|  |  |  |
| --- | --- | --- |
|  | Faculty of Computing, Engineering and Science |  |

**Assessment Cover Sheet and Feedback Form** 2024-25

|  |  |  |  |
| --- | --- | --- | --- |
| Module Code:  IS3S664 | Module Title:  Advanced Internet and Mobile Computing | | Module Team:  Daniel Cunliffe |
| Assessment Title and Tasks:  Noughts and Crosses game Android App | | | Assessment No.  2 |
| Date Set:  **23-Sep-24** | | Submission Date:  **10-Jan-25** | Return Date:  **10-Feb-25** |

**IT IS YOUR RESPONSIBILITY TO KEEP RECORDS OF ALL WORK SUBMITTED**

|  |
| --- |
| **Marking and Assessment** |
| This assignment will be marked out of 100%  This assignment contributes to 50% of the total module marks.  This assignment is unbonded. |
| **Learning Outcomes to be assessed** (as specified in the validated module descriptor [https://icis.southwales.ac.uk/](https://icis.southwales.ac.uk/studentmodules/11463/studentmodulespecifications) ):  To demonstrate the design and practical implementation of a mobile application. |
| *Provisional mark only: subject to change and / or confirmation by the Assessment Board* |

**Coursework Details**

During the course you have developed a basic noughts and crosses game Android App using Android Studio and Kotlin. The aim in this coursework is to develop this game further.

There are three deliverables for this coursework, the game app, a report and a video demo.

**The Game**

The game should be an enhancement of an existing noughts and crosses game, rather than an implementation from scratch. This existing code base can come from one of two sources:

1. It can be the basic noughts and crosses game developed during the course. In this case, marks will be awarded for a functional base game and for the enhancements.
2. It can be an external noughts and crosses game written by someone else and made freely available on the internet. In this case marks will not be awarded for a functional base game or the existing functionality provided, as it is presumed that working code was available. Marks will only be awarded for enhancements. In-code comments should be used to clearly identify the sections of code which you have written yourself and the indicate where amendments have been made.

**Enhancements to the game must be developed mainly in Kotlin**, though some elements of Java or C++ may be included if necessary.

**The target platform for the app is a Pixel 3 API 29.**

Further functionality that is added to the basic game will result in a higher grade, depending on complexity, integration, appropriateness to mobile platform and so on. You want to show technical ability, but you also want to develop a playable game.

The choice of enhancements is up to you. Some examples are shown below.

Basic enhancement examples:

Adding appropriate comments to the code

Minor cosmetic enhancements

Adding a reset button

Intermediate enhancements examples:

Extending the game to a 4x4 grid

Adding best of 3 option

Including some form of persistent data, e.g. storing game state

Add additional challenges, for example having to successfully answer a multichoice question before you can take your turn, otherwise you forfeit your turn

Adding a timer

Allowing the user to customise the game

Including multiple screens with navigation

Advanced enhancement examples:

Add an AI opponent to play against

Develop an additional game based on a different mechanic, such as Connect Four

The above are just suggestions – it is up to you to develop the game as you choose.

**Report**

The main purpose of the report is to describe in detail the coding work that you have undertaken. This includes both a high level description of what the code does, as well as a low level description of how this is achieved in code, including new code and amendments made to the code base.

The report should include the following:

* Clearly identify the base code used and the functionality it provides. This should include a link to an external code base if one is used.
* Clearly identify the enhancements implemented. Describe the additional functionality and features that have been added to the code base. This may include interface design documentation.
* Describe in detail how the enhancements have been implemented, with particular reference to modifications made to base code. This should include selected code extracts where appropriate. This may include software design documentation.
* Develop and apply an appropriate test plan for the application (including the code base).
* Include references for any sources used.

**Video Demo**

Create a brief video walkthrough, no more than 3 minutes long, highlighting the features you have implemented. The focus should be on the gameplay and user interaction rather than technical details of the implementation.

**Deliverables:**

There are three deliverables for this coursework, the game app, a report and a video demo.

The complete game app project must be uploaded to Blackboard. Please ensure that all required media items are included.

The report must be uploaded to Blackboard as a Word document.

The video, or a link to watch/download the video, must be submitted to Blackboard.

**Grading Criteria**

|  |  |  |
| --- | --- | --- |
| **Marking Scheme** | **Marks Available** | **Marks Awarded** |
| Noughts and Crosses game Android app developed in Kotlin | **60** |  |
| Report | **30** |  |
| Video demo | **10** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Fail (0 – 29)** | **Narrow Fail (30 – 39)** | **3rd Class / Pass** | **Lower 2nd Class / Pass** | **Upper 2nd Class / Merit** | **1st Class / Distinction** |
| **Game / 60** |  |  |  |  |  |  |
|  | Base game only partially functional  Code limited to that provided in external base code  Code poorly presented  Look and feel could be improved  No game | Base game only partially functional  Code largely limited to that provided in external code base  Code limited to that developed in class  Code poorly presented  Look and feel could be improved | Code largely limited to that developed in class  Enhancement code is straightforward  Code presented reasonably well  Extensions to base game limited  Enhancements do little to improve the game  Enhancements are not integrated well with the base game  Look and feel could be improved | Enhancement code goes some way beyond that in the code base  Enhancement code shows some degree of complexity and sophistication  Code presented well  Extension is well-conceived  Enhancements provide some improvement to the game  Overall game is reasonably cohesive and enhancements are integrated to some extent  Basic look and feel | Enhancement code goes well beyond that in the code base  Enhancement code shows a good degree of complexity and sophistication  Code presented very well  Reasonably innovative and thoughtful extension  Enhancements clearly improve the game  Overall game is cohesive and enhancements are well integrated  Good look and feel | Enhancement code goes significantly beyond that in the code base  Enhancement code shows a high degree of complexity and sophistication  Code presented to an excellent standard  Innovative and thoughtful extension  Enhancements greatly improve the game  Overall game is highly cohesive and enhancements are well integrated  Excellent look and feel |
| **Report / 30** |  |  |  |  |  |  |
|  | Code base not identified  Code base functionality not described  Enhancements not described  Enhancement implementation not described  No test plan  No testing  No references  No report | Code base identified, but not clearly  Code base functionality poorly described  Enhancements poorly described  Enhancement implementation poorly described  Implementation commentary lacks sufficient detail  Test plan weak  Testing poor  No references | Code base clearly identified  Code base functionality description fair  Enhancement description fair  Enhancement implementation  description fair  Implementation commentary lacks detail in parts  Test plan fair  Testing fair  Limited use of references  References could be presented more effectively | Code base clearly identified  Code base functionality description reasonable  Enhancement description reasonable  Enhancement implementation description reasonable  Reasonable level of detail in implementation commentary  Test plan reasonable  Testing reasonable  Some use of references  References reasonably well presented | Code base clearly identified  Code base functionality description complete and effective  Enhancement description complete and effective  Enhancement implementation description complete and effective  Good level of detail in implementation commentary  Good test plan  Good testing, well reported  Good use of references  References well presented | Code base clearly identified  Code base functionality description complete and to a high standard  Enhancement description complete and to a high standard  Enhancement implementation description complete and to a high standard  Excellent level of detail in implementation commentary  Excellent comprehensive test plan  Excellent testing very well reported  Excellent use of relevant references  References very well presented |
| **Video Demo / 10** |  |  |  |  |  |  |
|  | No video | Ineffective video presentation  Lacking sufficient level of description | Basic video presentation  Acceptable level of description | Reasonably effective video presentation  Fair level of description | Effective video presentation  Good level of description | Highly effective video presentation  Excellent level of description |