# CM4021 Games Development on the iPhone

**Individual Coursework**

**Hand-In Dates**

Submission One: 6/11/2015

Submission Two: 8/1/2016

**Brief: Game Development**

For the assessment for this module, you are required to build a game through Unity, which can be deployed to both the desktop and to a mobile device, usually the iPhone. The specific design of the game is for you to decide, but this assessment document will outline the marking scheme and the requirements to which your project must adhere.

The learning outcomes addressed by this coursework are as follows:

|  |  |
| --- | --- |
| 1. | Use the facilities available in the iPhone SDK for the development of a graphics application in conjunction with games design techniques to develop a touch and motion controlled game on the iPhone Touch. |
| 2. | Design, code and implement realistic 2D/3D graphics, for scene development and sprite animation, using blending and texturing techniques within a 3D graphics tool. |
| 3. | Employ game optimization and compression techniques, in conjunction with a game engine, to create an iPhone game. |
| 4. | Design, implement, test and document the overall touch and/or motion controlled game. |

This assessment is worth 100% of your module grade.

**Feedback Strategy**

Feedback for this module will be delivered according to the following strategy:

|  |  |  |
| --- | --- | --- |
| Type | When | Scope of Feedback |
| Individual | After submission of Game Brief | This feedback is intended to help you structure your planned submission for the module. You will received a signed off document that indicates approval for your project, along with guidance notes where appropriate. These guidance notes will then feed forward into your implementation. |
| Preliminary Group Feedback | After 25% of submitted games have been graded. | This feedback will be provided on the campus moodle, and will take the form of discussion notes based on observed patterns in the submitted work. All feedback will be generalised and non-attributable to the students in question, but will provide a big-picture overview of trends in the submissions. |
| Preliminary Individual Feedback | After your submission is marked | Your submission will be graded and weighted according to the marking scheme included in this document. You will also be given some general comments on your submission – these comments will be specific to you, although the group feedback will also be relevant. This is preliminary feedback, before external examination, and will be subject to adjustment. |
| Final individual feedback | After the course board sits | When your submission has been externally examined, you will receive your final grade for the module. |

Table 1 - Feedback Strategy

**Game Requirements**

There are no set genre requirements for your game, but there are certain elements from the course that you must incorporate to demonstrate mastery of the material. However, it is important to realise that the game you create for this module is not of the size, scope or sophistication of games which you may have played in the real world. Part of the submission for this module will be to scope out an achievable game design, which will form your submission. What you choose to do with your games outside the context of the module is up to you, and you can choose to exceed the brief if time and confidence permits.

However, there are some set requirements for what your submission must demonstrate. There must be at least one example of each of these in the submission:

1. Two scenes, each consisting of a game terrain, each consisting of a suitable plane, with applied textures, texture painting, and an appropriate skybox.
2. The storing and manipulation of persistent game state, through a script, or set of scripts, accessed externally by appropriate game objects. This might keep track of score, lives, health and so on. It should also feed data directly into other game objects or subsystems as required.
3. The inclusion of at least two game objects which implement the PhysX engine, along with the application of a custom physic material which you design yourself.
4. Some form of interaction with a game object, triggered through the user interface that you present.
5. WA level loader system implemented either through transitions within the game environment, or through a GUI selector.
6. Persistence of player state data between executions of the game, implemented through an appropriate save and reload system.
7. Scripts which implement actions for the following:
   1. A player entering a trigger zone
   2. A rigidbody colliding with the player
   3. A rigidbody colliding with another rigidbody
8. User input via mouse, keyboard and/or touch controls which trigger the instantiation and placing of prefabs within the game environment.
9. An example of using raycasting to identify objects within the player’s direct field of view.
10. A game object in a scene which reacts to the playing of an audio triggered in some way by the player.

There is no set requirement on how these elements are to be included, as these will vary from game to game. However, each of these must be met and you are required to indicate where they are met in the report which will accompany your submission. If your game design cannot possibly meet one of these requirements, check with the module lecturer to see if an alternative criteria can be considered.

Your game should hold together as a coherent entity – you will be graded in part as to how integrated these elements are in the game which you submit. The game you submit then should not be an isolated collection of implementations of these requirements – it should show evidence of deeper planning of interactions.

It is not necessary that your game be fully featured, or even **fun**. However, you should consider how best to implement these features to help bring out these qualities. Table 2 shows an example of an integrated design for a stealth game which meets all the required elements. You will be required to submit a similar table for the first part of your submission.

|  |  |  |
| --- | --- | --- |
| Requirement | Summary | How Met |
| 1 | Scenes | Two levels of the game, each representing the interior of a different house. |
| 2 | State | A game state system which handles treasure gathering and health of the player. |
| 3 | PhysX | Guard objects implemented as rigidbodies, and treasure chests. |
| 4 | Interaction | Treasure chests will open when clicked, depositing ‘treasure objects’ around them. |
| 5 | Levels | Levels loaded through trigger areas reached when the player has made it through to the end of the terrain. |
| 6 | Persistence | Game stores achievements for the player such as ‘most treasure found’ and ‘longest time spent hidden’ |
| 7 | Scripts | The following behaviours are implemented:   * When the player enters a trigger area, they are transitioned to the next/previous level. * When a guard object collides with the player, the game is over. * If a guard object collides with a player’s projectile, they are knocked unconscious. |
| 8 | Input | Touch controls handle moving around. Tapping the screen opens a chest.  Additionally, mouse controls and keyboard controls permitted. |
| 9 | Ray-casting | The player is visible to a guard if there is a ray-cast line between the way the guard is facing and the player. |
| 10 | Audio trigger | When the player moves, footstep sounds are triggered. When a guard hears a footstep, they turn towards the player. |

Table 2 - Table of Met Requirements

# Submission One: Project Brief

There are two submissions for this module. The first the form of an outline for the project you intend to develop. This will be signed off by your module leader to ensure that the extent of content is appropriate for the level of the module. This submission should include the following, produced as a 1000 word report and an accompanying table:

* A table showing where each of the requirements for the brief have been met by your game design, as shown in table 2 above.
* A discussion of the game, its key mechanics, and the thematic consistency.
* An overview of the **accessibility** of your game, and where there may be room to improve this.
* A user-testing plan which incorporates some form of external, qualitative testing of the game you have developed.

This submission is graded as part of your overall submission for the module, but you will receive an electronic sign-off and individual guidance notes as outlined in table 1.

This submission has a hand-in date of XXXX on the drop-box on the Campus Moodle

# Submission Two: Game Submission

Submission two is the primary submission for the module, and consists of the full game code you develop.

You are also required to produce a short screencast or video of your running game. This screencast should demonstrate all of the requirements you met as part of Submission One: Project Brief. You should annotate your screencast, either with a voiceover or with textual comments contained within the screen – these should identify the specifics of where your requirements are met, and any additional information you want to bring to the viewer’s attention. For example, if you’ve been particularly clever with something you’ve accomplished, make a note of it in the annotation.

You should provide your submission for this assessment in the form of a zip file of the full unity project file and all associated assets, and upload this to the course drop box by the submission date outlined above. Your zip file should also include your project report as outlined above. For this module, it is highly possible that your submission will be too large to submit via the Campus Moodle – when this is going to be the case, an external Dropbox or CD submission can be provided, but check this in advance with the module lecturer – there are digital storage and auditability considerations that must be handled on a case by case basis.

This submission has a hand-in date of XXXX on the drop-box on the Campus Moodle.

**Marking Scheme**

**Student Name Final Grade**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **GRADE** | **A** | **B** | **C** | **D** | **E** | **F** |  |
| **DEFINITION**  **Design Documentation**  **(30%)** | EXCELLENT | COMMENDABLE | GOOD | SATISFACTORY | BORDERLINE FAIL | FAIL |  |
| Design documentation effective and contain few minor omissions. | Design documentation reasonably effective but contains some minor omissions. | Design documentation mostly effective but contains a major issue and some minor omissions. | Design documentation contains several major and minor issues. | Design documentation flawed, and contains major and minor issues that render it a poor basis for implementation. | Design documentation not a suitable basis for implementation. |  |
| Fully demonstrates where all requirements have been met, | Fully demonstrates where most requirements have been met, | Fully demonstrates where at least half of the requirements have been met, | Demonstrates where some of the requirements have been met. | Partially demonstrates where some of the requirements have been met. | Demonstration where few or none of the requirements have been met. |
| Excellent screencast which shows all requirements that were met. | Very good screencast which shows most requirements that were met. | Good screencast which shows at least half of the requirements that were met. | Satisfactory screencast which shows some of the requirements that were met. | Poor screencast which shows few of the requirements that were met. | Unsatisfactory screencast which shows no requirements being met. |
| Contains a high quality overview of accessibility considerations. | Contains a good quality overview of accessibility considerations. | Contains a reasonable overview of accessibility considerations, although this overview has issues of correctness or relevance. | Contains an overview of accessibility considerations, although this overview has issues of correctness or relevance. | Accessibility concerns and considerations are noted, but not evaluated. | No suitable discussion of accessibility considerations, |
| **Implementation**  **(40%)** | Coding style excellent and considerable attention paid to efficiency of code. | Coding style very good and some attention paid to efficiency of code. | Coding style good but may have efficiency or stylistic errors. | Coding style satisfactory but has many efficiency or stylistic errors. | Coding style poor and has many efficiency or stylistic errors. | Coding style extremely poor and has many efficiency or stylistic errors. |  |
| No significant errors in implementation. | Some minor errors in implementation. | At most one major error in implementation along with several minor errors. | Several major errors in implementation along with several minor errors. | Submission is heavily flawed with major and minor issues. | Submission has very significant problems with implementation. |
| All requirements fully met as outlined in the brief. | All requirements fully or partially met as outlined in the brief. | Most requirements fully or partially met as outlined in the brief. | At least half of the requirements fully or partially met as outlined in the brief. | Less than half of the requirements fully or partially met as outlined in the brief. | Few or no requirements partially or fully met. |
| **Testing**  **(15%)** | All of implementation tested. | Almost all of the implementation tested. | Most of implementation tested. | Some of the implementation tested. | Little of the implementation tested. | None of the implementation tested. |  |
| Excellent review of qualitative testing with users. | Very good review of qualitative testing with users. | Good review of qualitative testing with users. | Satisfactory review of qualitative testing with users. | Review is present, but has serious quality issues in terms of correctness or relevance. | Review is not present or not satisfactory. |
| **Thematic Consistency**  **(15%)** | All game elements coherently work together to provide a consistent game experience. | Most of the game elements coherently work together to provide a consistent game experience.. | At least half of the game elements coherently work together to provide a consistent game experience. | Some of the game elements coherently work together to provide a consistent game experience. | Few of the game elements coherently work together to provide a consistent game experience. | No consistent game experience is presented. |  |
| **Comments** |  | | | | | |  |