INTRODUCTION TO VISUAL AND INTERACTIVE PROGRAMMING ASSIGNMENT

**** GROUP PROJECT ****

INTAKE : UCFF2308CTODL

LECTURER : MOHD NAJMUDDIN SUKI

DUE DATE : Week 12 (Application and Documentation)

CLO2	Apply the elements of computational thinking to solve a problem (C3, PLO2)	Project-Documentation
CLO3	Use the visual interactive programming tools to develop an application. (A1, PLO6)	Project - Implementation

1.0 ASSESSMENT DETAILS:

GROUP PROJECT (50%)

Work within a team to create a visual and interactive application using SNAP! based on the following guidelines:

1. You are required to create an interactive visual application based on the case study below.
2. You are advised to use your own logic flow to create the application.
3. Interactive **game** applications are **NOT** allowed.
4. You are required to form a team (a group of **FOUR (4)** persons).
5. Read the given case study thoroughly and come up with your solution.

CASE STUDY: HOTEL PREVIEW SYSTEM

Program Specification:

You are required to develop a Hotel Preview application to advertise the hotel and help guests estimate the cost of their stay. The application will showcase the brand and facilities before proceeding to help with hotel rooms info, choosing of rooms and calculation of estimated cost of stay.

The end user for this application will be the **Prospective Guests** of the hotel, who can learn more and plan their stay. Your weekly lessons will be useful in implementing the characteristics of the application as listed below:

Part 1: Showcasing the brand and facilities

Create an advertisement where you showcase the hotel brand name and the facilities available in the hotel. The advertisement will flow in such a way that the ending would be a call to action to choose the room;

Part 2: *View* rooms info, *Choose* rooms, *Calculation* and *Commit*

1. Users need to enter the booking room details. Your application should provide customers with hotel room information.
2. Users will be able to choose date(s) and hotel room(s).
3. The system will be able to store up to **TEN (10)** different dates of booking.
4. The estimated cost will be shown once the user has chosen the date(s) and room(s).
5. Once the user have learnt the cost, the system will provide the choice of committing (provide info to proceed with booking) or exiting.

DO NOT PANIC: You do not have to be an artist or a designer to complete the project.

Do's	Don'ts
Make it humorous	No blood & guts, violence, loping heads or/and missing limbs
Add details & make your application more expressive	No jumpy application, unless it's intentional and a small part of the overall application
Use ALL skills learned	No inappropriate humor touching on 3R or/and sexual in nature
Different scene as background images that develop your user interface	No stick figures. (Try to be more creative in the use of ALL tools)

Select **ONE (1)** person from your group to be the group leader. As a Group Leader, you are required to:

- Monitor and manage the works and tasks among your group members.
- Ensure the whole application is functioning along with the full completion of the documentation.
- Compile all your group tasks and works into a complete set of documentation.
- Report to the lecturer if any group member is not being cooperative in the group.

PART A – APPLICATION (40%)

Your application **MUST** include each of the following:

Interactivity. The application must be interactive. Users must be able to interact with your application. When there are animation(s), make sure there is a degree of control on the part of the user.

Technical Requirements:

- **Sprites –** Your application must have multiple characters(sprites), **not including** any sprites used to give instructions. **Do not use** *Alonzo (BYOB), Cat (Scratch)* or *Turtle (Snap)* sprite. You need to create or edit at least **ONE (1)** sprite.
- **Stage / Scene** – Have more than **TWO (2)** a different background.
- **Repetition** – Your application must have at least **TWO (2)** loops with a variable other than **"forever"** somewhere in the project.
- **Decision Making** - Your application must have at least **TWO (2) "if"** or **"if-else"** blocks. It is **compulsory** to have **nested if** with a variable.
- **Audio -** Any number of sounds is fine; the **MINIMUM** is **ONE**. Note that sound is important - if no one can hear it, no one will use your application.
- **Data representation** – Use appropriate blocks to enable the operation of variables or lists..
- **Operators** - Your application must consist of operators.
- **Events** – Use more than TWO (2) types of "event-handler".
- **Broadcast** - Your application must consist of at least **ONE** broadcast message.
- **Make a Block** - Your application must consist of at least **ONE** custom block with parameters and should be called accordingly.

Additional Technical Requirements:

- **Input Data** – Your application must have part(s) where there is interaction with users.
- **Bug free**:

1. Users must be able to operate all parts of the application without crashing or exhibiting unworkable behaviour or other hang-ups.
2. Students should test every part of their application to be sure there are no bugs and that it is fully functional.
3. Things to be consider:
    - Is it easy to play or use?
    - Does everything work?
    - Be alert to the problems that might be caused by unexpected user behaviour.
    - **Validation/Exception handling:** When the system asks "Yes" or "No" question (y/n?) questions, what will happen if the user keys in a number 9?
    - **User friendliness:** Include the instruction on how to operate your application.

The application should follow the **RULES** as below; ***(any violation may result in the reward of ZERO (0) mark for the development team)***

1. No vulgarism
2. No inappropriate graphics and contents
3. Application must be fully working, properly terminated and no errors.

PART B: DOCUMENTATION (10%)

The code must be documented appropriately. Prepare relevant documentation based on your application. Your documentation should consist of the following requirements:

- Assignment Cover Page
- Table of Contents consisting of:
    - **Introduction (including the title)**
    - **Stage 1 – Analysis**
        - Objective(s) of the application
        - Target users
        - Hardware, software, skills required to run and use the application
        - Gantt chart for application development
    - **Stage 2: Design**
        a) **Summary of Characters:** Describe each sprite's functionality
        b) **Initial Planning:** Come up with a wireframe for the system
        c) **Detailed Planning:** Design a pseudocode or/and flowchart
        d) **Showcase the Flow:** Storyboard of the application showcasing an "ideal run"
    - **Stage 3: Development**
        a) Programing Concepts with source code for explanation
    - **Stage 4: Implementation**
        a) User manual for the application
        b) Screenshots of sample input/output and explanation for an "ideal run"
        c) Testing results of different scenarios including non-ideal events
    - **Conclusion - Application capabilities and limitations.**

- **References** (**MINIMUM OF 3** references using **APA-Referencing Style**)
    - All information, figures and diagrams obtained from external sources must be referenced.
    - **Reminder:** Plagiarism is a serious offense, and your work will automatically be awarded **zero (0)** marks.

- **Document formatting:**

  Spacing: 1.5spacing

  Cover page: Font Size: up to 16pts; bold

  Heading or title: Font size: up to 14pts; bold and underline

  Flowchart / pseudocode: Font-style: century gothic; font-size: 10pts

  Documentation Content: Font style: Time News Roman; Font size: 12pts;

  Alignment: Justify

2.0 SUBMISSION REQUIREMENT

- **ONE (1)** team member will submit the work (documentation and the working application) in SNAP! Via Moodle. Submission methods other than via Moodle is **INVALID**.
- Last submission due date: **9th February 2024 (Friday), 6:00 p.m.**
- **Late submission** will result in **marks deduction** for the development team as follows:
  - **One day** (within 24 hours): **30% deduction**
  - **One week** (within 144 hours after the 24 hours): **50% deduction**
  - **After one week** (>168 hours after original submission date): **100% deduction**

3.0 DELIVERABLES:

Component	PIC	How to submit?	File Name
Group Project	Group Leader	1. DOC/DOCX file via Moodle. 2. Application file (link or xml) via Moodle	*IVIP_* UCFF2308CTODL *_GroupName*

4.0 Marking Criteria

Group Project (50%)	
Component	**Description**
PART A: APPLICATION (40%)	
Code Readability	The source code is well explained through the instruction provided in the project
Novelty	Creative and has a unique design.
Storyline	The project has a clearly developed plot with an obvious beginning, middle and end.
Use of visuals	Multiple sprites with changing costumes and appropriate naming for all sprites in the project. Relevant background design.
Motion of Sprites (Animation)	Animation(motion) is used effectively.
Use of Media (Sound)	Uses more than two types of media components.
User Interactivity	Uses more than two types of event handlers (Keyboard interactive) or/and excellent user prompts and the program responds to them well.
Use of Procedures - custom block	Custom block defined with parameter(s) and called.
Use of Procedures - broadcast message	Broadcast messages were used effectively (including the use of run / launch).
Conditional	Uses more than one “if then else” involving variables and expressions in the condition and it is working effectively.
Loops	Uses complex loops or “for each” loops with list items.
Use of Data (Variable or List)	Variables were created with meaningful names, initialized correctly, and used appropriately. List were defined and used appropriately.
Runtimes	Executes without errors.
Program termination	The program is able to terminate according to the program instruction.

Component	Breakdown	Description
Part B: Documentation (10%)		
Content	Objective	This includes accuracy and relevancy of facts and information with good use of language and grammar. Clear objective, with details on the purpose of the program and who will be the target user
	Target user	
	Hardware, software, skills	
	Gantt chart	
	Design	Very clear, understandable, readable, and organized. Pseudocode covers more than 80% of system requirements with correct logic/solution. Flowchart covers more than 80% of system requirements with correct logic/solution and shapes.
	Development	Program clarity, coding style, variable naming, use of modular programming technique, ability to implement required criteria and mapped according to produced design.
	Implementation	Completed all the testing. Correctness of program (i.e. ability to produce intended output).
	Conclusion	Accurately and appropriately ties with the whole purpose and implementation of the application.
	References	Professional formatting with the use of APA referencing style.