**ICS4U**

**Final Coding Project / Culminating Activity**

This project is worth 10% of your final course mark.

**Task:** Develop a large program that uses some of the functions and structures you’ve learned this year. You will work in pairs or groups of three.

**Components:**

1. Create a plan.
2. Create a UML diagram listing the classes and methods you’ll need
3. Code the program. Comment throughout. Test that individual components work in isolation
4. Deliver finished program
5. Peer-assess other projects
6. Review your project

**Plan**

Address all of these questions: What do you want to make? What features will it have? How will it use concepts taught in ICS4U?

Create a time-line for the work to be done. Base it by class period. A good time-line will mention who will do what work.

**UML Diagram**

Draw the UML diagram showing the classes you will use, and the dependencies (arrows) between them.

For each of the classes, list the “interface” i.e. the *public* methods (with parameters and return types) and constructors.

The Plan and UML Diagram will be due at the end of class on May 1st.

**Check-ins**

As you code, we will check in on two occasions, just to ensure that you're making progress and using your time effectively. Show us up-to-date code as well as the most recent *working* versions of parts of the program.

**Peer Evaluations**

On May 28th, all students must be present to peer-assess other projects. You will provide constructive feedback that will help others fine-tune their programs.

**Review**

Part a) Compare the plan to the actual performance of the finished program. What was a success and what would you recommend for improvement? What would you change based on the peer evaluations?

Part b) Based on the rubric for the program, select a mark for each item, and provide written justification with evidence.

Make sure to hand in the peer evaluation pages with your review.

**Timeline:**

April 11: Project given

April 27: Work on plan

May 1: Work on plan. Plan and UML diagram due at the end of the period.

May 3-22: Work periods. The AP class will have two periods reviewing for the AP exam.

May 24: Work period. It’s strongly recommended that you are close to finished, and are able to do “alpha testing” this day.

May 28: Peer evaluation

May 30: Project is due (this is not a work period)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Class schedule** | **Deliverables** | **Group plan** | ✓ |
| 4/11/2018 | Planning day |  |  |  |
| 4/27/2018 | Planning day / begin programming |  |  |  |
| 5/1/2018 |  | Plan / UML |  |  |
| 5/3/2018 | AP Exam Review |  |  |  |
| 5/7/2018 |  |  |  |  |
| 5/9/2018 |  |  |  |  |
| 5/11/2018 | AP Exam Review |  |  |  |
| 5/15/2018 |  | Check-in day #1 |  |  |
| 5/17/2018 |  |  |  |  |
| 5/22/2018 |  |  |  |  |
| 5/24/2018 |  | Check-in day #2 |  |  |
| 5/28/2018 | Peer evaluation day (**Mandatory attendance**) | Peer evals |  |  |
| 5/30/2018 | Exam review day 1 | Review and final program |  |  |

**Project ideas:**

Games: Clue, Monopoly, Dominion, Euchre, Chess

Non-Games: Bike route finder, Word lists, Hospital records

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Description** | **Estimated Mark** | **Final Mark** | **Reasoning** |
| **Knowledge** | Use of ifs, loops, arrays |  | 5/5K | Ifs, loops, and arrays are used extensively throughout the program and are in the correct syntax and are used correctly |
| Review (Use template below) |  | 5/5K | Review completely covers the requirements |
| **Application** | Use of ICS4U topics (at least 3 for full marks)   * Hierarchies * File i/o * 2d arrays * Recursion * GUIs * Sort/search |  | 8/8A | Hierarchies- they are used in the classes Space, OtherSpace. And property to create the board, an array of spaces so there could be tax spaces, card pickup, and properties  File I/O- file IO is used to create the board of a configuration file. This is useful because if one would want to make a different config file for a different board , they could do that. It is also easy to change the board because it is in a separate, easy to read file  GUIs- the GUI is used as the interface. It is used because it is much easier for the user to interact with than the console |
| Follows object-oriented approach |  | 4/4A | There are many different objects in the project to manage and store variables in the intended way. |
| Check-in days (2 marks for May 15) |  | 3/3A | We completed the check in days properly and on time |
| **Thinking** | Features of the program   * A variety of features that are effectively implemented * Appearance |  | 7/8T | The feature of the program are most of what monopoly has. Some interaction are possible in the real board game and are not here, such as trading money or more than 4 players.  The appearance looks nice and the interface is fairly user friendly although some functions may not be obvious |
| Bug-free   * Performs as expected, does not crash unexpectedly |  | 3/4T | The game and the code itself is robust but due to the compiler and programs used, one may encounter some bugs |
| Plan (Due May 1st)   * Answers the three questions in an organized manner |  | 3/3T | Plan was completed to the criteria and handed in on time |
| **Communication** | UML (Due May 1st) |  | 3/3C | The UML was accurate to the idea of the monopoly project structure we wanted. |
| User interface (usability) |  | 2/3C | The interface fully shows what the player and the project can do however the buttons and interface may not be very clear on which button does what functionality |
| Documentation (Comments, Follows naming conventions) |  | /4C | All proper naming conventions were held, and the code was properly commented |

**Review (due May 30th)**

Compare the plan to the actual performance of the finished program. Submit this as a Word document or PDF in the same folder as your source code. Hand in the peer evaluation slips with a printed copy of this page of the document.

1. What was a success and what would you recommend for improvement?

The main functions of the game are a big success, the movement, turns, rent, buying property. Improvements can be made to the interface to make it easier to use such as animations. Improvements can also be made to the code itself by reducing the size of individual classes and making the code itself easier to read and follow.

1. What would you change based on the peer evaluations?

Make the game easier to use and use a .jpg of the US version instead of the UK version sot he names on t GUI and the image match.

1. Based on the rubric for the program, select a mark for each item, and provide written justification with evidence. Indicate which ICS4U topics you used, in what class, and why.