Department of Computer Science

Coursework Assessment Specification

Module Number: 08101 Title: Programming 1

Lecturer: Rob Miles

Coursework Assessment Number: 2 of **2**

Title: Snake Game

Method of Working: INDIVIDUAL

Workload Guidance: Typically, you should expect to spend between **3** and **10** hours on this assessment in addition to time spent in scheduled laboratories.

Date of publication: Friday 16th of November, 2006

Date and time for submission: 9.15 am on **Thursday 30th of November, 2006**

This assignment should be submitted electronically via the Class Server program on the due date unless an extension has been authorised on a *Request for an Extension for an Assessment* form which is available from the Office or http://www.student-admin.hull.ac.uk/Mitcircs.doc. The extension form, once authorised by the lecturer concerned, should be attached to the assignment on submission (or given to the lecturer in the case of electronic submission). The assignment will be demonstrated by the student during the laboratory sessions on **Thursday 30th of November** and **Friday 1st of December.**

Notes:

Late penalties
For work submitted late the penalty is loss of **20% marks per day**. Work that is **5 or more days late** will automatically be graded as **FAIL**, and no re-submission will be allowed. For full details, see the Department's student handbook.

Use of Unfair Means
You are reminded of the University's Code of Practice on the Use of Unfair Means (http://www.student-admin.hull.ac.uk/unfair.html) and that the work you submit for assessment should contain no section copied in whole or in part from any other source unless where explicitly acknowledged by means of proper citation.

ASSESSMENT STRATEGY AND ASSESSED COURSEWORK

This assignment is worth **30** % of the module marks.

The overall assessment strategy is designed to evaluate the student's achievement of the module learning outcomes, and is subdivided as follows:

ASSESSMENT STRATEGY AND LEARNING OUTCOMES

This assignment contributes towards the following module learning outcomes.

LO	Learning Outcome	Method of Assessment
1	*Problem Identification*	*Software Submission*
2	*Program Analysis*	*Software Submission*
3	*C# Programming*	*Software Submission*
4	*Algorithmic Design*	*Code Review*
5	*Program Specification and Testing*	*Code Review*
6	*Specification and Design*	*Documentation Submission*

ASSESSMENT CRITERIA

Assessment Criteria	Contributes to Learning Outcome	Mark
Program works correctly	1,2,3,4,5,6	**40%**
Evidence of Good Design	1,2,4,5	**10%**
Appropriate Program Enhancements	1,2,3,4	**20%**
Appropriate Documentation	2,5,6	**10%**
Evidence of Appropriate Testing	1,5	**10%**
Appropriate Layout and Identifier Selection	3,5	**10%**

The Snake Game

The player uses the keys to control the direction of movement of the head of the snake. The head can move up, down, left or right. The body follows along after the head. The aim of the game is to guide the head of the snake towards food items whilst avoiding the walls and the snake body itself. The player is awarded points for the food that they eat and the length of time that they last in the game.

If the snake head touches a wall or another part of the snake body the game is over.

As the game continues the snake body gets longer, making it harder to avoid crashing.

Food items appear on the playfield at random positions and for a limited time. If the player manages to guide the snake head over the food a number of points are awarded for this.

Program Requirements

You are required to create a version of the Snake game which runs in the console window. This provides a playfield made up of 25 rows of 80 characters. In order to achieve a pass mark on this assessment your program must provide the following functions:

- The player must be able to steer the head of the snake around the screen using the keyboard. The snake will move continuously in the direction chosen by the player. If another direction is selected the snake movement must change to reflect this.
- If the snake head moves off the screen (i.e. the coordinates of the draw position are beyond the playfield area) the game must display an appropriate message and stop.
- At random intervals a food item should appear on the screen. This will remain on the screen for a random time. If the player manages to guide the snake head over the food item the score for the player must be increased. The current score should be displayed on the screen during the game. When the game finishes the final scores should be displayed for the player before the program stops.

Program Enhancements

In addition to the minimal requirements, you can also enhance the program to add further features:

- The head of the snake must be followed by a body which increases in length as the game progresses. If the snake head collides with any part of the snake body the game is over.
- Appropriate sound and colour effects may be added to enhance the gameplay. You should however not expend too much effort on this aspect of the game, as the work is being assessed for programming, not artistic, ability.
- The game may implement a highest score table, which is displayed at the end of the game.

Required Submission

You are required to submit your program and a user guide by means of Class Server. The precise requirements are as follows:

Program Source

You will be required to submit the source code of your program. It is important that this contains an appropriate level of comments to document the program behaviour.

User Documentation

You are also required a single page document for the player of your game. This document should explain how the game is played and give any other details that you feel are appropriate.

Program Demonstration

As part of the assessment of this work you will be required to demonstrate your solution in the Fenner Computer Suite in your practical session on either Thursday 30th November or Friday 1st of December. The timetable for demonstrations will distributed via email and also published on the 08101 Sharepoint site.

Ensure that you turn up in good time for your demonstration. At the appointed time you should have your program ready to run and a listing of your code displayed on the monitor screen. The demonstration will take no more than 10 minutes and will involve a test play of the game and an examination of your source code.

Rob Miles