**

The Department of Computer Science

**CIS4202**

**Interface Programming**

Level 7

Coursework 2

**Module Leader: Mark Anderson**

🕿 01695 65 7634

**Email: mark.anderson@edgehill.ac.uk**

***Administrators:***

🕿 01695 65 7603

# 

# Coursework – Project

Weighting: 60**%**

Hand in date: 17 December 2016 **@ 3pm**

Learning Outcomes Assessed:

LO1 Critically evaluate the key issues within the subject of human-computer interaction and the methods and techniques employed in interface design.

LO2 Critically evaluate a user centred system development process.

LO4 Develop a small application with appropriate GUI for a given case study that will enable the identification areas of weakness in existing interface design strategies, and to systematically propose alternatives or improvements to overcome those limitations.

**Introduction**

This assignment is a major part of the formal assessment for CIS4202 and is, therefore, compulsory. You will be working to complete the design and implementation of a hardware/software system, so time management is of the essence. Read this document carefully and make sure that you are clear about what you have to do, and what you have to hand in, before you attempt the assignment.

The aim of this assessment is to give you the opportunity to experiment and further develop your skills developing an effective user interface using the Kinect camera and/or Augmented Reality. You will construct software to drive the camera and make use of different means of gathering user input to implement your solutions.

**Background**

For this task, you will be required to implement a system to help visitors and new students orientate themselves to the Tech Hub and the immediate surroundings. A user should be able to interact with your software to find out more about the department, the staff and the facilities. Users might add information as well (for example, having a virtual pin board for notices, or send messages to staff through the software…although for the purposes of this exercise please only implement this for me!).

Examples of what you might implement include

1. An interactive wall using a Kinect camera (a display to be projected onto any wall or flat surface with interactions implemented through gestures using the Kinect)
2. An AR app using Vuforia which offers directions (like a sat nav) to a staff member’s office
3. Pop up information about classes that are scheduled in a lab (could be in either 1 or 2 above)
4. AR app that shows a photo and details of availability of staff when outside their office
5. Something I haven’t thought of…

Not all of the functionality needs to be fully implemented, and it can be mocked up in places as the interface design is the important element here. However, you will need to make sure that your software allows a user to interact in a number of different ways. For example, using a Kinect you will have multiple gestures and possibly voice comment. Using Vuforia you should not just rely on image targets. Use virtual buttons or other interaction.

In the first phase, you should critique user interface design guidelines for Windows (8 or 10) and OSX applications. Following this, you should produce designs and an implementation that utilises these guidelines.

In a second phase you should critique the Android and iOS UI guidelines and produce a design that adheres to these. You may implement these if you wish (as a separate interface), but this is not important.

The third phase should include a critique of the Microsoft guidelines for Kinect or consider the use of the guidelines above for the development of AR apps.

The final element will be to compare and contrast the guidelines. What, if anything, requires changing between the interface designs and why is this the case. Are there elements that contradict each other? Could you improve upon the Kinect guidelines, as Natural User Interfaces become more readily available? Or perhaps you implemented an AR app – how might the guidelines be amended to support best practice and facilitate the adoption of AR?

**Initial Planning**

Before commencing this coursework, you should familiarise yourself with the development environment to be used (Visual Studio is preferred – you may develop on other platforms, but support may be limited). Sheets are available in Blackboard to provide guidance in using this IDE for interface development.

When writing code, take care to layout your code clearly and ensure that you include useful comments in the code to explain how your code is working. As a guideline, a comment should be placed

1. At the top of each file to say when it was last updated and by whom.
2. At the start of every method/function to indicate its role.
3. Where selection or iteration is used to explain its purpose.

**What you should submit:**

You should submit a series of photographs or a video which represent the design that you have proposed for your applications. The design should be accompanied by a brief commentary which details for each gesture the role of the construct and how the application is intended to work.

For the project you should submit evidence that you have undertaken different aspects of the project design. To do this, you could either submit a “how to” guide for building the sections of the software that you have been responsible for (e.g. using photographs to show the build at each gesture completed complemented with a short description), or you could submit a short video show you building the project with a brief narrative. You should link the design of your language to the HCI guidelines that you developed for CW1 and demonstrate how (and why) these have been implemented. You should submit the source code for your areas of the project as a zipped Visual Studio project.

Through Blackboard you should also keep a **week by week diary** (blog) detailing what you have done, what worked, what didn’t and how you are going to approach the work the following week. These diaries will form the reflective component of the coursework. You should also submit agendas and minutes of any team meetings that take place.

The project will be marked as an individual piece of work. Your submission should be individual and your contribution must be clear.

## What will be assessed

1. Project submission for the “step-by-step” guide
   * Gesture and interface design with a supporting commentary for all aspects of the project (the four phases outlined above).
   * “How to” build guide
   * Source code for the project
   * Example program(s) written using your language
   * Comments in the code which clearly define the role of the code.
   * A bibliography of additional resources used.
2. Individual critique/reflection discussing how the task progressed, what were the issues that arose, how you overcame any issues and what would you do differently.
   * Individual weekly diary in the form of a blog on Blackboard

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Learning Outcome**  **Assessed** | **Mark  (0 – 29)** | **Mark  (30 – 39)** | **Mark  (40 – 49)** | **Mark  (50 – 59)** | **Mark  (60 – 69)** | **Mark  (70 -84)** | **Mark  (85 – 100)** |
| **Grade:**  **Fail** | **Grade:**  **Narrow fail** | **Grade:**  **Pass** | **Grade:**  **Good** | **Grade:  Very Good** | **Grade:  Excellent** | **Grade:  Outstanding** |
| **LO1:**  Critically evaluate the key issues within the subject of human-computer interaction and the methods and techniques employed in interface design  **Weighting: 25%** | Introduction/conclusion are not clearly present or are missing.  Minimal or missing analysis which does not relate to the intended goals of the report.  No attempt has been made to relate theory to code implementation, or the attempt utilises irrelevant concepts.    Key tasks or steps are missing.  Poor presentation. A number of GAPS and formatting errors throughout. No relevant content contained in report. Report is not structured. | Narrative with no details of review and analysis. Either may be missing (however, one should be evident).  Does not relate to body of analysis  An attempt at analysis which does not clearly relate to the problem being considered or the goals of the final report.  Incorrect application of theory to problem issues with major errors (e.g. code design/implementation ignored key software requirements).  Poor presentation. A number of GAPS and formatting errors throughout. Relevant content contained in report but unstructured and not using correct terminology. Some content is not relevant to the problem being considered. | Narrative with few details of review and analysis. Either may be missing (however, one must be evident).  Does not demonstrate evidence of further reading.  Does not clearly relate to body of analysis  Lacking analysis, only descriptive observations OR missing some points altogether.  Incorrect application of theory to problem issues.  Key tasks or steps are missing.  A number of GAPS and formatting errors throughout. Relevant content contained in report but unstructured and not using correct terminology. | As **Pass** but introduction/conclusion is mostly descriptive narrative with some details of review and analysis.  May demonstrate some evidence of further reading  Attempt to relate to body of analysis.  Analysis demonstrates limited depth of understanding. Mostly descriptive and may contain misconceptions/errors. Includes description of key points.  May have some missing elements of theory or minor misconceptions in analysis  GAPS errors and/or spacing errors. Report structured using appropriate language/terminology. May contain errors or structuring issues. Discusses most key points. | As **Good** but introduction/conclusion is mostly critical narrative with details of review and analysis.  Demonstrates evidence of further reading, possibly with reference list.  Intro/conclusion are clearly related to the analysis.  Analysis demonstrates some depth of understanding. Proportionally descriptive / analysis and may contain minor errors. Includes some critical analysis of key points.  May have minor misconceptions in analysis, but will include all key concepts successfully.  Few GAPS errors and/or spacing errors. Report structured using appropriate language/terminology. Discusses all key points. | As **Very Good** but introduction is relevant, structured, puts review in perspective. Clearly linked to analysis. Conclusion is present, structured and linked to analysis.  Clear evidence of further reading which is referenced and cited using Harvard.  Analysis demonstrates depth and understanding of issues. Critical evaluation of key points.  Professional presentation. No or minor GAPS errors. Clear and concise structure. Consistent use of correct technical terminology. | As **Excellent** but introduction and conclusion are clearly backed by relevant theory which demonstrates wider reading around the subject area.  Analysis develops a unique insight or offers original interpretations of findings. All findings are supported by relevant theory.  Report demonstrates evidence of wider reading around the subject. For example, cites relevant technical terminology from alternative sources that are relevant to the field. |
| **LO2:**  Critically evaluate a user centred system development process  **Weighting: 25%** | Software design/implementation follows none of the concepts from lectures and worksheets. A subsection of an application may be represented in the software design, but the design contains serious errors (e.g. selection of wholly inappropriate components).  Few of the exercises or initial activities have been completed with minimal evidence to support this. | Software design/implementation follows some concepts from lectures and worksheets but contains significant issues. A subsection of an application may be represented in the software design, but the design contains a number of errors (e.g. incorrect selection of components or incomplete designs).  Possibly one of the exercises has been attempted, but evidence of completing the step-by-step guides is submitted. | Software design/implementation generally follows key concepts from lectures and worksheets but may contain major issues. A number of software components are represented in the software design, but components may be missing or contain errors (e.g. incorrect selection of appropriate technologies/libraries).  Most of the exercises and additional exercises have not been attempted. | As **Pass** but software design/implementation mostly follow key design concepts. This will be evidenced through the correct representation of software components in designs which may contain minor issues (e.g. a small number of incorrect libraries used). Designs are provided for most of the software components.  A systematic approach to each build has been adopted (e.g. most classes/modules contain comments to determine function).  The exercises associated with the portfolio guides have been completed, but the additional exercises have not. | As **Good** but software design/implementation follow key design concepts. This will be evidenced through the correct representation of software components in designs which may contain minor issues (e.g. a small number of incorrect libraries used). Designs are provided for all of the software components.  The exercises associated with the portfolio guides have been completed and a number of the additional exercises have also been attempted. | As **Very Good** but software design/implementation closely follow key design concepts. This will be evidenced through the correct representation of software components in designs making appropriate use of libraries/technologies. Designs are provided for all of the software components.  A systematic approach to each build has been adopted (e.g. fully documented code using comments in each module/class).  All the exercises and additional exercises have been attempted and mostly completed (e.g. one or two additional exercises attempted but not successfully completed). | As **Excellent** but software design/implementation represent unique solution to problem. This should be evidenced through designs provided for all of the software components along with evidence of supplementary reading.  All the exercises and additional exercises have been completed. There may be evidence of additional experiments undertaken. |
| **LO4:**  **Develop a small application with appropriate GUI for a given case study that is robust and capable of being understood, reused and modified**  **Weighting: 30%** | Software design/implementation follows none of the concepts from lectures and worksheets. A subsection of an application may be represented in the software design, but the design contains serious errors (e.g. selection of wholly inappropriate components).  No systematic approach to design/build.  Code print outs submitted for few software components. Unsuccessful attempt at implementing most of the functionality in the designed language.    None, or a minimal, attempt has been made to make use of the available library functions to support interaction by the user with the Kinect camera.  Code contains no comments.  None of the programming in relation to the project has been attempted. | Software design/implementation follows some concepts from lectures and worksheets but contains significant issues. A subsection of an application may be represented in the software design, but the design contains a number of errors (e.g. incorrect selection of gestures).  Possibly only one of the project areas has been attempted and submitted.  Code print outs submitted for a number of project areas. Limited implementation of functionality. Includes broken base functionality from code derived from portfolio exercises.  A limited attempt has been made to make use of the available library functions to support interaction by the user with the Kinect camera (e.g. uses simple built in gestures only). Contains major errors which result in code that  does not compile or execute correctly.  Code contains a comment at the start of each program module to define its functionality.  Only makes attempt to code in a small area of the project. | Software design/implementation generally follows key concepts from lectures and worksheets but may contain major issues. A number of software components are represented in the software design, but components may be missing or contain errors (e.g. incorrect selection of appropriate technologies for input capture).  Approach to build is not wholly systematic demonstrated through the design documents for the visual/gesture-based language.  Code print outs submitted for overall project. Limited implementation of functionality. May include broken base functionality derived from portfolio.  An attempt has been made to make use of the available library functions to support interaction by the user with the Kinect camera. May contain major errors which result in incorrect gestures being read or relayed through the methods.  Code should compile, but may not deploy or execute correctly.  Code contains a small number of comments to define its functionality.  In a number of areas the programming for the project has not been attempted. | As **Pass** but software design/implementation mostly follow key design concepts. This will be evidenced through the correct representation of software components in designs which may contain minor issues (e.g. a small number of incorrect libraries used). Designs are provided for most of the software components.  There is evidence that the approach to the build is systematic demonstrated through the submission of the designed gesture-based programming language.  Base functionality specified in problem specification successfully coded. Should have an attempt at programming in most areas of the design. This will be evidenced through the submission of code print outs.  An attempt has been made to make use of the available library functions to support interaction by the user with the Kinect camera. May contain minor errors in function calls.  Code should compile and run without errors.  Code contains a number of comments to define its functionality (e.g. most methods and constructs are commented). | As **Good** but software design/implementation follow key design concepts. This will be evidenced through the correct representation of software components in designs which may contain minor issues (e.g. a small number of incorrect libraries used). Designs are provided for all of the software components.  A mostly systematic approach to each build has been adopted demonstrated through the submission of the designed gesture-based programming language.  Base functionality specified in problem specification successfully coded. An attempt to program all areas of the design has been undertaken, including multiple skeletal tracking, although some areas may not be complete. This will be evidenced through the submission of code print outs.  Code is mostly well commented to define its functionality (e.g. all methods and constructs are commented). | As **Very Good** but software design/implementation closely follow key design concepts. This will be evidenced through the correct representation of software components in designs making appropriate use of libraries/technologies. Designs are provided for all of the software components.  A systematic approach to each build has been adopted demonstrated through use of methodology and/or use of programming patterns.  Base functionality specified in problem specification successfully coded. An attempt at all areas of programming has been made, including multiple skeletal tracking, which is mostly successful although may contain minor errors. This will be evidenced through the submission of code print outs.  Library functions have been successfully implemented to support interaction by the user with the Kinect camera.  Code is well commented to define its functionality in detail (e.g. all methods and constructs are commented to include definition of gestures). | As **Excellent** but software design/implementation represent unique solution to problem. This should be evidenced through designs provided for all of the software components along with evidence of supplementary reading.  There may be evidence of additional experiments or design work undertaken in relation to critical comparison of technologies.  Extended functionality from that specified in problem specification successfully coded. All of the coding for the project has been successfully completed. May provide evidence of further extensions. This will be evidenced through the submission of code print outs.  Library functions have been successfully implemented to support interaction by the user with the Kinect camera . Evidence of using additional functionality which is supported through references to external sources being used. |
| **Critique/Diary**  **Weighting: 20%** | Minimal evidence of engagement in the course displayed through a no entries in the weekly diary to be submitted through Blackboard, or one entry submitted at end of coursework.  Diary comments made are not reflective.  No bibliography submitted | Significant lack of evidence of engagement in the course displayed through a small number of entries in the weekly diary to be submitted through Blackboard (e.g. one or two entries entered at the end of the coursework).  Diary comments made are not reflective.  No bibliography submitted | Lack of evidence of engagement in the course. Evidenced through a small number of entries in the weekly diary to be submitted through Blackboard, or entries being retrospectively entered.  Limited evidence of problem-solving through reflective comments in weekly diary.  Limited evidence of additional reading through the submission of a limited bibliography to support documentation. | As **Pass** but evidence of engagement in the course through a mostly complete set of entries in the weekly diary to be submitted through Blackboard. Some entries may be missing or retrospectively entered  Some evidence of problem-solving reflective comments in weekly diary. For example, some problems encountered may be discussed but there may be no evidence of an attempt to find solutions.  Some evidence of additional reading through the submission of a bibliography to support documentation which contains mostly relevant sources. | As **Good** but evidence of engagement in the course through a complete set of entries in the weekly diary to be submitted through Blackboard.  Evidence of problem-solving through reflective comments in weekly diary. For example, most problems encountered will be discussed and most are supported with potential solutions. | As **Very Good** but evidence of thorough problem-solving evidenced through reflective comments in weekly diary. For example, all problems encountered will be discussed and supported with working solutions.  Evidence of additional reading through the submission of a thorough bibliography to support documentation which contains relevant and recent sources to the module. | As **Excellent** but evidence of thorough problem-solving evidenced through reflective comments in weekly diary. For example, all problems encountered will be discussed and supported with working solutions which provide a unique or alternative approach.  Evidence of additional reading through the submission of a thorough bibliography to support documentation which contains relevant and recent sources to the module and related topics. |