Synopsis: Projet 2A S3 SDD

EFREI 2022

Chicky Graphs

Team Tardigradus

Gabriela de Almeida — Teaching Manager
Chloé Carayon — Communication Manager
Vincent Mouillon — Project Manager
Victor Taillieu — Technical Manager
Luca Vaio — Design Manager

Figure 1: Chicky Graphs logo

November 2, 2018

Contents

1 Analysis

1.1 Goals

This project is intended to make us imagine and conceptualize new solutions, to get us involved into something else than only computer programming. Of course, we must rely on the skills acquired during the lessons of this first semester, that is to say C data structures such as lists and trees.

By realizing this project during three months, we will learn different knowledge on several subjects. It will evolve in order to produce a useful project and learn how to communicate on it.

1.2 Subject

We had in mind a project aimed to teach and be useful to people. To make the project even more interesting, the subject should appear to be slightly complicated, which will lead to a real apprenticeship.

1.3 Goals of Chicky Graphs

Our idea is to develop a learning software dedicated to learning how graphs work and how to use them. A tree is in fact a particular graph, the latter being a bit more complicated but with many applications and which represents an unknown subject for most people. Thus, that is a way for us to approach this notion and transmit what we learned through the different stages of creation of this project.

1.4 Challenges

Also, our project should please people, so it needs to be playful and interactive, well designed and done so that the user doesn't get lost and enjoys using it: the software should draw the user's attention at first sight. Thereby, through this project we will step in both students' and teachers' shoes, thus leading us to confront ourselves to the difficulties of teaching and point them out to maximize the user experience. Finally, our project could be used as a learning tool to help students through their learning.

2 Conception

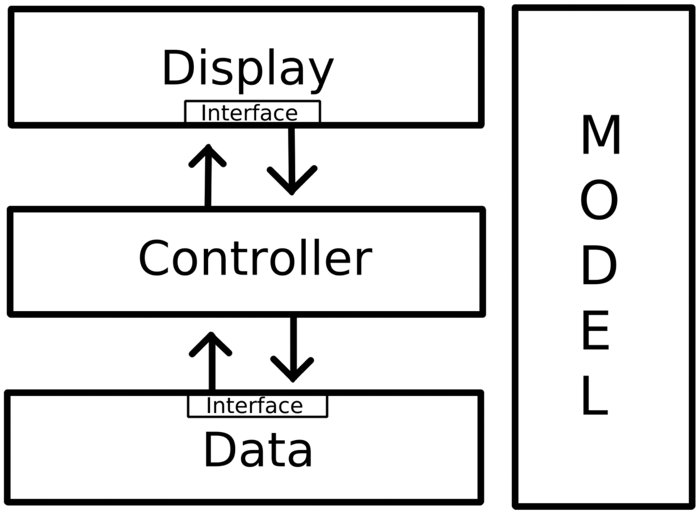


Figure 2: Structure of our code

2.1 Organization

After we had the final idea, we worked on the architecture of the project. We opted for an MVC architecture (Model View Controller):

There is the Display which is the input and output of our program. This Design part will represent 40% of our work where we will achieve the graphical interface.

Then, there is a Data interface that takes care of transforming the assets, the data, in data structures that the controller can take care of. The Data part will be 20% of our work. The model is here to define the different data structures that will be used in our code, while the Controller manages the data by transforming it or updating it. This last part will be the brain of our program and will represent 40% of our work.

2.2 Solutions

We will be using SDL2 for the graphical interface because of the better performances and text management. Our gameplay loop will be composed of a function to handle inputs and events in general. Then, another one updating the data and a last one to render the result to the screen.

The data interface will have a save and load feature to save the state of the game, or the current state of the sandbox. Another function will be here to load the data for a mission such as its text and its different requirements for the sandbox as well as other useful assets. The graphical assets will be transformed into usable textures with SDL2 and stored in a tree with multiple children, working like a succession of folders to be more flexible.

So, we will have to implement and manage a graph, a tree with multiple children. Thus, we will also need to implement a simply linked list for the children of a certain node. The graph will be the main part of our program since this is what we want to teach. we will have to display the graph with its edges, as well as useful information for the user. Another problem is to make good gamified teaching materials capable of providing the user with useful information about graphs without losing its attention.

3 Plan

3.1 First Phase

Our first goal is to present graph notions using several exercises (called missions here), with increasing difficulties. Firstly, we will create the Sandbox in order to accomplish missions to learn basic information on graph.

In the Basic Sandbox, we will define and manage basic properties of graphs as the creation, the deletion and the addition of nodes and edges, using our specific functions. This Basic Sandbox will be used for the missions, where the user will have to do interactions to learn graphs, and also for the "free Sandbox" where he will be allowed to freely create graphs, use what they learned and play with all of their possibilities.

This first step will be quite hard because it will have to perfectly work and it will take more than two weeks for the development and for the graphical interface. We will have to be able to represent them in a great design.

3.2 Second Phase

Then, we will focus on the Menu, thanks to which the user will be able to access the missions and therefore the lessons part, to use the "free Sandbox" and to access to backups.

We will increase the functionalities of the Sandbox by applying weights on the edge between nodes or by adding probabilities to our graphs. This higher level will allow to present an even more interactive platform by offering more difficult missions in order to develop the technical skills of the user. The three specific parts - Controller, Display and Data - will have to work hard on it more than two weeks. By adding those new functionalities to the "Free SandBox", the users will have a space where they will be able to freely test their knowledge.

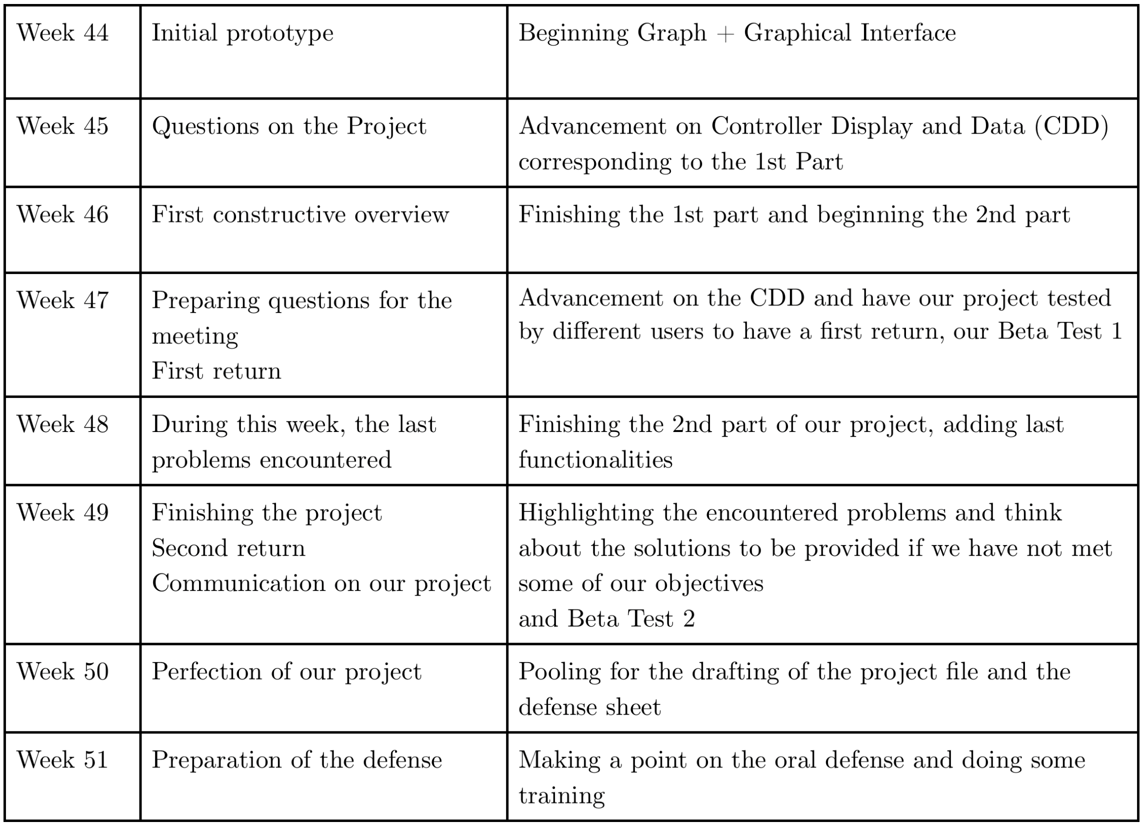
Moreover, we will develop a story which will be in the heart of the application in order to introduce several missions : This will attract the user.

3.3 Third Phase

Thirdly, we will develop a harder Sandbox, it will teach the user to realize a neural network during the last missions. And it will be also possible with the "Free Sandbox".

3.4 Schedule

In this view of our project, we will do one meeting a week to focus on the work done during the week and the objectives to accomplish for the next week using our Trello with all of the work to accomplish and our Discord to assign tasks. In order to present at the end of week 51 an E-learning Platform, we will work in short sprints of one week long. We will try to follow this organization with weekly meetings:



Week 44	Initial prototype	Beginning Graph + Graphical Interface
Week 45	Questions on the Project	Advancement on Controller Display and Data (CDD) corresponding to the 1st Part
Week 46	First constructive overview	Finishing the 1st part and beginning the 2nd part
Week 47	Preparing questions for the meeting First return	Advancement on the CDD and have our project tested by different users to have a first return, our Beta Test 1
Week 48	During this week, the last problems encountered	Finishing the 2nd part of our project, adding last functionalities
Week 49	Finishing the project Second return Communication on our project	Highlighting the encountered problems and think about the solutions to be provided if we have not met some of our objectives and Beta Test 2
Week 50	Perfection of our project	Pooling for the drafting of the project file and the defense sheet
Week 51	Preparation of the defense	Making a point on the oral defense and doing some training

Figure 3: Schedule

4 Initial Prototype

4.1 Basic Prototype without interface

Our project is mainly based on graphs so the first version of our program should be able to manage them using basic functions. We will work on how to represent graphs in the program so that it is easy to use and to implement in the project. The prototype will contain several functions performing specific actions such as creating a node, linking or unlinking two nodes, deleting a node, set and get the data from a node. With these basic operations, the prototype will allow us to make tests and prepare future improvements. The very first version will work on the console to focus on the algorithmic issues.

4.2 Prototype with a beginning of interface and an operation

After that, we will begin to learn how to use a graphical interface (in our case SDL) because we did not all use it previously. We will have to create a window, display images, receive mouse input, treat the different actions and display the results. This version will handle basic graphs with no operation and no weight on the edges. The user will be able to create himself a graph of the form he wants by creating nodes and connecting them. When these options work well, we will implement new features like a contagion. A node will have an information to transmit and all his neighbours connected with him will have the information after starting the simulation. This will be a good starting point for our project.

4.3 Roles for the prototype

If we had to realize this prototype for tomorrow, we would spread the work as followed. Victor would have to create graph structures and functions to handle them, this will be part of the controller. Luca and Vincent would manage the graphical interface, respectively implementing SDL and assets for the visual aspect. Chloé and Gabriela would create the data and work on how it is loaded by the program. They would also help Victor on the technical part. Finally, this prototype will be a solid base for the project and its future improvements.

List of Figures