



**ISCG5420 Programming Fundamentals
Assignment 1: Haunted House**

Total marks: 100
Course Weighting: 25%

Department of Computing

Due Date: 21:00, Sunday, 8 June 2014

Assignment Brief:

You are tasked with improving the code for the Haunted House game. Please read the associated **hand-out** and the **code** provided on Moodle. Although the implementation of your code is your own work, the ideas for which to implement and improve the code are free for public discussion. You are free to use code from the Internet or other sources but please cite the sources in the relevant program code. You will be given debate time and implementation time in class where we can collaborate on ideas and improve our understanding of coding in order to complete the tasks.

This task should take approximately 42 hours of implementation to complete. you will be given approximately 21 hours of class time to collaborate and undertake the individual tasks, use all of your time wisely.

Difficulty Level: Very Low

1. Improve the **readability** and **structural design** of the code by improving the function names, variables, and loops, as well as whitespace. Move functions close to related functions or blocks of code related to your organised code (software architecture).
(5 Marks)
2. Remove any **runtime**, **semantic**, and **syntax errors**. Make comments in the code and list the number of errors you have corrected and how you corrected each specific error.
(5 Marks)
3. Improve the **game** by including multiple commands that invoke the same functionality. For example the commands "LIST," "INVENTORY," "ITEMS" and "CARRYING" all list the items being carried by the player. You should be able to do this for all of the commands. However, you may limit your improvement to five of the commands. Improve the **game** by allowing players to enter mixed case input.
(5 Marks)

Difficulty Level: Low

1. You will add functionality to the game in the form of a **Dungeon Master** who will explain the task to complete the game (explain the win condition in a conversational style). You will define a win condition for the game, for example, collecting all items and returning to the entry point (location 0).
(10 Marks)
Note: The current win condition is to have a score of 17 and be in location 57. This is really obscure. It is possible with a condition like this when it is not explicit that players will never win. This is often called a dead man walking scenario.
2. You will re-factor (update) the display of available exits and other messages to be less cryptic, i.e., instead of "NE" you will display a message similar to "there are exits to the North and East."
(10 Marks)
3. You will add functionality to quit or exit the game using an appropriate command.
(10 Marks)

Difficulty Level: Medium

4. You will update the map functionality, such that only locations that have been visited will be displayed. (10 Marks)
5. You will have functionality to “Save” the game state so that the game can be continued at a later stage. You will implement the functionality for “Loading” the saved game. Implement separate commands to save and load the game. (20 Marks)

Difficulty Level: High

6. You will implement **one** of the following:
 - a. Improve the **game** by adding more features, for example you can examine more items, more props etc. You may implement this using more lists regarding items and props, remember, you should check if the object is being carried or in the location of the player.
 - b. Introduce **random monster** encounters and battles, for instance in the Haunted House you could introduce vampires, ghosts, werewolves, etc. You may have to look at role-playing combat systems that already exist, try to use existing combat systems.
 - c. Introduce a **non-player character** that uses AI to communicate with players (see the Eliza project online). You may be able to find existing code on the Internet for this, if you do please cite the source in the code files.

(25 Marks)

(Total 100)

BONUS POINTS

This question is designed for you to improve your overall score for ISCG5420— Programming Fundamentals. If you have missed points on the practical tests you may be able to increase your aggregate score for this course! The additional question will require approximately 8 hours additional implementation time.

Difficulty Level: your choice.

7. Add other functionality of **your choice** to the game. Justify its inclusion in comments next to the new functionality; get guidance and advice from your lecturers. Some ideas for this appear in the associated Haunted House hand-out. You may choose another option from question 6.

(Marks 20)

Submission Details:

This is an individual assignment. You must work on the individual tasks by yourself and all work on these tasks must be your own. Your final submission should contain a power-point slideshow following the format provided in class, the template is available on Moodle. When submitting the work via the Moodle as part of your assessment submission you agree that the work is your work and your work alone.

Assignment hand-in:

Assignments submitted after the due date and time without having received an extension through Special Assessment Circumstances (SAC) will be penalised according to the following:

- 10% of marks deducted if submitted within 24hrs of the deadline
- 20% of marks deducted if submitted after 24hrs and up to 48hrs of the deadline
- 30% of marks deducted if submitted after 48hrs and up to 72hrs of the deadline
- No grade will be awarded for an assignment that is submitted later than 72hrs after the deadline.
- Students submitting assignments after the due date and time will be ineligible to resubmit a failed assignment.

Special Assessment Circumstances:

A student, who due to circumstances beyond his or her control, misses a test, final examination or an assignment deadline or considers his or her performance in a test, final examination or an assignment to have been adversely affected, should complete the Special Assessment Circumstances (SAC) form available from Student Central.

Within any semester, a student may have only one SAC per course.

When requesting an SAC for an assignment, the SAC form must be submitted (along with work completed to date) within the time frame of the extension requested; i.e. if the Doctor's certificate is for one (1) day, then the SAC form and work completed must be submitted within one (1) day.

Unacceptable Assistance

- Working together on one copy of the assessment and submitting it as own work.
- Giving another student your work.
- Copying someone else's work. This includes work done by someone not on the course.
- Copying from books, Internet etc. and submitting it as own work. Anything taken directly from another source must be acknowledged correctly: show the source alongside the quotation.
- Changing or correcting another student's work.

Have a query? Want to improve your work?

You could:

- Talk it over with your lecturer, course coordinator, and programme leader.
- Visit Te Puno Ako or Maia for learning advice and support.
- Visit the Centre for Pacific Development and Support.
- Contact USU Advocate for independent advice.
- For contact details and more information, go to www.usu.co.nz (<http://www.usu.co.nz>)

This is an individual assignment. You must work on the individual tasks by yourself and all work on these tasks must be your own.

Please sign the statement below to declare that this assignment submission is your own work and hand in the signed statement with your assignment. Failure to sign and include this statement may mean your assignment is not marked.

ISCG5420—Programming Fundamentals

Assignment 1

I declare that the individual part of this assignment submission is my own work. Where I have incorporated work by other people, I have correctly acknowledged the source in my assignment.

Student Name Student ID

Date: