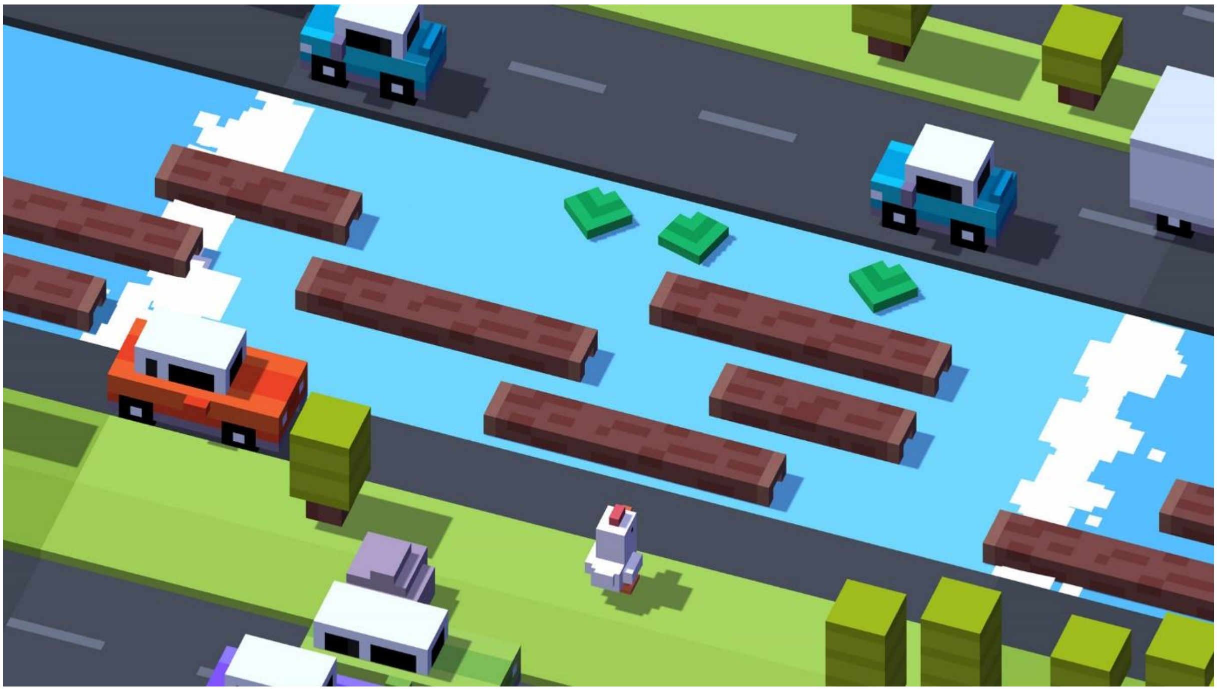
- **Time constraint:**
  The project must be handed in no later than 11:59 pm on 05/17/2024.

- **Deliverable constraint:**
  The project must be rendered as a Windows executable in the release branch of the source manager. Source code must be accessible.

Mains fonctions

Crossy Road's Gameplay

Game loop



Crossy Road is an infinite scrolling game, meaning that the end of the game depends solely on the character's death. The player controls the mascot as it moves through a procedurally-generated course littered with obstacles. The mascot can move in all 4 cardinal directions, and must avoid the following obstacles:

- Moving vehicles (cars, trucks, trains), which cause the character's immediate death.
- water, in which the mascot immediately drowns
- Non-moving objects (trees, rocks) that prevent the mascot from moving in that direction.

In the case of water, there are tree trunks that move according to the course of the water; if a mascot lands on them, they move along with the trunk. There are also water lilies that are fixed to the water.

The character is also forced to move forward. If there is no movement for a certain period of time, an eagle will attack and cause the end of the game.

Scoring and gold pieces

The player is encouraged to play again for two different reasons:

- To improve their score: the score is established according to the number of different lines the mascot has touched.
- To collect gold coins: gold coins are scattered around the field and can be used to pay for new game mascots.

Starting menu

When you launch the game, you can choose to play, select different difficulties, view the difficulty ranking or quit. This menu is also accessible after the mascot has died, if you choose to return to it, and also from the pause menu.

Pause menu

At any time, the player can pause the game to continue later, start from scratch or return to the start menu. This menu must be accessible at the touch of a button.

Settings menu

This menu allows you to activate or deactivate the game sound.

Pathfinding algorithm

A robot must be designed to play the game automatically, without requiring any input from the user. It will use a path-finding and collision avoidance algorithm (see game loop).