ICS3UC CPT – Culminating Project

Introduction

Your CPT will be a review of all the skills your have acquired over the course. The goal is to create a program that demonstrates as many of those skills as you can.

Requirements

Design a program using PyGame. It could be more elaborate adventure game (combining text and graphics), it could be a simple 2D video game or a computer simulation (interactive visualization? Data analysis?).

Use your imagination, keeping in mind that the purpose is to highlight whatever skills you have acquired through the course. The graphics don't have to be perfect or the story extravagant, these are just the context for you to demonstrate your programming skills.

If the program looks great, but there is very little technical skill in making it, you could get a lower evaluation than one with poor looking graphics but has the needed skill mastery in the code. Remember good code is not about how many lines you have, but how well it's designed, structured, documented etc.….

You will want to review the rubric in the Evaluation section on this document (slightly different for 3U and 3C) to see what skills are required as this might guide your decision of what to create.

There will also be a **theme** selected that all projects must incorporate in some manner into the project. A twist to allow you to get a little creative and ensure we get getting something unique. This will be announced the day of the project introduction.

Overall Timeline/Milestones:

- Wed Jun 3 – Introduction to project
- Fri Jun 5 – Design Proposal is due
- Fri Jun 12 – Checkpoint #1 - Draft implementation
- Wed Jun 17 – Checkpoint #2 – Working Implementation
- Fri Jun 19 – Final Implementation and Report Due
- Thurs/Fri Jun 25-26 – Feedback sessions

Checkpoints does not have to be finished code and are not evaluated. They are meant to show a progression of your work throughout the time period. Basically, you will be required to hand in whatever you have so far, and is an opportunity to get some feedback.

All milestones must be adhered to and be a continuation of the work in order to ensure authentication. Sudden surges in progress or last-minute grids at the end may invalidate the project as we cannot ensure the work is your own.

Milestone #1: Design proposal

You must complete a written proposal of your CPT project. A short one-page description of what you are planning to implement. Some of the things you might want to highlight:

- Overall idea for the program
- How the class theme is incorporated
- What coding paradigms you plan to use (Procedural, Objects or Sprites)
- What is the user interactions (mouse, keyboard etc…)
- How many levels, characters, objects will be included?
- Suggest you break it down into "for sure" elements and "if I have time" ones.
- You may want to highlight if there are certain parts of the rubric that you are not planning on trying to achieve (use of Objects for example).

Some of your elements may change throughout the project (added or removed) but the once submitted you will not be allowed to change your overall proposal.

Checkpoint #1 - Draft implementation

First checkpoint is a draft of your implementation. Hopefully you will have some features working and likely others in various states of development. It doesn't necessarily have to run but hopefully you should be able to "comment out" part that are not complete and still have something that runs.

Checkpoint #2 - Working implementation

This checkpoint you will need to append the current working version of your project. At this point it should be a working program, might still have some problems, in need of clean-up etc... From now forward it should be considered "feature frozen" meaning the last few days are spent fixing what you have and cleaning up the appropriate documentation and report that will go with it. You are not trying to add new features to the program, just fixing up what you have.

Final Implementation and Report

The final milestone is for you to hand in your completed program (internal documentation required) along with a completed report of the project. A Report Template is included in this document and has sections address Design, Implementation and Testing for you to complete. Ensure that you leave enough time to complete the report as a summary of the project.

You will receive a better evaluation for a fully documented and clean project that has only some of the proposed functionality than one that has all the functionality but has incomplete documentation and report.

Evaluation

- CPT is worth 30% of you course grade
- There are aspects of every category included and they are weighed as per the course outline (App 30%, Know 25%, Think 25% Comm 20%)
- Once a project has been proposed, it will not be changed except at the request of the teacher.
- Final Rubric is a checklist of skills across categories. Each is pass/fail, you either demonstrated the skill throughout the project or you did not.
    - Some of those will be trivial (proper header included in all files)
    - Some are more subjective (Efficient use of Code)
- Checklist of skills translate to a level in each category
- You may choose to ignore certain skills on the checklist. This is fine but it will limit the highest level of achievement that you can get on the project.

Academic Honesty

It is expected that the project represents your own work.

You are allowed to look for inspiration and solutions online for components of your project, but they must be clearly documented in the code and cited. When the code is evaluated against the checklist, the cited code will be ignored (aka only your original work will be assessed).

Oral questioning may take place if concerns arise as to the authorship of the work handed in. Work that is not believed to have been produced by the student will be given a mark of zero.

Use of objects and components not learned in this class are allowed but will almost certainly lead to inspection of the project unless brought to the attention of the teacher earlier.

In short, make sure you understand every aspect of the project you are handing in.

ICS3U – Python Final Project Rubric

Design (10 Think)	***Documentation (10 Comm)***
_ Written explanation of design _ Use of High Level Design _ Use of Low Level Design _ Flowcharts for all complex logic _ Designed use of Variables _ Designed use of Selection _ Designed use of Repetition _ Designed use of Subprograms _ Designed use of Class and Objects _ Unique characteristics in design	_ Author's Name(s) _ Proper header in all files _ Headers for each subprogram/class _ Comments all logical blocks _ Proper spacing of blocks of logic _ Proper indentation throughout _ Completed project report _ HLD diagrams are clear _ LLD diagrams are clear _ Report is well written
Implementation (10 App)	***Testing & Deployment (10 Know)***
_ Implements all aspects of the design _ Free from syntax errors (It runs) _ Good use of Variables _ Good use of Sequence _ Good use of Selection _ Good use of Repetition _ Good use of Lists _ Good use of Subprograms _ Good use of Class and Objects _ Efficient code	_ Program without error (any!!) _ Problem accomplishes designed task _ All file(s) are appropriately named _ Files and documents are organized _ Software is portable _ Design documents match the code _ Screenshots included of key features _ Testing section - how was tested _ Analysis section – what was learned _ Project exceeds expectations

Assessment

The project will receive a mark based on the number of checks from the above lists in each category. The level will be calculated as follows:

Category	Level 0	Level 1	Level 2	Level 3	Level 4-	Level 4	Level 4+
Think	< 5	5 checks	6 checks	7 checks	8 checks	9 checks	10 Checks
Comm	< 5	5 checks	6 checks	7 checks	8 checks	9 checks	10 Checks
App	< 5	5 checks	6 checks	7 checks	8 checks	9 checks	10 Checks
Know	< 5	5 checks	6 checks	7 checks	8 checks	9 checks	10 Checks

ICS3C – Python Final Project Rubric

Design (8 Think) _ Written explanation of design _ Use of Design Tools (pick one) _ Flowchart for piece of complex logic _ Designed use of Variables _ Designed use of Selection _ Designed use of Repetition _ Designed use of Subprograms _ Unique characteristics in design	***Documentation (8 Comm)*** _ Author's Name(s) _ Proper header in all files _ Comments most logical blocks _ Proper spacing of blocks of logic _ Proper indentation throughout _ Completed project report _ Design diagrams are clear _ Report is well written
Implementation (8 App) _ Implements all aspects of the design _ Free from syntax errors (It runs) _ Good use of Variables _ Good use of Sequence _ Good use of Selection _ Good use of Repetition _ Good use of Lists _ Good use of Subprograms	***Testing & Deployment (8 Know)*** _ Program without error (any!!) _ Problem accomplishes designed task _ All file(s) are appropriately named _ Files and documents are organized _ Screenshots included of key features _ Testing section - how was tested _ Analysis section – what was learned _ Project exceeds expectations

Assessment

The project will receive a mark based on the number of checks from the above lists in each category. The level will be calculated as follows:

Category	**Level 0**	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Level 4+**
Think	< 4	4 checks	5 checks	6 checks	7 checks	8 checks
Comm	< 4	4 checks	5 checks	6 checks	7 checks	8 checks
App	< 4	4 checks	5 checks	6 checks	7 checks	8 checks
Know	< 4	4 checks	5 checks	6 checks	7 checks	8 checks

Project Report Template

Title page

Project documentation will start with a title page. It should include some graphical representation of the projects (pictures? themes?) and must include the following information:

- Project Title
- Author(s)
- Course
- Date Delivered

Design Section

Your design should outline the solution you envisioned. It can include descriptions, mock-ups, design diagrams (IPO/Stepwise/UMLs/Flowcharts for programming) as appropriate based on the complexity of the project. All features of the project should be addressed in one manner or another in the design. It may also include aspects of the design which did not get completed in the implementation.

Implementation Section

In this section you record the final scope of the project and everything that you accomplished in it. You will describe everything the program does and how it does it (highlight the use of different techniques and how everything fits together). It is not just showing your product works but also documenting how you have used what you learned in the course to complete the project. Remember a picture says 1000 words so include screen shots and thumbnails to enhance your documentation.

You do not need to include any code in the report as you will be handing you code separately and fully documented.

Testing Section

Outline how you tested your product – what test cases you considered, how components were tested both independently and combined. Provide examples of how and where testing helped improve your product (be specific).

Analysis Section

You will then complete a short summary (1-2 page) as a final analysis of the project. In it you should review the whole project and describe what went well in the project and what areas need more time/effort. Include what aspect of the project that you are most proud of and list at least two things that you would improve if you had the opportunity.

References

You must clearly cite any references to material which does not represent your own work on this project. For example, if sections of code were written by someone else (perhaps the teacher gave you some of the solution) you must clearly indicate what sections represent their work. There is nothing wrong with leveraging other people’s work, so long as they are given appropriate credit and you are not attempting to claim their work as your own. This applies to code segments, algorithms, diagrams, graphics/images and any written documentation.