**TEMASEK POLYTECHNIC**

**SCHOOL OF INFORMATICS & IT**

**DIPLOMA IN GAME DESIGN & DEVELOPMENT**

**AY2023/2024 OCTOBER SEMESTER (LEVEL 2) TERM A**

**GAME MATH AND PHYSICS (CGE2C15) TERM A**

**Project (40%)**



**PART 1 (15%)**

You must form groups of 3 students. There may have to be one group with only two students, depending on the class size – check with your tutor (the preference is groups of three).

Each group will submit a video presentation about one of the math or physics-related topics listed below.

* DESTRUCTIBLE OBJECTS
* GRAVITY manipulation
* VEHICLE (car or bike)
* CLOTH
* TERRAIN GENERATION
* WATER/BUOYANCY
* FLIGHT/JETPACK
* RAYCASTING
* PARTICLE SYSTEMS in Unity (focusing on forces and motion)
* PROJECTILES
* Your own suggested topic, but you must check with your tutor first.

Each group must choose a unique topic. You must inform your tutor of your chosen topic. Some suggested references for each topic are given in the comments section at the side of the page, but if you only present these, you’ll get a very low mark!

The video presentation should be in *two* parts, (a) and (b), as discussed below.

**Video Presentation Part (a)**

For each topic you must present the following sections:

THEORY – A *theoretical overview* of the topic, discussing the main problem to be solved (e.g., make an object destructible), with details of the math and physics involved.

UNITY IMPLEMENTATION OVERVIEW– An overview of how this might be *implemented in Unity*. This should include any core components needed to implement a solution to the problem (e.g., how to make an object destructible in Unity). You do NOT have to present an actual implementation (this will be done in Part 2).

CASE STUDIES – A discussion of how professional games have used the topic, to provide insights into how industry professionals tackle these challenges and the creative solutions they come up with. You must give at least THREE different example games. Choose one of these game for a more in-depth case study. (Note: this section will be the most difficult to score a good mark for, so don’t assume it’s an easy option!)

Each group member must present one of the sections above. If there are only 2 students in a group, the CASE STUDIES section should be ignored.

**Video Presentation Part (b)**

Each student in the group must then present their own implementation of the topic, as required by Part 2 below (page 4). This should include a description of the Unity components used and any code written.

The overall structure of the video should therefore be:

**PART (a)**

* Student1 THEORY
* Student2 IMPLEMENTATION OVERVIEW
* Student3 CASE STUDIES

**PART (b)**

* Student 1 own implementation
* Student 2 own implementation
* Student 3 own implementation

**Presentation slides format**

Refer to **GMAPS Project Guide.docx** for guidance on what type of content you might include in and how you might structure your presentation slides.

**Assessment criteria**

A brief explanation of the assessment criteria for Part 1 is given below.

The full marking rubric for Part 1 is at the end of this document.

**Topic Overview**

Covers the key concepts of the topic, e.g., for the technique of swinging/grappling, the mathematics of pendulum motion would be considered a key concept. If this is not covered, it would be considered a serious omission. If you only use the provided references, you will score a low mark!

**Focus on Details**

Each topic is deliberately quite wide, to give you scope to explore. But you cannot be too vague. You must focus on some key concepts in more detail, e.g., in the Unity Implementation section for swinging/grappling, you would be expected to focus in more detail on how the rope or chain is constructed from joints.

**Quality of Presentation Materials**

Your presentation slides must be well designed, with no spelling errors, poor design (e.g., images overlapping text, too small font, etc). If your materials appear to be simple copy/paste from reference websites, this will be considered poor design. Diagrams and screenshots are better than too much text—make sure you strike a good balance.

**Peer Teaching**

The point of this assignment is for you to peer teach your classmates about physics topics that we don’t have to time to cover in class.

All GMAPS students will eventually be given a link to your videos for their own independent learning. So, you must make sure that all teaching points are clearly stated (make sure everyone knows what you are talking about).

All topics must be covered in a clear manner with no confusion (if you don’t understand it, no one else will).

Finally, all topics must be communicated at an appropriate level for your classmates’ learning (e.g., no complex math equations that are not explained properly, no quotations from advanced research papers and textbooks, no incomprehensible diagrams, etc.)

Your combined video should be no more than 30 minutes in length.

**PART 2 (15%)**

Part 2 is to reinforce understanding of your chosen advanced math or physics topic by implementing a hands-on coding exercise in Unity.

You must each implement an example of your chosen topic and present this in Part (b) of your group’s video.

Each example must be different. For example, if you choose rope physics, then one student might implement a character swinging on a rope from one side of a chasm to another, while another student might implement a character walking across the chasm on a rope bridge.

Your group must submit a single Unity Project. The project must include a separate folder for each student’s work. Each folder should be named after the student whose work it contains (full name as in the register).

You CANNOT use any plugins to implement the core functionality, e.g., if you use a Unity plugin for rope physics, you will FAIL this component (zero marks).

A brief explanation of the assessment criteria for Part 2 is given below.

The full marking rubric for Part 2 is at the end of this document.

**Functionality**

This assesses the integrity of your code. Your code should execute without any errors and align with the stated objectives. Your code must also be relevant to the chosen topic.

**Complexity & Depth**

Did you go beyond the basics? Simple solutions that do just enough will score a pass grade, but adding more advanced features or doing something extra will score higher.

**Code Quality & Organization**

This relates to the logical organization, clarity of syntax, use of comments, and adherence to established coding standards and conventions, for your code.

**PART 3 (10%)**

Part 3 is the written part of your project.

You must submit **GMAPS\_Project\_Submission.docx**.

Rename this file to:

**Class**\_**Name-as-in-register\_GMAPS\_Project\_Submission.docx**

e.g., **P01\_Lee Wei Ming\_ GMAPS\_Project\_Submission.docx**

This is an individual submission on LMS.

You must include a YouTube link to your group’s video presentation in this document.

The requirements and marks awarded are in GMAPS\_Project\_Submission.docx.

The full marking rubric for Part 23 is at the end of this document.

*If you have any questions or need any clarification about any part of your project requirements, you must contact your tutor.*

*All rubrics are on the pages below*

**PART 1 – Presentation Rubric (15%)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criteria** | **Fail  (0 marks)** | **Poor  (1-2 marks)** | **Satisfactory  (3 marks)** | **Good  (4 marks)** | **Excellent  (5 marks)** | **MARK**  **/5** |
| **Topic Overview** | Omits major concepts and heavily relies on provided references. | Omits several key concepts or relies heavily on provided references. | Covers basic key concepts, primarily from provided references. | Covers most key concepts, with some independent research beyond provided references. | Comprehensive coverage of key concepts, showing a deep understanding and independent research beyond provided references. |  |
| **Focus on Details** | Vague and lacks specific focus. | Covers some details but lacks depth. | A general overview with some focus on specifics. | Detailed focus on many aspects of the topic. | Delves deep into specific aspects of the topic, shows a clear understanding. |  |
| **Peer Teaching** | Teaching points are unclear, confusing, and at an inappropriate level.  Poorly designed slides with many errors and mostly copy/paste content. | Some points are clear, but others are confusing or not appropriately pitched.  Slides have design errors or rely on a few direct copy/paste sections. | Most teaching points are clear and at an appropriate level, with a few areas of confusion.  Fair design quality with some balance between text and visuals. | Almost all teaching points are clear, with content well-tailored for the target audience.  Well-designed slides with minor errors or areas of improvement | All teaching points are clearly communicated, designed to maximize classmate understanding.  Professionally designed slides with a good balance of text, visuals, and original content. |  |
| **TOTAL/15** | | | | | |  |

**PART 2 – Implementation Rubric (15%)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criteria** | **Fail  (0 marks)** | **Poor  (1-2 marks)** | **Satisfactory (3 marks)** | **Good  (4 marks)** | **Excellent  (5 marks)** | **MARK**  **/5** |
| **Functionality** | Code doesn’t run or is incomplete. | Code has major bugs that impede functionality. | Code runs, but there are noticeable issues or missing features. | Minor bugs, but core functionality is intact. | Code runs without error.  Mechanic is fully implemented and operates smoothly. |  |
| **Complexity & Depth** | Very rudimentary or missing implementation. | Implementation lacks depth and seems rushed. | Basic implementation that meets minimum requirements. | Solid implementation with a few advanced techniques. | Goes beyond basic implementation, showcasing advanced techniques or unique features. |  |
| **Code Quality & Organization** | Code is unreadable, lacking structure or organization. | Code is somewhat organized but lacks clarity and comments. | Code is organized, with minimal comments. | Code is mostly clean and commented. | Code is clean, well-commented, and follows best practices. |  |
| **TOTAL/15** | | | | | |  |

**PART 3 – Documentation Rubric (10%)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criteria** | **Unsatisfactory (0 marks)** | **Needs Improvement (1-2 marks)** | **Satisfactory (3 marks)** | **Good  (4 marks)** | **Excellent  (5 marks)** | **MARK**  **/5** |
| **Technical Problems & Solutions** | No discussion of problems or solutions. | Mentions some problems but lacks detail on solutions. | Describes problems faced and provides some solutions. | Provides a detailed account of challenges faced and innovative solutions devised. | In-depth analysis of technical issues encountered and sophisticated solutions proposed. |  |
| **Reflection on Project & Subject** | No reflection or cursory statements without depth. | Limited reflection; lacks critical evaluation. | Reflects on both the project process and the subject, with some insights. | Thoughtful reflection with insights on project's highs and lows but somewhat generic evaluation of the subject. | Insightful reflection showcasing clear personal growth, understanding of the project's journey, and a personal evaluation of the subject. |  |
| **TOTAL/10** | | | | | |  |

**TOTAL / 40 =**