# Solent University

# Coursework Assessment Brief

# Assessment Details (please delete all blue sections before publishing to students)

|  |  |
| --- | --- |
| Module Title: | Programming Fundamentals |
| Module Code: | CUP455 |
| Module Leader: | Mark Bennett |
| Level: | 4 |
| Assessment Title: | Project AE1 |
| Assessment Number: | AE1 |
| Assessment Type: | Software Project and Report |
| Restrictions on Time/Word Count: |  |
| Consequence of not meeting time/word count limit: | There is no penalty for submitting below the word/count limit, but you should be aware that there is a risk you may not maximise your potential mark.  Assignments should be presented appropriately in line with the restrictions stated above; if an assignment exceeds the time/word count this will be taken in account in the marks given using the assessment criteria shown.\* |
| Individual/Group: | Individual |
| Assessment Weighting: | 100% |
| Issue Date: | 26/09/2023 |
| Hand In Date: | 15/12/2023 |
| Planned Feedback Date: | January 2024 |
| Mode of Submission: | Online through SOL  Only FINAL submissions will be accepted. DRAFT submissions will not be considered an attempt and will not be marked. |
| Anonymous Marking | This assessment: Is exempt from anonymous marking. |

# Assessment Task

You are to create a turn-based strategy game that pits a player-controlled creature against a similar creature controlled by the computer. This will be a simple version of an arena combat game, inspired by popular trading card games where two opponents move in turns until one is defeated. You will submit the Game Project along with a Project Report documenting its development from analysis through to final testing.

The Game Project will be written in either C++ or C# using Visual Studio 2022. You may use either an empty project or a Win32 Console project to get started.

Detailed specifications for the task are available in a separate document: Assessment Task Specification.docx, which is on the module SOL page under the Assessment tab.

You are also required to submit a report which consists of your design (either flowcharts or pseudocode) and your test plan and test results. Your design should be clearly labelled, and test results should show analysis of the problem and suggested solutions where possible.

# Assessment criteria

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CRITERIA** | **A1 – A4** | **B1 – B3** | **C1 – C3** | **D1 – D3** | **F1 – F3** |
| **ANALYSIS & DESIGN:**  **Use appropriate problem-solving methods and tools to design simple programs.**  **Approach and record problem-solving and program development steps in an organised and structured way.**  **(30%)** | **An excellent and sustained use of problem analysis and solving techniques, all well documented.**  **Consideration of multiple approaches and justification for best one.**  **Well documented evidence of iterative design.**  **Design solves problem completely and elegantly.**  **Design may usefully extend the required functionality.** | **Thorough and consistent use of problem solving techniques, with a detailed analysis of the problem.**  **Evidence of appropriate redesign.**  **Design fully solves problem.** | **Clear use of appropriate problem solving techniques.**  **A full analysis of the problem, but lacking in detail.**  **Design may have minor errors and omissions.**  **Design fully solves problem, but in a basic fashion.** | **Some use of basic problem solving techniques.**  **Partial analysis of problem with some inaccuracies.**  **Design is appropriate, but may not fully solve problem.** | **Inadequate understanding of theory leading to poor or non-existent problem solving attempt.**  **Analysis is simplistic with inaccuracies and omissions.**  **Design not appropriate or does not solve problem.** |
|  |  |  |  |  |  |
| **CRITERIA** | **A1 – A4** | **B1 – B3** | **C1 – C3** | **D1 – D3** | **F1 – F3** |
| **IMPLEMENTATION & FUNCTIONALITY:**  **Recognise and use programming design syntax and language constructs.**  **Develop algorithms to solve common problems implemented in software programs.**  **(45%)** | **Excellent and wide ranging knowledge of programming.**  **Fully functional application may incorporate advanced techniques.**  **Application fulfils design in an elegant and/or efficient way.** | **Shows a clear understanding of programming syntax and constructs, used appropriately to create a fully functional application.**  **Application fully realises design requirements.** | **Basic level of competence in all programming syntax and constructs.**  **Application fully functional, but may not fulfil entire design.**  **Occasional minor errors in application.** | **Basic understanding of some syntax shown, but may not be complete or fully appropriate.**  **Occasional minor errors in application, some parts may fulfil design.** | **Significant deficiencies in understanding of program syntax.**  **Application missing or non-functional, or clearly does not fulfil design.** |
|  |  |  |  |  |  |
| **CRITERIA** | **A1 – A4** | **B1 – B3** | **C1 – C3** | **D1 – D3** | **F1 – F3** |
| **TESTING & QUALITY:**  **Produce and test software using an Integrated Development Environment.**  **(25%)** | **Thorough and detailed test plan.**  **Excellent and sustained use of multiple testing techniques with a full analysis of the application.**  **In-depth analysis of test results and well documented resolutions to problems encountered, with useful reference to code involved.** | **Thorough and consistent use of several testing techniques, following a detailed plan.**  **Full analysis of the application.**  **Comprehensive analysis of test results, with documented reference to problem resolution.** | **Clear use of some appropriate testing techniques**  **A full analysis of the application, though lacking in detail.**  **Tests cover all program paths, but may have minor errors and omissions.**  **Moderate analysis of test results with some reference to problem resolution.** | **Some use of basic testing techniques following a basic test plan.**  **Some evidence of the use of the debugger and/or program output to test the program.**  **Little analysis of results or diagnosis of problems found.** | **Little or no testing carried out, or inappropriate or unrelated testing reported.**  **No analysis of results or discussion of test results.** |

# Learning Outcomes

This assessment will enable you to demonstrate in full or in part your fulfilment of the following learning outcomes identified in the Module Descriptor:

**Living CV**

As part of the University's Work Ready, Future Ready strategy, you will be expected to build a professional, Living CV as you successfully engage and pass each module of your degree.

The Living CV outputs evidenced on completion of this assessment are:

1. I can understand a software specification document

2. I can use software design methods to develop a solution

3. I can implement my design using C++

4. I can develop a test plan for my solution

5. I can test my solution using my test plan and document the results

Please add these to your CV via the Living CV builder platform on Solent Futures Online [Solent Futures Online](https://eur03.safelinks.protection.outlook.com/?url=https%3A%2F%2Fsolentfutures.careercentre.me%2Fprogrammes%2F%3FprogrammeID%3DThzJ%252bRbk%252bQXoSlEaujPR0g%253d%253d&data=04%7C01%7Cian.harris%40solent.ac.uk%7Cf1bda34c4d564e82f6cb08da067fdf48%7Cd684e4cd491a4577bf33546478d72e3c%7C0%7C0%7C637829443517919744%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=ObCFbM3zY7CgU6SVNtitaq1udg0%2Bzlp1GuCAJ1y1utw%3D&reserved=0)

# Important Information

[Solent University Academic Regulations 2023-24](https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/2o-assessment-principles-and-regulations.pdf)

# Late Submissions

You are reminded that:

1. If this assessment is submitted late i.e. within 7 calendar days of the submission deadline, the mark will be capped at 40% if a pass mark is achieved;
2. If this assessment is submitted later than 7 calendar days after the submission deadline, the work will be regarded as a non-submission and will be awarded a zero;
3. If this assessment is being submitted as a referred piece of work, then it must be submitted by the deadline date; any Refer assessment submitted late will be regarded as a non-submission and will be awarded a zero.

[Assessment regulations](https://www.solent.ac.uk/about/documents/assessment-regulations.pdf)

# Extenuating Circumstances

The University’s Extenuating Circumstances (EC) procedure is in place if there are genuine short term exceptional circumstances that may prevent you submitting an assessment. You are able to self-certify for up to two assessment dates in any semester without supporting evidence for an extension of up to seven calendar days for coursework or to defer an exam to the resit period.

Alternatively, if you are not 'fit to study’ (or you have used up your two self-certification opportunities), you can request:

* an extension to the submission deadline of 7 calendar days, or
* a request to submit the assessment at the next opportunity, i.e. the resit period (as a Defer without capping of the grade).

In both instances you must submit an EC application with relevant evidence. If accepted under the university regulations there will be no academic penalty for late submission or non-submission dependent on what is requested. You are reminded that EC covers only short-term issues (20 working days) and that if you experience longer term matters that impact on your learning then you must contact the Student Hub for advice.

Please find a link to the EC policy below:

[Extenuating Circumstances](https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/2p-extenuating-circumstances.pdf)

# Academic Misconduct

Any submission must be your own work and, where facts or ideas have been used from other sources, these sources must be appropriately referenced. The University’s Academic Regulations includes the definitions of all practices that will be deemed to constitute academic misconduct. You should check this link before submitting your work.

Procedures relating to student academic misconduct are given below:

[Academic Misconduct](https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/4l-student-academic-misconduct-procedure.pdf)

**Ethics Policy**

The work being carried out must be in compliance with the university Ethics Policy. Where there is an ethical issue, as specified within the Ethics Policy, then you will need an ethics release or ethics approval prior to the start of the project.

The Ethics Policy is contained within Section 2S of the Academic Handbook:

[Ethics Policy](https://staff.solent.ac.uk/official-documents/quality-management/academic-handbook/2s-solent-university-ethics-policy.pdf)

**Grade marking**

The University uses an alpha numeric grade scale for the marking of assessments. Unless you have been specifically informed otherwise your marked assignment will be awarded a letter/number grade. More detailed information on grade marking and the grade scale can be found on the portal and in the Student Handbook.

[Grade Marking Scale](https://staff.solent.ac.uk/official-documents/quality-management/academic-handbook/2o-assessment-regulations-annex-1-grade-marking-scale.pdf)

**Guidance for online submission through Solent Online Learning (SOL)**

[Online Submission](http://learn.solent.ac.uk/onlinesubmission)

The Game Project and Project report must be submitted online before the deadline to receive full marks. Your submission must contain two files:

1. Your Project Report document, containing the design, testing, conclusions and any supporting diagrams, tables or charts.

* 1. The front page of this report should be a title page that contains at least the following information, "[Student Number]\_[LastName]\_[FirstName]\_CUP455\_AE1\_Report". (square brackets indicate placeholders, they should not be in the final file name)
  2. The document must be in either MS Word or .pdf format.
  3. Your source code may optionally be included as an appendix in the report.

2. A compressed/zipped file containing your Game Project. Zip up the *entire project folder*, not just the source file.

3. Only use .zip files, not .rar or other formats. If I can’t open your project I can’t mark it.

4. These two files should be submitted individually, not in one zip file, only the project file should be zipped.

5. Name the file "[Student Number]\_[LastName]\_[FirstName]\_DAC416\_AE1\_Project", (square brackets indicate placeholders, they should not be in the final file name).

6. Include all of solution and source code files: .h and .cpp files and any additional files you have created as resources for you project.

7. Make sure you include all source and resource files necessary to build and run your program, including any files supplied to you.

8. You do not need to include temporary or intermediate files created during the build process.

9. Due to the volume of network traffic especially near deadlines, the online submission system may be slower than you expect. Extensions for connection problems will not be granted unless university-wide. Give yourself plenty of time!

10. It is strongly recommended that you keep an exact copy of your project as backup in case of submission failure.

11. Ensure that the PC used to submit your work or create your submission package is free of viruses or malware. Submitting digital media containing any form of malware will result in significant penalty or failure.