Practice /30

Dr. Azanzi Jiomekong

Master 1 in computer science
Mobile application development

Objective: The main goal of the overall exercise is to allow students to understand how to use OOP techniques to develop software. On the other hand, they will learn how to implement software using Object Oriented Programming techniques. Thus, firstly, the students will collect data on some of their daily activities. Thereafter, they will use this data to design and develop an application.

NB: The overall exercises are out of more than 30. However, students' marks will be maintained out of 30. For students having more than 30, the rest of their marks will be used for the continuous assessment bonuses.

Evaluation: Wednesday, 04th January 2023

Exercise 0: setting up the development environment **(3pts)**

1- Set up your Operating System (Linux) **0.5pt**

2- Make a table of your development environment for each programming language involved in this exercise **0.25x4=1pt**

3- Install your development environment **0.25x4=1pt**

4- Write the "Hello World" program in your new development environment **0.5pt**

Exercice 1: Food Image annotation **(15pts)**

Adequate nutrition is an essential catalyst for economic and human development as well as for achieving Sustainable Development Goals - Goal 2: Zero Hunger and Goal 3: Ensure healthy lives and promote well-being for all at all ages. However, understanding Food information can allow people to have a healthy diet. To this end, food information engineering involves the acquisition, the processing and the diffusion of up-to-date food information to different stakeholders. These information are compiled from several data sources and used for a variety of purposes such as food recommendation, recipe substitution, food image recommendation, nutritional agendate, etc. To recommend a food to a person, a task can consist of first recognizing this food from a food image. To this end, Food images should be annotated.

Food image annotation or tagging is the assignment of metadata in the form of captioning or keywords to a digital food image or a set of food images (for instance to name a dish). Actually, many dishes may contain different kinds of foods, making it difficult to identify the food contained in a dishe.

The goal of this exercise is to apply the OOP techniques to annotate and recognize African food images. To this end, a set of images should be downloaded, relevant objects identified in each image. The annotators should identify specific objects, segment an image into relevant regions, which are specific points of interest in the food image. To ensure labeling accuracy, each member of the group should contribute to labeling the same image. Thereafter, the image should be annotated by indicating the shape and label of each food item in the dishe and the name of the dishe.

This work can be done individually or by a group of 2-5 students. Each student or group of students should execute the following tasks:

1- Choose one country in Africa and create an image dataset of the food eaten in this country (at least 50 images per person) **0.5pt**

2- From each dish, identify food objects and build an object diagram **1pt**

3- Use the question 2 to organize foods into classes and determine the relations that are between these classes and build the class diagram **0.5+0.5+0.5 = 1.5pt**

4- From questions 2 and 3, and a tool of your choice (for instance Computer Vision Annotation Tool) to Identify objects in images and use lines, polygons, or markers to draw lines along the object's perimeter or circumference. **4pts**

Img_id	Dish name	Objects found (dish content) Ingredients Source		
		Author	Title	Link

5- Using the question 4, propose a simple mobile application for food image annotation **4pts** Given to the question 4 and 5, you decided to build a system for food recognition

6- Complete your development environment with new tools that you need (give the role of each tool) **2pt**

7- Define and train your model using images annotated in question 4 and the tools of question 2 **4pts** 8- Use the results of question 6 to build a mobile application for food recommendation **4pts** 9- Evaluate your application of some foods images **2pts**

Exercice 2: Nutritional agenda **(35pts)**

Following exercise 1, the goal of this exercise is to collect information on your food habit and use it to recommend healthy behavior. Actually, the Gastrointestinal (GI) track consists of the mouth, stomach, and intestines. Together with the liver, gallbladder, and pancreas, these organs work together to absorb nutrients and expel waste. Disturbances to this process can cause a range of health problems such as bloating, cramping, gas, abdominal pain, diarrhea and constipation, etc. The goal of this exercise is to build a tool that will help people know which food causes digestive disorders or allergies. To this end, the following tasks should be done:

1- Collect data on the food you are eating in a daily basis by filling the following table **2pts**

Date (Monday xxx)	Foods eaten	NB times food eaten per day	Quantity of water drank	Quantity of other liquid drank	Eating fruits and legumes	NB bowel movement	Health problem

Consider that the food eaten can be organized with an heterogeneous graph in which the nodes represent foods, and his digestive consequence.

2- Propose a mobile application that can be used to facilitate the work done in question 1 (define the architecture and propose the application) - This application should integrate a dashboard describing your eating habit **4pts**

3- Propose the graph modeling what you eat in a daily basis **1.5pt**

4- Propose a mobile application implementing the graph of the question 3 **5pts** 2

University of Yaounde 1
Department of Computer Science

algorithm that can be used to predict the food that you'll eat **1pt**

5- Use the graph of question 2 to propose a model of your eating habit in a daily basis **1.5pt** 6- Propose an

7- Propose an algorithm that can be used to describe your eating habit (this is a graph transversal algorithm) **1pt**

If we consider that our body is a bag that should be filled with food1 (x1 quantity), food2 (x2 quantity), food3 (x3 quantity), water (y quantity). Xi is the maximum quantity of foodi that the body should contain.

8- Propose the fill max bad program that can allow you to have healthy eating habit **1pt** 9- Implement an algorithm which predict given to your nutritional agenda what you should eat **5pts**

10- Propose a mobile application which tell you if a food from a food image is safe for you **5pts**

11- To develop your search engine that will provide a reliable solution, you decided to identify all the types of questions that can be asked in your system. These questions can be generally classified based on Wh-terms (Who, What, When, Where, Which, Whom, Whose, Why) and other terms such as How many, Did, etc. These questions have as answers granular answer type classification. Thus, for each question, the answer category ("resource"/"literal"/"boolean") and answer type (for literal, we have "number", "date", or "string" - for boolean category, we have the "boolean" answer type. Concerning relations identification, it should be noted that relations are implicit or explicitly represented in the question and should also be identified. Entities are existing or real things. Fill the following table with the question and answer category, type, relation, entity. **(10pts)**

Each question = 0.01 pt

Question	Question type	Answer category	Answer type	Relation	Entity

Question type = Simple question, Comparison question and Constraint question

Simple question: "How much sugar is in honey?"

Comparison questions: "Which food among koki, hero, couscous has less energy?"

Constraint question: “What Cameroonian dishes can I make with water and corn?”

12- Use the question 11 to build the search engine of your mobile application **(10pts)**

Exercice 3: Surviving in a competitive environment **(10pts)**

This is the game of life in which we consider two types of entities: the mouse and the cat. The mouse and the cat forage, fight and bargain their way through procedurally generated environments to outcompete other mice and cats trying to do the same. We consider in this game that the world has a beginning and an end that can be parameterized during the game initialization. At its end, a dashboard presenting the entities and their rest of life. The goal is to simulate populations of agents in procedurally generated virtual worlds. It is inspired by classic massively multiagent online role-playing games. We consider that:

- The game starts with a map made up of NxN tiles (parameterized at the beginning of the game); ● In each tile, only one agent can be present. In some cases, two agents of the same species can be present only if one is a male and another one a female. In this case, in a certain number of times, these agents give birth to babies and all the agents move to the adjacent tiles, leaving the female in the current tile.

- The maize appear randomly on the tiles ● The mice and the cats should eat to survive ● The cats eat the mice and the mice eat maize
- If a cat and a mouse meet in a tile, the cat will kill the mouse and if the cat is not full, it will eat him ● If any entity did not eat for a certain time, it will die
- Two entity of the same sex should not meet in the same tile
- Each entity can make multiple actions: move, eat, reproduce.
- Each entity can observe NxN tiles around him
- If an entity move out the map, it will die

To build this game, reply to the following questions:

1- Draw a map made up of 25 tiles, 3 cats and 5 mices **1pt**

2- Simulate the game with the environment of question 1 (you can use activity diagram or sequence diagram, etc.) **1pt**

3- From the picture of question 1, identify objects and build an object diagram - for each object, identify the messages that it can send or receive **1pt**

4- Use the question 3 to propose the class diagram **1pt**

5- Build the state-machine diagram of the main objects of the system **1pt**

6- Implement the classes of question 4 in Java, C++, Python and JS **2pts**

7- Implement the whole application **3pts**