CS 632p – Project 1 (Using Python)

Your final project 1 (documentation and code) is **due at the start of class on 10/23/19.**

Note: We will allocate time in class for students to try each other's game. It is therefore important to make sure your game is easy to use and able to deal with malicious users and spurious input! During this class you may have a brief interview with me -- you will be expected to explain your work and answer questions on it.

Interim deadline: 10/2/19   I would like to see the first version of your game flowchart in this class on this date so that I can check the logic makes sense and is of sufficient complexity to obtain a passing grade if completed.   I want to also check that your design ideas can be implemented using the constructs introduced in class. Bring your design to class and be prepared to talk about the design with me.

Important
This is an individual project and you must not share designs, code or any documentation with other students. Please remind yourself of the student responsibilities regarding academic integrity.

INTERACTIVE TEXT BASED COMPUTER GAME

Using the example given in class as inspiration, your task is to invent and program your own interactive text based computer game. It is very important that you start thinking about your project early on and do things in stages, as the month you have to do your project will pass by before you know it! There is no excuse for late or missing work!

You should have sufficient knowledge to start designing and writing pseudocode for your project. Firstly, draw up a flowchart for your game. It is important that you examine and think about any decisions that may need to be taken based on user input and any iteration that may need to occur. Try writing the logic for the various component aspects of your game. Start simple and have some fund during this stage. How are you going to error check?

Once you have designed your game and are happy it will work on paper, your task will be to think about how you can translate this into a python program so that a user can interactively play the game. Some of the constructs you will need to implement your project will have already been covered in class. Other constructs will be presented during the next few classes. The more you can use the constructs from class, particularly those constructs that improve the style of your program (e.g. methods) the better. Again, start simple and code small bits of your program and test it. Add bits slowly and surely as you write them, as shown in class. Leave any fancy dialogue you want to use to interact with the users until you have the logic sorted. Don't try to do everything in one go.

While it will be possible to complete the coursework using the Python covered in classes, you are free to use other features of the language if you want. You may need to reference Python documentation and books if you decide to take this route.

Important: this is an open-ended project and the game can be as simple or as intricate as you want – it is up to you to take your project as far as you can go with it! The game must contain (at a minimum) assignment statements, variables, conditionals, loops and methods. You will be awarded marks on how appropriately and neatly you use these constructs, and try to use the available python modules

effectively, and on how well you comment and document your code, how exhaustively you test all the paths in your program and there will be a mark for originality and complexity in your idea. While you can use it for inspiration, DO NOT simply copy the example used in class. Each game should be unique. You will get extra marks if it's fun or cool. Bottom Line – I prefer a neat, clearly organized, well thought out and thoroughly tested small program over a huge one that is a mess and not properly tested!

NO SPAGHETTI CODE – SERIOUSLY !!
NO UNTESTED CODE – VERY SERIOUSLY !!
& ABSOLUTELY NO DUPLICATION   BE SMART ABOUT THIS!

Remember that the more complicated your game becomes, generally the more testing you must do to validate

If you are a little further along with your coding, here are some ideas to extend your game:
- Allow multiple players
- Have the computer take on the role of one of the other players (the computer must make intelligent choices)
- High score table which shows the top 10 players names and scores
- Store the current game state and/or high score table to disc to re/load the next time (requires input and output to file coupled with exception handling)
- Introduce a time factor into your game (e.g. playing against the clock)
- Have random elements
- Look at including additional python collections, algorithms and data structures

Assessment
Your project requires you to design, code, test and document quite a challenging piece of software! This project is worth 10% of CS632p, so use your time accordingly. The marks will be allocated as follows:

- Design of game [10 marks]
    - Flowchart to show the overall game structure and logic – descriptions of you design prior to coding (e.g. your algorithm, story board, fishbone diagram, MECE tree, etc)
- Creativity [10 marks]
    - Is your game interesting, novel and fun? (perhaps add some random elements)
- Code [50 marks]
    - Overall structure (e.g. neat – no spaghetti code – no duplication – use of methods)
    - Exception / error handling (e.g. handles tricky users and does not crash)
    - Useful comments (where needed)
    - Use of sensible variable / method names
    - Correct implementation of control structures (conditionals and loops)
    - Attention to user interaction (spelling!)
- Testing [10 marks]
    - Demonstrating you code actually implements your design
    - Demonstrating your program works with different user inputs
    - Try this out with your friends – make sure they understand it and it actually works (perhaps get them to fill out a questionnaire)
    - Don't just include screen dumps of sessions – let me know what you tested, why, how, with what results
- Supporting Documentation [20 marks]

- o write up to accompany your software design, instructions on how to play, implementation and testing (bound with code listing, test output, and title page)
- o marks will be gained for professional presentation – professional software is not just code, it is code coupled with the documentation needed to understand, use and maintain it – marks will be lost instantly if not present (the work should be understandable to your grandmother and your 5 year old neighbor)
- o summarize what you think is good about your project, what you think is weak about your project, and what you would do to improve it if you were to write a second version. What enhancements would you make, what would you take out, what devices and consoles are a good fit for your game? Which ones are not?

Deliverables Guide
You are expected to hand in a complete package which is a hard copy, bound and page numbered containing (as a minimum):

- Cover page (with your name(s), date, project name, course etc)
- Table of Contents and page numbers
- Documentation which should include as a minimum:
- Instructions for running and using your program
- Screen shots of your program running
- Flow charts and explanation of your design
- List of all the core features of your game
- Evidence and explanation of testing (you are expected to demonstrate path testing (i.e. that every path through your code has been executed and demonstrated to work)
- List of known bugs
- Details of further enhancements you would like to make
- Source Code (which should be fully commented and show the source code file name)
- Zip file (sent via email) which is labeled with your name, course details and date, containing all your source code, documentation, details on how to run your code and a version of your program that is ready for me to run (with any necessary detailed instructions if files etc are involved)
- Self assessment (a marking sheet will be provided on blackboard) print a hard copy and supply it with your deliverables

If you omit to include any of the above, you will lose marks. Remember that I am your customer and completeness and presentation of your deliverables is very important! Code has to be debugged and maintained. Clean design documentation and instructions are essential for this.

Only paper copies (and .zip file) will be accepted as detailed above. You are advised to make 2 copies of your overall deliverable (one for me and one for yourself) because I will keep a copy of your work.

You will be demonstrating your working application in class on 10/23/19 so please be ready to show it and your documentation.

Deadline – start of class on 10/23/19
Your projects will be assessed by demonstration and then by review of your supporting documentation. Demonstrations will take place in class so make sure your programs work in the labs. As preparation, you must arrange for your files to be available and usable. No excuses!

Why am I doing this?
The purpose of this project is to give you practice in writing a reasonably complex python program on your own. A program that combines concepts, statements, and techniques covered separately in the classes and exercises. As you learn more about object orientation, we will look how your games could be redesigned. You could also look at how you could create a graphical user interface for your game in the future too.

Getting Help
I am here to help along the way. To make sure you get the best possible marks for your project, you can show me or the Seidenberg tutors the things that you do each week as you progress. Don't get stuck – you never lose marks for asking questions and seeking help. You can talk to your classmates and get help but do not copy each other's work. This is easy to spot you know! So always list whoever helped you and with what in your write up!

Complexity
In class we will review a game with sufficient complexity to receive a perfect score (assuming proper documentation). You should try to challenge yourself with this assignment, but realize that you have a firm deadline for this deliverable.

Most importantly. **HAVE FUN WITH IT!!**

Joe Everett
Department of Computer Science, Pace University
Fall 2019