# Assignment Briefing Sheet (2024/25 Academic Year)

## Section A: Assignment title, important dates and weighting

|  |  |  |  |
| --- | --- | --- | --- |
| Assignment title: | Software Development Project: Task 1 | Group or individual: | Group |

|  |  |  |  |
| --- | --- | --- | --- |
| Module title: | Software Development 01 | Module code: | 5FTC1306 (NHC)  5FTC1307 (WHC) |

|  |  |  |  |
| --- | --- | --- | --- |
| Module leader: | Rob Chambers | Moderator’s initials: | ST in lieu of MB |

|  |  |  |  |
| --- | --- | --- | --- |
| **Submission deadline:** | **03/11/2024** | Target date for return of marked assignment: |  |

|  |  |  |
| --- | --- | --- |
| You are expected to spend about | 36 | hours to complete this assignment to a satisfactory standard. |

|  |  |  |
| --- | --- | --- |
| This assignment is worth | **50%** | of the overall assessment for this module. |

## Section B: Student(s) to complete

|  |  |
| --- | --- |
| **Student ID number** | **Year Code** |
|  |  |
|  |  |

|  |
| --- |
| **Notes for students**   * For undergraduate modules, a score above 40% represent a pass performance at honours level. * For postgraduate modules, a score of 50% or above represents a pass mark. * Late submission of any item of coursework for each day or part thereof (or for hard copy submission only, working day or part thereof) for up to five days after the published deadline, coursework relating to modules at Levels 0, 4, 5, 6 submitted late (including deferred coursework, but with the exception of referred coursework), will have the numeric grade reduced by 10 grade points until or unless the numeric grade reaches or is 40. Where the numeric grade awarded for the assessment is less than 40, no lateness penalty will be applied. * Late submission of referred coursework will automatically be awarded a grade of zero (0). * Coursework (including deferred coursework) submitted later than five days (five working days in the case of hard copy submission) after the published deadline will be awarded a grade of zero (0). * Regulations governing assessment offences including Plagiarism and Collusion are available from <https://www.herts.ac.uk/about-us/governance/university-policies-and-regulations-uprs/uprs> (please refer to UPR AS14) * Guidance on avoiding plagiarism can be found here: <https://herts.instructure.com/courses/61421/pages/referencing-avoiding-plagiarism?module_item_id=779436> * Modules may have several components of assessment and may require a pass in all elements. For further details, please consult the relevant Module Handbook (available on Studynet/Canvas, under Module Information) or ask the Module Leader. |

# Assignment Briefing Sheet (2024/25 Academic Year)

|  |
| --- |
| **This Assignment assesses the following module Learning Outcomes (from Definitive Module Document):**   * describe and differentiate current and emerging technologies in software development. * differentiate the tools, models and methods used to design software. * produce solutions for complex technical issues in software development. * select and use suitable tools and techniques productively in the development of software. |
| **Assignment Brief:**  *see attached for more information.* |
| **Submission Requirements:**  *See attached for more information.* |
| **Marks awarded for:**  *see attached for a complete percentage breakdown.* |
| **Type of Feedback to be given for this assignment:**  *Feedback will be provided via Canvas and be based on the assignment activities illustrated in the attached assignment.*  *see attached for more information.* |

**Mini Project 1 (50%) w/c - 02/10/24**

**Hand in Date: 03/11/24**

**Introduction**

This module is designed to focus on practical work with an emphasis on software development as well as the tools and techniques needed to create solutions for complex technical issues.

Although each coursework is designed to be a challenge its completion will give you the opportunity to gain experience and help improve your expertise with tools and techniques used in software development. The outcomes and personal experiences you gain will help provide valuable insights and skills that can be transferred to other software tasks (including later software development modules and work-based learning projects).

This module adopts a repeated and consistent approach to development while setting a variety of different software challenges and technical issues. This method will give you the opportunity to show how you reflect on and apply previous learning and experience to help improve your current success and will offer you a practical platform to demonstrate your knowledge, skill and attributes.

Throughout this task you are expected to select and use suitable tools and techniques productively, demonstrate good software development standards and practice as well as careful project management. All projects include a set of standard guidance activities with some suggested times (activities 1 through 5 are group based while 6 is post project and must be completed individually). Please note: this project contains multiple activities with a tight time constraint meaning that you have a limited window with which to complete the task – it is therefore of critical importance that you effectively **plan**, **organise** and **scale** the project appropriately to ensure you work efficiently and can complete the work.

During in-class activities you are free to ask questions, seek tutor assistance or research and collaborate on how best to address, resolve and explore issues. As a module you are expected to complete two separate in-class software development mini-projects (coursework 1-2). As an individual you will be expected to demonstrate improved knowledge, experience and insight with regards to how each project is completed.

**Coursework Tasks**

**Part A (Group Work): Project using ‘Prototyping’ (20%)**

Organise your group to explore and discuss which of the following three mini projects you will attempt. Each possible project includes a core statement to help illustrate the nature of the task. You are expected to evolve this statement into a viable software specification that details the requirements for your project – remember: it is very easy to overly complicate and over specify software designs; ensure you scale this task appropriately.

* Project: Pairs

Rows of hidden items (often playing cards) will be selected by the player then matched with other hidden items. Players gain points by finding matched pairs. If no pair is found both items are hidden again

* Project: Colour-It (e.g. paint-by-numbers)

The user is presented with a designed image and is given the ability to select various colours with which they can use to paint the image.

* Project: Wordle/Numle (e.g. guess the word or number)

Players are presented with a string of characters and attempt to find the hidden word or number sequence. Players will be informed when they find a correct character and whether it is in the correct position.

Review the suggested guidance (shown below as Project Activities: 1, 2, 3, 4, 5 and 6) to help plan, prepare, design, complete the design documentation (Part B) this practical challenge. Depending on the task’s overall development strategy it may mean that you can ignore some activity elements while having to repeat others. However, part of the challenge is in how you design, manage and complete this task.

**Part B (Individual Work): PowerPoint Design Document (30%)**

Combine and compile ALL group based project evidence into a well-structured, proofread PowerPoint style presentation (review Project Activities 6 for guidance on what should be included). To avoid overloading slides ensure you use ‘notes’ to store extended detail or supporting information. Ensure you include **ALL** necessary evidence to help you achieve a good grade. Any images embedded within your presentation should be cropped and optimised to help reduce file size and make submission easier and faster. Draft documents compiled and completed during group activities should be individually reworked, improved and supported with extended annotation and analysis.

**Project Requirements:**

Platform: Web / Online

Languages: JavaScript, CSS, jQuery, HTML5

Collaboration: GitHub, SourceForge, Bitbucket, etc.

**Project Activities**

**Activity 1 Group Guidance – Requirements/Creative Session**

1. Review all available projects before deciding which to develop
2. Consider your primary target audience (see Possible Project Target Demographics)
3. Create an overall specification based on user and system requirements (including HCI, game/application-rules and the game/application-mechanics (e.g. what are the rules for the game, how will the game be controlled, how will any non-player characters interact, etc.)
4. Create two different user profiles (based on your selected target type) – these profiles should include basic details regarding individual’s wants and needs associated with the game
5. Determine the project’s high-level ‘functional specifications’ (e.g. what are the hardware requirements, operating system environment, application functions, collisions, AI, scores, timers, etc.)
6. Determine the project’s high-level ‘non-functional’ specifications (aesthetic, usability, ease of use, feedback style, basic needs, etc.)
7. Create mock-ups for the overall look, user interface design, dialogue windows and input mechanisms, etc
8. Construct basic storyboards associated with the game-play/app use
9. Identify and rank potential risks to the project’s success (such as: technical knowledge, coding, testing, scope, dependencies) – this should not include time management
10. Identify and establish your software development strategy
11. Define an overall test plan (this will be used later and may be repeated several times to ensure the project is successful).

**Activity 2 Group Guidance – Design Analysis Session**

1. Refine and improve your overall specification and ideas by removing inappropriate or out-of-scope elements, simplify requirements, identify opportunities to improve and streamline HCI elements, game-play and game mechanics, etc.)
2. Agree your project’s requirements ensuring you can deliver them successfully.
3. Use basic pseudo code to help define, establish and quickly test high-level in-game functions, actions and logic *(depending on your preference you may prefer to complete step 4 before step 3)*
4. Use basic UML flowcharts to help plan, design and test game logic, interaction, mechanics and flow
5. Establish game state management (start, win, lose, draw) – confirm how the state could be monitored, detected or changed?

**Activity Group Guidance 3 – Coding (***this activity is likely to be very time consuming***)**

1. Create a Git repository (or similar) to organise your coding strategy
2. Allocate responsibilities
3. Code (regardless of strategy you must continually test).
4. Complete this stage of the application (depending on strategy this may be repeated or be an incremental activity)

**Activity Group Guidance 4 – Testing**

1. Conduct end-to-end system testing (as defined in activity 1.11)
2. Identify and log any errors (including code, logic, interaction, state, etc.)
3. Fix and Repeat (depending on strategy this may mean returning to Activity 2 and or 3)
4. Repeat while necessary

**Activity Group Guidance 5 – End of Project Review (Part A: 20%)**

1. Complete project source code listing (25%)

Your listing should demonstrate:

* + Consistent evidence of good standards (such as: naming, comments, etc.) (25%)
  + Consistent evidence of good structure (such as: indenting, continuity, etc.) (25%)
  + Consistent evidence of good coding practices (such as: modular, defined functions and clear logic) (25%)

**Activity Individual Guidance 6 – Post Project Design Documentation (Part B: 30%)**

1. Gather and document evidence from ALL of your project’s activities and outcomes.
2. Combine and compile ALL project evidence into a well-structured, proofread PowerPoint style presentation\* that:
   1. Identifies and describes the project (5%)
   2. Identifies each member of your group and summarises their contribution to the project. (5%)
   3. Well-annotated screen shots demonstrating and highlighting features and functions of your software (including: segments of source code illustrating how problems were resolved (e.g. NPC, AI and player control) supported by any initial high-level pseudo code, run-time screen captures, errors as well as embedded digital photos of any hand drawn images, diagrams, initial flowcharts or the results of whiteboard brainstorming sessions, etc.) (15%)
   4. A summary of the ‘target user’ including an example profile (5%)
   5. An organised list of user requirements (based on the target user) (5%)
   6. An organised list of system requirements (e.g. the project, the hardware, the software strategy, the game-rules and mechanics, etc.) (10%)
   7. A review of system testing (including test logs) (10%)
   8. A cited\*\* review of the development strategy used to include advantages and disadvantages (10%)
   9. An evaluation of how well your project met each of the requirements together with a statement of the project’s overall success and/or failure (15%)
   10. A critical reflection (with evidence) on **your** contributions and the work **you** undertook individually in order to help complete this project (including your strengths and weaknesses as well as discussing how you will further develop and/or improve these areas) (15%)
   11. References (*\*\*Harvard*) (5%)

*\*Presentations: Extended information should be placed in the ‘notes’ area to help avoid individual slide being overloaded with content. Embedded images must be optimised to reduce overall document size*

*\*\*Harvard: All references and citations should use Harvard formatting*

x

**What to Submit**

**Part A (20%) – ‘Project Review Statement’ & Source Code – Canvas Submission**

1. As a group you have to select and complete a chosen project (this includes activities 1 through 5).). A peer assessment form will be available to provide you with an opportunity to document details on the general contributions made by each member.
2. Share your Git repository containing all source code listing – ensure you clearly identify all code designed and developed by the group as well as referencing any third-party code (if used). Regardless of origin - all code must demonstrate good standards of practice, be internally documented and well structured (see activity 5).

**Part B (30%) – PowerPoint Design Document - Canvas Submission**

This is an individual post project activity that follows the group work. As an individual you will be expected to have taken copies of ANY and ALL project documentation created in the group session and will be expected to fully expand, refine and formalise them to an appropriate academic standard.

1. Create a presentation (review Project Activities 6 for guidance on what should be included together with weighted percentages) that documents and fully reviews the project, strategies and outcomes as well as offering a critical reflection on your contributions and possible areas of improvement.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Numeric Grade** |  |  | | | |
|  | **Demonstration of Knowledge:**  The report shows integration of software development theory into work and/or breadth/depth in applying theory and/or knowledge of software development concepts**.** | **Application of Technology / Demonstration of Practical Skills:**  Use of programming languages in the development of the solution, applying software development phases and programming constructs with clear explanations in the design documents. | **Ideas/Concept Development:**  Demonstration of; development of ideas and/or design, and/or concept development, problem solving. Produce a prototype or full product from the designs. The prototype of the software application integrates and demonstrates the most important functions stated in the designs. | **Analysis, Critical Evaluation and/or Reflection (with derivation of solution):**  Demonstration of evaluation, testing and reflection of the project development, the individual performance and the lessons learned from the project. |
| **90-100** | **Outstanding** | Outstanding breadth and depth demonstrated. Outstanding integration of literature and/or theory into work.  Outstanding exploration and demonstration of topic showing in depth knowledge and understanding | Outstanding use of appropriate technologies as applied to the problem domain. Occasionally stepping beyond expectations using sophisticated solutions. Consistently accurate and outstanding application of skills and techniques demonstrated | Solutions are novel within context. Profound depth of engagement and incisive ideas development. Planning is meticulous; decision making is perceptive. Methodologies are used in an outstanding manner. | Outstanding level of analysis, critical evaluation and/or reflection with outstanding application to derived solutions (where required).  Highly developed / focused work. Original and well-informed personal response |
| **80-90** | **Excellent** | Excellent breadth and depth demonstrated. Excellent integration of literature and/or theory into work.  Excellent exploration and demonstration of topic showing in depth knowledge and understanding | Excellent use of appropriate technologies as applied to the problem domain. Occasionally stepping beyond expectations using sophisticated solutions. Consistently accurate and Excellent application of skills and techniques demonstrated | Solutions reframe task within context. Profound depth of engagement and incisive ideas development. Planning is meticulous; decision making is perceptive. Methodologies are used in an Excellent manner | Excellent level of analysis, critical evaluation and/or reflection with Excellent application to derived solutions (where required).  Highly developed / focused work. Original and well-informed personal response |
| **70-79** | **Very good** | Very good breadth & depth demonstrated. Very good integration of literature and/or theory into work.  Very good level of knowledge and understanding demonstrated. Covers all relevant points and issues. | Very good use of appropriate technologies as applied to the problem domain. Very good and highly accurate application of skills and techniques demonstrated. Minor errors in technique and/or application with little or no impact on deliverables. | Solutions are innovative within its context. Considerable depth of engagement and successful ideas development clearly documented. Detailed planning and clear rationale for decisions. Methodologies are used in a very good manner | Very good level of analysis, critical evaluation and/or reflection of issues with Very good application to derived solutions (where required). Well-developed personal response |
| **60-69** | **Good** | Good breadth & depth demonstrated appropriate to topic.  Literature and/or theory integrated very well.  Good level of knowledge and understanding demonstrated. | Good use of appropriate technologies as applied to the problem domain. High level and very accurate application of skills and techniques understanding demonstrated. Small errors in technique and/or application with little impact on deliverables | Solutions relate directly to task and may step beyond conventions  Strong engagement with subject material and processes, evaluation of alternatives, solutions come from process. Strong use of methodologies to derive solutions | good level of, analysis, critical evaluation and/or reflection but not consistently taken to full extent with good application to derived solutions (where required). Partial personal response tends towards descriptive |
| **50-59** | **Clear Pass** | Good use of legends. Depth appropriate to topic BUT moderate breadth or vice versa. Literature and/or theory integrated into work.  Good grasp of the topic and some of its implications. Knowledge and understanding is demonstrated. Minor errors / omissions. | Evidence of use of appropriate technologies as applied to the problem domain. Good and reasonably accurate application of skills and techniques demonstrated. Some errors in technique and/or application with minor impact on deliverables | Solutions are appropriate to task, work well within conventions. Evidence of use of subject material and planning processes, range of alternatives put forward and evaluated  Experimentation to support implementation within conventions Good use of methodologies to derive solutions | Evidence of analysis and/or reflection but critical evaluation could be expanded on further. Good application to derived solutions (where required). Primarily descriptive personal response, sometimes restricted to immediate concerns |
| **40-49** | **Marginal Pass** | Satisfactory use of legends demonstrated but limited in breadth OR depth. Uncritical and quoted without comment where necessary. Satisfactory content / level of knowledge of the topic. Addresses part of the question. Some errors / omissions. | Satisfactory use of appropriate technologies as applied to the problem domain. Satisfactory application of skills and techniques demonstrated but with minor inaccuracies. Errors in technique and/or application with some impact on deliverables | Solutions limited to task and address conventions. Solutions found or adopted  Some planning but completion is rushed. Some experiments but limited alternatives. Methodologies are applied to derive solutions but steps are missed. | Satisfactory level of analysis and/or reflection but limited evidence of critical evaluation. Satisfactory application to derived solutions (where required). Descriptive personal response mainly restricted to immediate concerns |
| **30-39** | **Marginal Fail** | Limited in breadth and depth demonstrated. Some parts of legends used/quoted without comment. Limited content / knowledge. Limited or muddled understanding of the topic/question. | Limited use of appropriate technologies as applied to the problem domain. Limited application of skills and techniques demonstrated. Many errors in technique and/or application with high impact on deliverables. | Solutions reframe task inappropriately and do not address conventions. Basic use of strategies, few alternatives, limited evaluation with limited experimentation. Limited use of methodologies to derive solutions. | Limited evidence of analysis, critical evaluation and/or reflection. Limited application to derived solutions (where required).Too descriptive in parts. Limited personal response. |
| **20-29** | **Clear Fail** | Lacking in breadth and depth. Some parts of legends irrelevant to topic area. Lacking knowledge Content irrelevant / inaccurate.  Does not address the question and therefore does not meet the learning outcomes. | Very little use of appropriate technologies as applied to the problem domain. Very little skill and application of techniques demonstrated. High number of errors with very high impact on deliverables. | Lacking in appropriate solutions with very limited use of strategies, no evaluation and little evidence of ideas development. Little use of methodologies | Lacking in its level of analysis / critical evaluation and/or reflection. Minimal application to derived solutions (where required)  Mainly descriptive, lacking in personal response. |
| **0-19** | **Little or Nothing of Merit** | No / unsatisfactory evidence of Legends used and irrelevant to topic area  No / unsatisfactory level of knowledge demonstrated.  Content not appropriate to the topic. | No use of appropriate technologies as applied to the problem domain. No skill and application of technique demonstrated. Very high number of errors in deliverable or no deliverable submitted. | No or completely inappropriate solution. No use of strategies, no planning and no experimentation. No application of methodology | Unsatisfactory level of analysis / critical evaluation and or reflection. No application to derived solutions (where required)  Wholly descriptive. No personal response. |