

Specification

[INSERT NAME]

[INSERT NAME]	Version: 0.1
Specification	Last modification: 07/04/19

Introduction	1
Presentation of the project	2
Context	2
Analysis of what already exists	2
Organisation and contributors	2
Objective and general goal	3
Functional description	4
Bull chart	4
Octopus diagram	4
Table of functionalities	5
Priorities	7
Planning	8
Description of problems	10
Assets and prerequisite	11
Knowledge	11
Ressources	11
Glossary	12

Introduction

This specification is the first document realized for this project. Its goal is to describe the context in which the project was carried out and the expectations regarding the game to be realized.

It has to present the different functionalities that the game will contain in its final state. It should establish a beginning of prioritization of the functionalities to be implemented for the development team according to the initial requests. This document also provides an overview of the project's problems and the skills already acquired and to be acquired to carry it out.

I. Presentation of the project

A. Context

As part of the program of EFREI PARIS (École Française en Électronique et Informatique), and as part of our second year in the undergraduate program, students of the international session had the choice to working either has a four or five EFREI student group, or collaborate with Coventry students on a transversal project with the purpose of solving a social or environmental problem.

In our case, we chose to work with Coventry students. The team is composed of three EFREI students as well as three Coventry students.

The team decided to implement a gamified learning experience.

The project has to be implemented for the end of the fourth semester. An oral presentation is asked.

B. Analysis of what already exists

As this project is a game, we will write it from scratch in C++. We will however use a set of tools to help us in the process of creating this game. We will use a game creation library called raylib, as well as a header only implementation of ECS called EnTT. We will also use a library to take care of the storage and retrieval of data as a json format. We will possibly use some code from past projects as inspiration for this one.

C. Organisation and contributors

Vincent Mouillon – Team Leader

Jasper Dahil – Lead Designer

Ibukunoluwa Adegoke – Designer

Victor Tailleu – Lead Programmer

Luca Vaio – Programmer

Vincent L. – Programmer

Jesse Prescott – Programmer

D. Objective and general goal

The objective of the project is to realize a game that will teach the user how to code. This will be a story-based game and the user will unlock the different lessons as he progresses in the game.

The objective of the project is to realize a game that will teach the user how to code and use common tools so that he becomes independent.

To do this, we are going to make a story game on data structures and other computer science related topics like variables, functions, the different major architectures, design patterns etc...

The user will have to retain the information. We will thus make him practice his reasoning skills and programming knowledge. To do so, we need to implement an editor inside our application with puzzles that the user will have to solve. We will consequently check his program with test functions. The user will be able to run his program.
if the user succeed in completing the puzzle, he will go to the next lesson of story based mission.

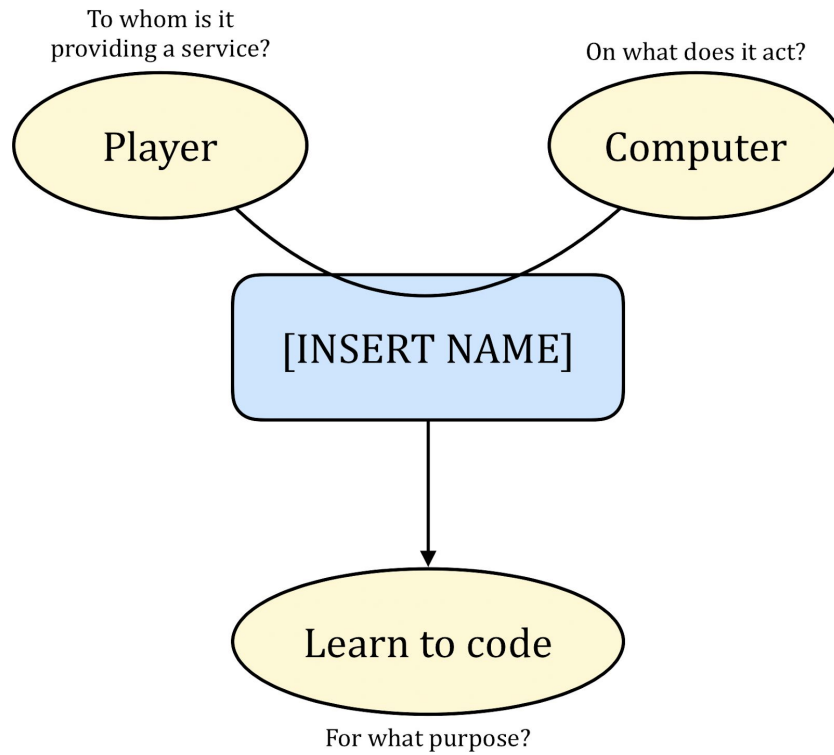
But why making a story out of it ?

We really think that making something involving helps in learning and retaining concepts. Making it really concrete and having tangible consequences to your actions helps remembering what you did.

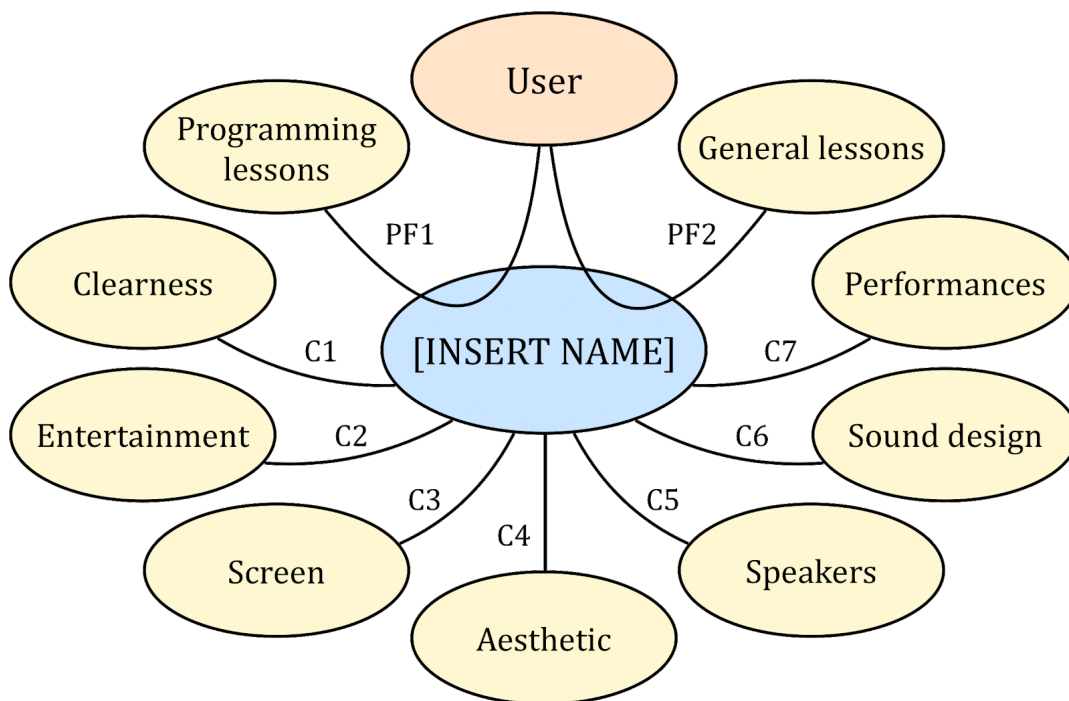
The code also needs to be flexible so that the addition of a new lessons or new gameplay won't cause important modification of the code.

II. Functional description

A. Bull chart



B. Octopus diagram



C. Table of functionalities

Name	Functions	Criteria	Implementation
PF1	Teach the user how to code	<ul style="list-style-type: none"> - Teach a programming language syntax - Make an editor - Test the user's code or knowledge - Teach the user data structures - Teach the user major program architectures 	<ul style="list-style-type: none"> - Teach C - Be able to write in a text file or drag and dropping the file the user created - Test functions for each data structures - Teach: <ul style="list-style-type: none"> - Variables - Pointers - Arrays - Multidimensional arrays - SLLs (singly linked list) - DLLs (doubly linked list) - Binary Trees - Balanced Binary Trees - Search Binary Trees - AVLs - Trees with an "infinite" number of children (acyclic directed graph) - Graphs - Teach: <ul style="list-style-type: none"> - OOP - ECS - Functional Programming - Programming patterns
PF2	Make the user independant	<ul style="list-style-type: none"> - Present the user the major editors, IDEs, languages and tools 	<ul style="list-style-type: none"> - Initiate to Emacs, Vim, Sublime, Atom, VS code - Initiate to Visual Studio, Code::Blocks, Clion - C/C++, Python, Java, Haskell - gcc, gdb, valgrind, make, cmake, raylib, entt, SDL
C1	The actions proposed to the player has to be clear and coherent	<ul style="list-style-type: none"> - Have a good tutorial - The lessons need to be really clear - Help menu 	<ul style="list-style-type: none"> - Having a help and hint menu for any mission - Having an intuitive AI
C2	The actions proposed to the player	<ul style="list-style-type: none"> - The lessons must be playful and enjoyable for the user 	<ul style="list-style-type: none"> - Develop at least 1 story driven missions at the beginning of each world

	has to be enjoyable and entertaining	- There must be a story to keep the players attention	
C3	Display the game to the screen correctly	- No display bug - Screen option menu - Support common screen resolutions	- Two modes: windowed and fullscreen - Support 720p 1080p
C4	Has to be visually attractive	- Good graphics - Interesting Visual design - Clear UI/UX	- Unifying style - 1 texture for each data structure - 1 background for each world
C5	Playing music and effect correctly through the speakers	- No sound bug - Sound option menu	- Sound and music options
C6	Has to be auditively pleasing	- Good sound effects - Good ambiance music	- 2/3 ambiant musics - at least 10 sound effects
C7	Has to perform well	- The game must be playable	- Always more than 60 frames per seconds
C8	Adapt to the starting knowledge of the player	- have an option to skip certain steps without skipping the story	- A test step at the end of each world accessible from the point of entering the world

III. Priorities

The C1 constraint will be addressed immediately as it has to be addressed at the design stage. The C3 constraint should be kept in mind during the developpement time of the project. The several modes of rendering shouldn't be a priority at first. The PF1 function is separated into multiple parts for the ease of implementation. The lessons shall be implemented in the correct order. The C2 class shouldn't be forgotten at the design stage as it will be impacting on the all game.

IV. Planning

Function to implement	Contributors	State of advancement	To-Do	Deadline	Priority
PF1 Teach the user	Vincent Mouillon Vincent Lim	We know what we'll be teaching	-Writing the scripts	27/05/19	very High
PF2 make the user independent	Vincent Mouillon Vincent Lim	We know what we'll be teaching	-Writing the scripts	27/05/19	Medium/ High
C1 Coherent experience	Vincent Mouillon Vincent Lim	Nothing	-Designing menus -Designing the missions	27/05/19	High Design stage
C2 Enjoyable experience	Vincent Mouillon Vincent Lim	Nothing	-Verifying that the lessons aren't repetitive and are entertaining -Involve the user into the teaching experience	27/05/19	High Design stage
C3 Display to screen	Vincent Mouillon Vincent Lim Victor Taillieu Luca Vaio Ibukunoluwa Adegoke Jesse Prescott	Added and compiled raylib on all system. displayed level select screen	-Implement each screen with raylib -Adding a video option menu	27/05/19	very High
C4 Attractive look	Jasper Dahil Ibukunoluwa Adegoke	Designed the intro screen and animation Designed the level select screen	-Designing the different screens -Create image assets for the game such as characters, buttons, backgrounds	27/05/19	Medium/ High
C5	Vincent	use of raylib as a	Implement	27/05/19	Medium

Sound	Mouillon Vincent Lim Victor Taillieu Luca Vaio Ibukunoluwa Adegoke Jesse Prescott	sound library	sound in each screen adding a sound option menu		
C6 Attractive sound	Vincent Mouillon Vincent Lim Luca Vaio	found a DAW (LMMS) and multiple free VST plugins (LABS, Komplete Start) Started to work on a synthwave background music	Create sound assets for the game such as music and FX	27/05/19	Low/Medium
C7 Performance	Vincent Mouillon Vincent Lim Victor Taillieu Luca Vaio Ibukunoluwa Adegoke Jesse Prescott	Already done by architecture if no abuse is made with an ECS design and the use of a dedicated fast graphical library (raylib)	Make sure that the screens are optimized enough	27/05/19	Very low Design stage
C8 Adapt to player	Vincent Mouillon Vincent Lim	Designed what the process could be to enable a player to proceed to further levels while skipping unnecessary lessons	Designing the tests of each world	27/05/19	Low

V. Description of problems

There is multiple problems that are brought up by this project:

- How to keep the player interested during the all experience:
 - The experience shall be compelling and fun but this is really hard to do for lessons.
 - The story should be emotionally realistic which is really difficult to capture.
 - The player should have goals.
- How to do a reasonable editor considering the time limit.
- How to implement fun and interactive lessons for multiple data structures without doing a tutorial each lesson.
- How to do reasonable assets considering the time limit.
- How to separate work for six people in such manner that everyone has the same amount of work to do.

VI. Assets and prerequisite

A. Knowledge

Has students in Computer Science, we were taught how to code in the C language. The coventry students did a course on data structures and on C++. As EFREI students, we were taught the basis of C++ and Object Oriented Programming during the last few weeks.

The team will have to learn how to use Raylib as well as EnTT to be able to work on this project. They will also have to be trained on OOP and data structures as well as ECS design.

The team will also have to learn how to cooperate efficiently to finish the project in time.

B. Ressources

A budget will be needed for hosting the game installer and/or the embedded game on a site.

VII. Glossary