**Southampton Solent University**

**Assessment Brief**

**Assessment Details**

|  |  |
| --- | --- |
| Unit Title: | Advanced Games Programming |
| Unit Code: | CGP600 |
| Unit Leader: | Nick Thomas |
| Level: | 6 |
| Assessment Title: | AE1 - Group Project |
| Assessment Number: | AE1 |
| Assessment Type: | Group Project Submission |
| Restrictions on Time/Length : | 2000 words |
| Individual/Group: | Individual reports with Group design evidence to be submitted once per group |
| Assessment Weighting: | 25% |
| Issue Date: | September 2017 |
| Hand In Date: | 20th October 2017 16:00 |
| Planned Feedback Date: | 4 weeks after submission |
| Mode of Submission: | Electronic online submission |
| Mode of Marking: | Electronic online submission |
| Mode of Feedback: | Face-to-Face |
| Number of copies to be submitted: | 1 |
| Anonymous Marking | This assessment willbe exempt from anonymous marking as it falls within an exempt category under the University’s Anonymous Marking Policy. |

# Assessment Task

See below

# Assessment criteria

See below

## **Learning Outcomes**

This assessment will enable students to demonstrate in full or in part the learning outcomes identified in the unit descriptors.

## **Late Submissions**

Students are reminded that:

If this assessment is submitted late i.e. within 5 working days of the submission deadline, the mark will be capped at 40% if a pass mark is achieved;

If this assessment is submitted later than 5 working days after the submission deadline, the work will be regarded as a non-submission and will be awarded a zero;

If this assessment is being submitted as a referred piece of work (second or third attempt) 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.

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2o-assessment-policy-annex-1-assessment-regulations.pdf?t=1411116004479>

## **Extenuating Circumstances**

The University’s Extenuating Circumstances procedure is in place if there are genuine circumstances that may prevent a student submitting an assessment. If students are not 'fit to study’, they can either request an extension to the submission deadline of 5 working days or they can request to submit the assessment at the next opportunity (Defer). In both instances students must submit an EC application with relevant evidence. If accepted by the EC Panel there will be no academic penalty for late submission or non-submission dependent on what is requested. Students are reminded that EC covers only short term issues (20 working days) and that if they experience longer term matters that impact on learning then they must contact a Student Achievement Officer for advice.

A summary of guidance notes for students is given below:

<http://portal.solent.ac.uk/support/official-documents/extenuating-circumstances/extenuating-circumstances.aspx>

## **Academic Misconduct**

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

Procedures relating to student academic misconduct are given below:

<http://portal.solent.ac.uk/support/official-documents/complaints-conduct/student-academic-misconduct.aspx>

## Ethics Policy

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

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

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2s-university-ethics-policy.pdf>

## **Anonymous Marking**

A copy of the University’s Policy on Anonymous Marking, process details and student guidance on submission sheet completion can be found on the following links, which are also uploaded on the Student Portal.

### Fact Sheet:

<http://portal.solent.ac.uk/documents/academic-services/policies-procedures-guidelines/anonymous-marking-fact-sheet.pdf>

### Process:

<http://portal.solent.ac.uk/documents/academic-services/policies-procedures-guidelines/anonymous-marking-process.pdf>

## **Grade marking**

The University uses a letter grade scale for the marking of assessments. Unless students have been specifically informed otherwise their marked assignment will be awarded a letter grade. More detailed information on grade marking and the grade scale can be found on myCourse.

### Policy:

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2o-assessment-policy.pdf>

### Fact sheet:

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-4/4o-grade-marking-briefing-for-students.pdf>

# Assignment Task

The assessment for this unit consists of the software design and implementation of a 3D game using a variety of good software practices and technical components, such as, work-breakdown structures, testing plans, high-performance graphics, game mechanics, optimisation, and collision detections.

For this coursework you have to work in group sizes of 2 (or 3 in the extreme case of odd numbers). You CANNOT choose to work alone, if you do not have a group then you will be assigned to one at the lecturer’s discretion.

# Feedback

Feedback will be given in the laboratory sessions so students are able to discuss their work and justify/show any choices, issues or custom features of innovation that are relevant. This will be done on a group basis (or as individuals where necessary).

# Game Specification

The group should decide on a **design theme** that the game should try to follow, but as the game design is **not being assessed** this does not have to be sophisticated or elaborate. Remember, it is the **software design** for your game that is being assessed in the first coursework.

The group should develop the game according to the agreed game design specification.

The following requirements **must be met**:

1. It should be a 3D game, written with Object Oriented C++, using Visual Studio and DirectX.
2. The player should be able to move around an environment (real-time & interactive).
3. The environment should be quite large, and must have static and moving obstacles (e.g., blocks, statues, and rolling rocks). These must be models (though they can be simple if you like). Some of these objects should be able to be pushed by the player, and others should be collectable.
4. The objects in the environment should have textures and some form of lighting.
5. The player should collide with objects; either stopping for static objects, or pushing them if they are moveable, or pick them up if they are collectable.
6. The environment must contain non-player entities represented by a model, or at least a series of connected 3D shapes, using textures and lighting.
7. The entities should move around the environment in some fashion (e.g., patrol an area, chase or run away from player, and fly).
8. The entities should collide with objects. This should cause the entities to perform an action (e.g., random, predefined direction change, and fly away).
9. There should be some form of interaction between the entities and the player based on collision detection (e.g. player/entity damage/death, change of behaviour, and start conversation).

Additional grades are achieved through enhancements to the basic game requirements (e.g., innovation and robustness).

Examples of such enhancements include:

* extending the game to use advanced features, such as, procedural content, physics, or AI;
* using *advanced* DirectX/Windows/Shader techniques;
* using more sophisticated techniques for *core* game features, such as, managing the scene and collision detection;
* optimising game performance;
* good object-orientated design, with the game and its constituent objects extensible and reusable.

These are just a small set of possibilities, look at what other games do and use your imagination to come up with others. Some of these enhancements will require additional research of 3D, object-oriented and game programming techniques not explicitly covered in the unit.

# What you need to do

1. Each Group Member needs to create a formal report in their words documenting the software design and development process for the game. In addition, the report must contain a reflective section describe what you did and why. Each report should be put together into one document - this must also be supported by the group design evidence. For example, this evidence should include Derived Tasks, Pseudocode, Flow Charts, Class Diagrams, Entity Relationship Diagrams, Dependencies, Testing Plans, Critical Elements, and Schedules. Be sure to clearly highlight and reference each group members’ contributions

# Part 1 - All members: Identify and Allocate Tasks

The group should analyse the game design to initially **break it down** into a **small number of tasks** that will contribute to the final full implementation of the design. These tasks describe the functionality of the basic brief and any additional functionality you have decided to implement (e.g. player movement, graphical techniques, collisions, HUD, game states, entity management, and AI). You should carefully consider how the separate tasks will work and/or **communicate with each other**. Don't forget you need to keep track of who did what and when and why for the final submission (e.g., keep track of how each member contributed and any problems encountered). Remember you are being **assessed on the software design** not the game design. Inclusion of the game design as a separate appendix might be useful for context, but it will not contribute towards the final grade.

Each member of the group should be allocated a number of the tasks. Care should be taken to **distribute these fairly**.

# Part 2 - Individual Members: Software Design of Individual Tasks

Each member should perform software analysis and design on each task that has been allocated. You do not need to follow a particular formal method for this, but you should show evidence of logical analysis of the problem (e.g. by using pseudocode and flow diagrams to work out the logic of each task) and derivation of classes with methods/attributes required to implement the task.

# Part 3 - Critical Conclusion of Group Work

Each group member should individually reflect on the group work that has been done, including the processes used, problems encountered and solutions found. Some discussion of strengths, weaknesses and improvements that could be made to the process should be considered.

# What you are required to hand in

Each group should submit one copy per member. All team members’ names must be clearly shown on the report front cover AND in the file name!

You need to **submit an electronic version online** (upload on the learn.solent.ac.uk website before the deadline).

The submission should be a **single .doc report**. The report should be presented as a formal document outlining the software design of the game, while giving a reflective account/justification/evidence.

1. How you and the group analysed the requirements to create the tasks and the process of allocating the tasks. If you reference other group members’ contributions it must be made clear which group member did what.
2. How you designed your allocated tasks, along with any problems encountered and the application of additional research performed.
3. A critical discussion of the methods, benefits and constraints of group work in the design process.
4. Supporting evidence, including Derived Tasks, Pseudocode, Flow Charts, Class Diagrams, Entity Relationship Diagrams, Dependencies, Testing Plans, Critical Elements, and Schedules.

Note: All work is expected to use correct spelling, grammar, references with images and equations cited in the text explicitly.

**Design/Analysis should be done as a group – but the report should be written by each individual.**

# Assessment Marking Scheme

The assessment is broken down into three core components. The breakdown and weighting of the different components is provided below. This provides a clear set of guidelines for the student on how the overall mark is calculated.

0 – no feature

1 – attempt (but incorrect)

2 – working (basic but needs work)

3 – outstanding (excellent example that is faultless)

**Documentation, Analysis, and Design (55%)**

Formatting (.doc, individual reports then design docs, consistent fonts) [3]

Report clear sections/ writing style [3]

Images/equations (referenced/cited within the text) [3]

References (e.g., external data, similar games, inspirations, and tools) [3]

Functionality, classes, and core requirements [3]

Flow diagrams (game mechanics) [3]

Development techniques (in relation to the design/game-play) [3]

Object-orientated design [3]

**Task Breakdown and Rational (35%)**

User story [3]

Critical paths, tasks, time scales, dependencies, proprieties [3]

Testing plans (how often, automated, user testing, ) [3]

Work breakdown structure (WBS) *(Task breakdown that is well thought*

*out and balanced, with good rationale)* [3]

Grid tasks/times (relates the work to the WBS, time, importance, …) [3]

**SMART** (specific, measurable, achievable, relevant, time) [3]

**Critical Reflection and Discussion of Group Work (10%)**

Evidence of equal distribution of work [3]

Reflection on the design process [3]

Identification and resolution of problems [3]

Version Control (GitHub 3+ weeks, evidence good industry practice and

gives a log of who did what and when) [3]

# Grading Criteria

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CRITERIA** | **A1 – A4** | **B1 – B3** | **C1 – C3** | **D1 – D3** | **F1 – F3** |
| **Documentation, Analysis, and Design**  **(learning outcomes K1, C1, P1)**  **(55%)** | **Complete design of operations of the game has been derived from requirements, *including* justification for the design of the classes.**  **Advanced functionality has been incorporated into the design beyond the core requirements**  **Design is elegant, and/or justifiably derived from multiple potential solutions.**  **Optimisation, extensibility and reuse are considered.**  **Wide ranging discussion of game/OO/3D development techniques and/or design changes.** | **Complete design of operations of the game has been derived from requirements, *including* justification for the design of the classes.**  **Significant extra functionality has been incorporated into the design beyond the core requirements**  **Design is complete, clear and justified.**  **Good discussion of game/OO/3D development techniques.** | **Complete design of the basic operations of the game, derived from the requirements.**  **Design should include a small number of features besides the core requirements.**  **Basic discussion of game/OO/3D development techniques.** | **Mostly complete design of the basic allotted tasks, derived from the requirements.**  **The design may be vague and inconsistent in parts.**  **Little discussion of game/OO/3D development techniques.** | **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.**  **No discussion of game/OO/3D development techniques.** |
|  | | | | | |
|  |  |  |  |  |  |
| **CRITERIA** | **A1 – A4** | **B1 – B3** | **C1 – C3** | **D1 – D3** | **F1 – F3** |
| **Task Breakdown and Rational**  **(learning outcomes C1, T1)**  **(35%)** | **Highly detailed list of User Stories, including story justification, refinements, task derivation, time scales, dependencies, priorities and critical paths.**  **Task breakdown well thought out and balanced, with good rationale.** | **Detailed list of user stories, including derivation of required tasks, time scales, task dependencies and priorities.**  **Task breakdown reasonably well thought out and balanced, with some basic rationale.** | **Well defined list of user stories, including derivation of required tasks and time scales.**  **A small number of tasks may be missing or ill-defined, or the task breakdown is suboptimal.** | **Basic list of User Stories, but lacking in detail.**  **Partial derivation of tasks of the core requirements of the game. The tasks may be vague and inconsistent in parts.** | **Inadequate understanding of problem leading to poor or non-existent attempt at designing user stories and deriving tasks.** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **CRITERIA** | **A1 – A4** | **B1 – B3** | **C1 – C3** | **D1 – D3** | **F1 – F3** |
| **CRITICAL DISCUSSION OF GROUP WORK**  **(learning outcomes C2)**  **(10%)** | **A thorough and detailed reflection on the group work design process with detailed explanation of what problems occurred and how they were or could have been resolved.** | **Detailed evidence of reflection on the group work design process, including what has been learnt and what improvements could be made** | **Evidence of reflection on the group work design process, including valid strengths and weaknesses.** | **Some evidence of reflection on the group work design process. May be brief or unclear in parts** | **Inaccurate and/or largely incomplete reflection of the group work design process.** |