Southampton Solent University

Coursework Assessment Brief

# Assessment Details

|  |  |
| --- | --- |
| Unit Title: | Mobile Game Fundamentals |
| Unit Code: | CGP503 |
| Unit Leader: | Kostas Dokos |
| Level: | 5 |
| Assessment Title: | AE2 – Software Product: Swashbucklers |
| Assessment Number: | 1 |
| Assessment Type: | Software Product and Report |
| Restrictions on Time/Length : | 1500 words plus source code and testing |
| Individual/Group: | Individual not including tables of appendices. |
| Assessment Weighting: | 70% |
| Issue Date: | 25th Sept 2017 |
| Hand In Date: | 12/01/2018 |
| Planned Feedback Date: | Four weeks after submission date |
| Mode of Submission: | Online |
| Number of copies to be submitted: | 1 |
| Anonymous Marking | This assessment will :  be marked anonymously |

## Scenario

## You are a one man team, developing games for a living. One day you get a phone call from your pal, Jimbo, who is a game designer intern at a big Studio. He is urgently asking you to make a working prototype of a mobile game about the life of Pirates, called “Swashbucklers”. You are to help your pal out with your insane mobile programming skillz. You have until the 12th of January, else Jimbo will get fired! He convinced the studios to entrust YOU with their prototype.

## The studio has supported your cause with some very basic prototype assets and a Game Design Document that your pal has written for them. You shouldn't have any problems with it hopefully. You are expected to deliver this working prototype with those assets in place. The studio will then add actual high quality assets into this game and build up upon it. It’ll then be released into the wild, hopefully making hundreds of pounds. Jimbo will also become a full-time game designer, so many things are at stake here!

## Swashbucklers GDD

Swashbucklers is a game about the life of pirates. You are the captain of your trusted ship, “The Killer Chipmunk”, and you go out in adventures, gathering loot, destroying enemy ships and casually upgrading your ship in friendly islands while also managing your crew members.

The game is played out in two phases. Phase 1 is about **Management**. You are docked at a friendly island, where you can visit one of the 2 available buildings and utilize their offered service.

* In the **Market**, you trade your cargo for gold
* In the **Shipyard**, you upgrade your ship using gold

We will analyse them later

Phase 2 is about **Naval Combat**. You are able to control your ship in a vast sea level and destroy other enemy ships that attempt to attack you. After you sink your foes, you are able to sail over the dropped loot and gather it as cargo. When combat is over, you return to your island to upgrade your ship.



Figure 1 - This is a sample of Naval Combat. Ignore the health bars and the Islands.

So, that’s your game overview!

In order to fully understand what our ship can do, we will first analyse phase 2.

Our ship has the following stats:

1. **Hull Health:** If this reaches zero, the ship sinks. This is an upgradeable stat in the shipyard.
2. **Sail Health**: This determines the movement speed as well as turn speed. We move at 100% our movement speed and turn at 100% our turn speed if our sail health is at 100%. We move at half our speed and turn at half our turn speed when sail health is 50%, and we stop moving and turning if our sail health reaches zero. Maximum sail health does NOT get upgraded, and it has a value of 100. However, we can upgrade our sails so as to increase turn speed.
3. **Cannonball Damage**: As we shoot our 3 cannonballs, each cannonball that collides with the enemy causes damage based on the formula:  
     
   Final Damage = (Cannonball Base Damage + Random) \* Defender Ship Quality  
   Random = [Cannonball Base Damage \* -30% of Base, Cannonball Base Damage \* 30% of Base]

*Example:   
Base Damage = 100. Random would be -30 to 30. Let’s say it randomizes to 20.  
Final Damage would be 100 + 20 = 120. We multiply this with the quality of ship under attack. If quality score is 0.5, then final damage would be 60.*

1. **Cannonball Cooldown:** Every time, we attack, towards a specific direction, we have a 3 second cooldown for that specific direction.
2. **Cannonball Range:** Cannonballs only fly towards a direction for 1.75 seconds. After that time passes, they are hitting the sea and sink.
3. **Cargo pull**: There is a radius of pulling loot in cargo. The radius is basically a circle with a radius of X units (Defined by developer). If a loot is in that radius, then it is pulled towards the ship. When the cargo touches our ship, the cargo is destroyed and we gain the cargo. This happens only if our ship can support it. The movement speed towards our ship is a constant. It can be any value you like.

* Our Ship is **moving always at a constant speed**, towards the direction that the **front of the ship is facing**. The speed is common for all ships. It could be any value you feel is ok.
* We are able to turn left or right at our own will through touch. This simply turns our ship, thus changing facing direction.
* We are also able to attack with cannonballs through touch. We can either attack towards our left or our right. Every time we attack, we shoot **3 cannonballs** **in an arc** towards either our left or our right. The difference between the cannonballs in angles is 45 degrees.
* We have 3 types of enemies: The Sloop, the Caravel and the Brigantine. Those enemies have their own pseudo-AI which has the following traits:
  + Step 1: They chase after us
  + Step 2: Once they are close in a certain range they start turning. The range is something that the developer should decide.
  + Step 3: Once they have clear line of sight towards us, they start shooting. The enemies shoot only 1 cannonball, and they know they have clear line of sight based on a physics calculation.
  + Step 4: They repeat from step 1 or 2 depending on a condition.
* When we take damage from enemies, we have a 20% chance to apply this to Sail Health.
* When enemies are being destroyed, they **drop loot**. More specifically, they drop a randomized amount of gold as well as cargo. The loot is determined by a loot table found in the appendix. The loot appears within a radius of 5 units of the destroyed ship. The loot stays in water for 20 seconds and then if not picked up, it’s lost forever as it sinks.
* For each different loot dropped, there is a chance that it will be one of the mentioned cargo types based on the table at the appendix. There is a chance for a cargo to be dropped **multiple times**. For example one ship may drop Grain and Silk, while another one may drop Grain and Grain.
* Combat is over when we are sunk or all enemies are sunk.
* If we get sunk, we lose all our collected cargo, 25% of our gold, and we return to our island.
* If all our enemies sink, we can free roam the level for 10 seconds. We are then automatically teleported to our island.
* The whole phase is played out in an open-sea level with no boundaries or obstacles.
* If we collide with an enemy, we take 50% damage of our **max health**. This can happen once every 10 seconds.

Now, let’s analyse phase 1.

While at our island, we can use the Market and the Shipyard, or we can go to a “Sea Raid”. The Sea Raid actually gets us to phase 2. Once we go to phase 2, we appear in a vast sea, and our enemies get randomized based on a table found in the appendix.

The **Market** can be used to trade-in the looted cargo you got from enemy ships for gold. A table with the full pricing of each available cargo is found further below.

The **Shipyard** can be used to trade-in gold for ship upgrades. The cost of these upgrades should be increased in a linear fashion as the upgrades are happening. The upgrades are the following:

* 1. **Sails**(Increases Maneuverability – Turn Speed)    
     (x 1.1, x 1.2, x 1.3, x 1.4, x 1.5, x 1.6, x 1.7, x 1.8, x 1.9, x 2.0)   
      The numbers above indicate the rate that the ship can turn.
  2. **Base Cannonball Damage**(Increases Damage)    
     (20, 25, 30, 35, 40, 45, 50, 55, 60, 65)   
      The numbers indicate the damage that each cannonball can do
  3. **Hull**(Increases Health)    
     (50,60,70,80,90,100,110,120,130,140,150)   
      The numbers indicate the maximum health that the ship will have
  4. **Storage**(Increases Max. Cargo storage)   
     (10, 15, 20, 25, 30, 35, 40, 45, 50, 60)   
     The numbers indicate the maximum number of cargo the ship can hold at one time
  5. **Quality**(Increases Armor)    
     (x 0.95, x 0.90, x 0.85, x 0.80, x 0.75, x 0.70, x 0.65, x 0.60, x 0.55, x 0.50)  
     The number above is multiplied by the damage to calculate final damage

**Appendix**

**Sea Raid**

|  |  |
| --- | --- |
| Percentage | Enemies |
| 60% | 2-4 Sloop 0-1 Caravel |
| 30% | 2-6 Sloop 1-3 Caravel 0-1 Brigantine |
| 10% | 7 Sloop 2-6 Caravel 2 Brigantine |

**Loot Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cargo | Price | Drop Chance (Sloop) | Drop Chance (Caravel) | Drop Chance (Brigantine) |
| Grain | 1 Gold | 25% | 22% | 18% |
| Fish | 2 Gold | 22% | 19% | 16% |
| Oil | 3 Gold | 18% | 15% | 13% |
| Wood | 5 Gold | 13% | 12% | 12% |
| Brick | 8 Gold | 10% | 11% | 11% |
| Iron | 10 Gold | 8% | 9% | 10% |
| Rum | 15 Gold | 4% | 6% | 8% |
| Silk | 20 Gold | 0% | 4% | 5% |
| Silverware | 30 Gold | 0% | 2% | 4% |
| Emerald | 50 Gold | 0% | 0% | 3% |

**Ship Statistics and Drop Rules**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ship Type | Gold Dropped | Cargo Dropped | Health | Damage | Turn Speed | Quality |
| Sloop | 2-8 | 0-2 | 25 | 15 | x1.0 | x0.8 |
| Caravel | 4-15 | 1-4 | 50 | 30 | x1.2 | x0.75 |
| Brigantine | 8-25 | 3-8 | 90 | 50 | x1.5 | x0.7 |

## Requirements

## Jimbo says that, first of all, you need to make sure that your game is properly exporting and running on a mobile phone! (5%)

## Tier 1 Features: Powder Monkey (10% for the whole Tier)

To complete this module with a basic pass grade you must write a report detailing your design, and implementation of the product whilst completing the following features in Unity:

**Management Phase** (2%, break down explained below)

1. The game begins with the Management Phase. (0.2%)
2. Management Phase and Naval Combat Phase have their own Scenes (0.2%)
3. The player should be able to enter the “Market” or the “Shipyard” (0.2%)
4. There should be an option to do a “Sea Raid” , from the Management Scene (0.2%)
5. The game should navigate smoothly from Management to Naval Combat if the player wishes to do so, and the enemies spawned should follow the rules of Sea Raid in appendix. (1%)
6. The game should play a unique music track when we are in Management Phase. (0.2%)

**Naval Combat – Generic** (8%, break down explained below)

1. The game should play a unique music track when we are in Naval Combat Phase. (0.2%)
2. The game should play sound effects when a ship is shooting cannonballs, when a cannonball hits the sea and sinks, when a ship sinks, when a ship collides with another ship, when cargo is picked up and when a cannonball hits a ship. (0.5%)
3. The player should be taken to Management Phase once he is sunk. (0.3%)
4. The player should navigate to Management Phase 10 seconds after destroying all his enemies. (1%)
5. The player should lose 25% of gold and all cargo if his ship is destroyed. (1%)
6. All physics in the game must obey Physics2D rules. No Box Colliders or 3D Raycasts. (2%)
7. The Sloop should be smaller than our ship. (0.33%)
8. The Caravel should be like our ship in size. (0.33%)
9. The Brigantine should be bigger than our ship in size. (0.33%)
10. Winning Condition: All enemy Ships are sunk. (1%)
11. Losing Condition: Our Ship is sunk. (1%)

**Tier 2 Features: First Mate** (10% for the whole Tier)

To achieve an average pass in this unit, you must complete the previous features as well as the following:

**Naval Combat – Ship Stat System** (4%, break down explained below)

1. Our ship has its’ own “**Storage**” Stat. (0.3%)
2. All ships should have their own “**Hull Health”** Stat and should work as stated. (0.4%)
3. The player can clearly see the Hull Health left at any time (GUI). (0.3%)
4. Our ship has its’ own “**Sail”** Stat. (0.2%)
5. The player can clearly see the Sail Health left at any time (GUI). (0.4%)
6. The Sail Health stat should work as intended in the design. (0.4%)
7. All ships should have a “**Base Cannonball Damage**” Stat and should work as stated. (1%)
8. All ships should have their own “**Quality**” stat and should work as stated in design. (1%)

**Naval Combat – View** (3%, break down explained below)

1. The Naval Combat level should be an Open-Sea level with no boundaries. (0.5%)
2. The camera view point is top-down. (0.5%)
3. The camera should always follow the ship in regards to the position. Ie, the ship should always be in the middle of the screen. (1%)
4. The camera should NOT follow the ship in regards to rotation. Ie, if the ship turns around, we want to see it turning, rather than turning the camera along with it. (1%)

**Naval Combat – Movement** (3%, break down explained below)

1. The ship is always sailing towards the direction its’ facing. (1%)
2. There should be visible feedback that the ship is actually moving forward. This should be given in the form of a background, or particle system (0.5%)
3. The player should be able to turn the ship left and right using touch input. (1%)
4. The ship’s sprite should give the impression that it is turning. (0.5%)

**Tier 3 Features: Quartermaster** (10% for the whole Tier)

To achieve a highest than average pass you must complete all the above features as well as the following:

**Naval Combat – Combat System** (7%, break down explained below)

1. The ship should be able to shoot 3 cannonballs left or right. (0.5%)
2. The cannonballs should be shot in an arc with 45 degrees difference from each other. (0.5%)
3. Left and Right cannonball attacks have their own unique 3 second cooldown. (0.3%)
4. The user should be able to see a unique live cooldown for each cannonball attack. (0.3%)
5. The user should not be able to shoot cannonballs as long as they are in cooldown. (0.4%)
6. The enemy ships should be destroyed when their health reaches zero. (0.5%)
7. The player should sink if his hull health reaches zero. (0.5%)
8. Cannonballs should be able to inflict damage to enemy ships when they collide. (0.3%)
9. Cannonballs should be able to inflict damage to enemies based on the damage formula given. (1%)
10. Cannonballs that haven’t reached enemy ships for 1.75 seconds should sink. (0.3%)
11. Cannonballs that have either sunk or hit enemies should be removed from the game. (0.3%)
12. Each Cannonball should have its’ own graphic (0.1%)
13. If we collide with an enemy, we take 50% damage of our **max health**. This can happen once every 10 seconds. (1%)
14. When we take damage from enemies, we have a 20% chance to apply this to **Sail** Health. (0.5%)
15. Enemies don’t have Sail stat, so they always take the damage to their **Hull** stat. (0.5%)

**Upgrade System** (3%, break down explained below)

1. The game should have sound effects played when the user is clicking on a button, and when an upgrade is taking place. (0.5%)
2. The upgrade system should have a GUI helping the player choose an upgrade as well as the current upgrade level of each of the ship’s stats, and the costs. (0.3%)
3. The player should also know how much gold he/she has at the moment at the aforementioned GUI. (0.2%)
4. The player should be able to use up his gold to upgrade his ship in the 4 different categories. (0.3%)
5. The gold cost should be increasing in a linear fashion as each upgrade is getting applied. (0.4%)
6. Each upgrade should have its own unique cost. (0.3%)
7. Each upgrade should be reflected in gameplay clearly. (1%)

**Tier 4 Features: Captain** (10% for the whole Tier)

To achieve the highest pass, you must complete all previous features as well as the following:

**Naval Combat – Cargo System** (4%, break down explained below)

1. Our player can clearly see the Current and Maximum Cargo based on Storage Stat at any time. (0.2%)
2. The cargo should appear within a radius of the destroyed ship. (0.3%)
3. The “**cargo drop radius”** can be as big or small as required but should be realistic, ie it shouldn’t be x3 times bigger than the ship itself. (0.3%)
4. The cargo appeared should follow the Drop Rules, depending on each ship type. (1%)
5. The cargo should stay in water for 20 seconds and then sink. (0.2%)
6. The player should be awarded gold based on Drop Rules after an enemy ship is sunk. The gold received should obey the cargo rules found in the appendix. (1%)
7. The player should be able to pull cargo as soon as the cargo is within a radius. (0.25%)
8. The “**cargo** **pull radius**” can be as big or as small as required by should be realistic, ie it shouldn’t be x2 times bigger than the ship itself. (0.25%)
9. The player should be able to pick up cargo as soon as the cargo touches the ship. (0.25%)
10. The player should only be able to pick up cargo if he has space left. (0.25%)

**Market System** (3%, break down explained below)

1. The market should have its’ own GUI the player can interact with, with a “Sell All” button. (0.5%)
2. The game should have sound effects played when the user is clicking on a button, and when cargo is exchanged for gold. (0.5%)
3. A player should be able to trade-in all cargo with one button click, and exchange all of it for gold. (1%)
4. There should be visual indication as to how many types of cargo we currently have in our storage (GUI). (1%)

**Naval Combat – Enemy Ship AI** (3%, break down explained below)

1. The enemy ships should be able to sail towards our ship. (0.5%)
2. The enemy ships should start turning once they have reached a specific distance from our ship, so they can move on to shoot. (1%)
3. Once the enemy ships have clear sight of our ship, they start shooting cannonballs. (1%)
4. The enemy ships only shoot 1 cannonball instead of 3. (0.2%)
5. The enemy ships have a 3 second cooldown between their attacks. (0.3%)

***THERE ARE NO MARKS FOR EXTRA ART***

## The report

The supplementary report must contain the following headings

### Design (10%)

Here you will break down the stories into smaller tasks with diagrams on how each task can be accomplished using all or some of the following: flowcharts, UML, pseudo code. This **MUST** be done **before** you start your implementation

### Implementation and Source Code (15%)

A print out of your source code must be included. Marks will be divided based on your code in the following categories:

* Code Clarity & Nesting (5%)
* Variable Naming & Comments (5%)
* OO Design (5%)

### Testing (10%)

The testing part is not included in your word count, as you never know how many issues you’ll have.

To help you out, testing should have the following table format. You are supposed to carry out a testing plan for each feature you built in this game. In this table, you are required to write down which feature you tested, which problem or bug occurred, the severity of it in regards to your game, the cause of the bug, steps to reproduce the bug and the solution you thought about.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Feature ID | Bug | Severity | Cause | Reproduction | Solution |
|  |  |  |  |  |  |

*Reflection* (10%)

State your reflection in regards to this project. You are required to reflect about its’ design, implementation, and testing, how you did it, which parts of it were challenging and what would you change if you would.

You are also required to state how you would assess your own work, in terms of marking. Would you give it a C? Would you give it an A? And why would you think that mark is appropriate. This is to get an insight on your self-assessment methodology and see how good you can be a judge of your own work.

*Video Demo* (5%)

A short, 1-min video that will be used for your future showreel, showing off the mechanics you built for this prototype.

## What you’re required to hand in

You will be uploading your submissions online through SOL. This will be done in the form of a .zip or .rar file containing a folder labelled **[LastName]\_[FirstName]\_[Student Number]\_CGP503\_AE2** which contains the following:

* Your completed APK file
* Windows Executable File
* Unity Project with Source Code
* Personal Report
* A short video demonstrating the game you built

## Assessment criteria

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CRITERIA** | **F3 – F1** | **D3 – D1** | **C3 – C1** | **B3 – B1** | **A4 – A1** |
| **Design**  **(10%)** | No effort to describe any of the stories | Stories are broken down into overly complex or incomplete tasks.  All user stories for the D grades, have been designed. | Stories are broken down into simple, well thought out tasks  All user stories, for the C grade, have been designed. | Stories broken down into comprehensive tasks and good thought given to how each task could be solved.  All user stories for the B grades, have been designed. | Designs completed for each story attempted with no errors or omissions and are of high quality.  All user stories for the A grades, have been designed. |
| **Software Product**  **(40%)** | Not all stories are complete or hasn’t been demonstrated to work on a phone | All stories for the D grades have been implemented | All stories for the C grades have been implemented | All stories for the B grades have been implemented with few bugs | All stories for the A grades have been implemented with no bugs |
| **Implementation**  **(25%)** | Code is messy and very difficult to read | Code is not commented effectively.  Code works but is not well thought out or of high quality.  Code is overly complex  Code is inefficient.  Makes little or no use of code reuse. | Code is well thought out, good commenting, basic use of methods.  Use of inheritance and code reuse.  Code is overly complex in some, but not all methods.  Code is efficient.  Attempts use of advanced features such at state machines, coroutines, etc. | Code is very well thought out, good use of commenting.  Good use of inheritance and code reuse throughout program.  Good use of advanced features such at state machines, coroutines, etc. | Code is extremely well thought out, wise use of commenting, and shows good use of encapsulation.  Excellent use of inheritance and code reuse.  Excellent use of advanced features such at state machines, coroutines, etc.  Application is of release quality throughout. |
| **Testing**  **(10%)** | Little or no testing done | Basic testing of obvious faults | Finds a small number of issues, describes how they were fixed | Finds a selection of issues, describes the severity and how they were fixed.  Tests application with several testing techniques. | Covers a wide range of testing, covers their severity, reproduction steps, and if they were fixed, how.  Tests applications thoroughly with several testing techniques. |
| **Reflection**  **(10%)** | Little or no reflection on the product, testing or design | Student has provided some reflection on the final product, testing and design but has not fully considered the whole project or the ramifications of any obvious or identified issues. | Student has provided a reasonable reflection on the final product, testing and design.  Student has identified strengths and weaknesses from the project | Student has provided good reflection on the final product, testing and design.  Student has identified strengths and weaknesses from an over view of the whole project.  Student has suggested methods to improve weaknesses. | Student has provided excellent reflection on the final product, testing and design.  Student has identified strengths and weaknesses from an over view of the whole project. And has discussed their impact.  Student has suggested methods to improve weaknesses and has provided rational with good designs where necessary. |
| **Document Structure and Bibliography**  **(5%)** | Document is littered with incorrect spelling or grammar.  No attempt to format the document to a professional standard has been made. | Some attempt at appropriate spelling and grammar.  Some attempt at formatting the document to a professional standard.  Little or no thought given to presentation of document. | Appropriate use of formatting, including headings, main content, page headers and footers, to a professional standard.  Mostly correct spelling and grammar, throughout the document.  Over all good presentation of document. | Good and appropriate use of formatting, including headings, main content, page headers and footers, to a professional standard.  Correct spelling, grammar and punctuation throughout the document.  Over all good presentation of document.  Appropriate use of language | Excellent and appropriate use of formatting, including headings, main content, page headers and footers, to a professional standard.  Excellent spelling, grammar and punctuation throughout the document.  Excellent presentation of document over all.  Excellent use of appropriate language. |

## 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>