ASSESSMENT AND INTERNAL VERIFICATION FRONT SHEET (Individual Criteria)

(Note : This version is to be used for an assignment brief issued to students via Classter)

Course Title	**IT6-A4-23 BSc Year 1 - Digital Games Development**		**Lecturer Name & Surname**	**Owen Sacco**	
Unit Number & Title	ITDGM-506-2303 - Procedural Content Generation				
Assignment Number, Title / Type	Procedural Content Generation Home Assignment				
Date Set	22/03/2024	**Deadline Date**	21/05/2024		
Student Name		**ID Number**		**Class / Group**	

Assessment Criteria	**Maximum Mark**
SE2.4: Generate meshes and shapes that can be generated through code in games	10
KU3.1: Describe how noise, voronoi tessellation, midpoint displacement, splat maps and L-systems can be used to generate terrains	5
AA3.2: Produce and develop code for generating trees and vegetation	7
AA3.3: Produce and develop code for generating natural elements (for example but not limited to: erosion, rain, wind, clouds, fog, sky etc.)	7
SE3.4: Generate random terrains	10
KU4.1: Describe different techniques for generating random levels	5
AA4.2: Produce and develop code that generates random levels	7
SE4.3: Generate random levels in games	10
Total Mark	61

Notes to Students:

- This assignment brief has been approved and released by the Internal Verifier through Classter.
- Assessment marks and feedback by the lecturer will be available online via Classter (Http://mcast.classter.com) following release by the Internal Verifier
- Students submitting their assignment on Moodle/VLE will be requested to confirm online the following statements:

Student's declaration prior to handing-in of assignment

❖ I certify that the work submitted for this assignment is my own and that I have read and understood the respective Plagiarism Policy

Student's declaration on assessment special arrangements

❖ I certify that adequate support was given to me during the assignment through the Institute and/or the Inclusive Education Unit.

❖ I declare that I refused the special support offered by the Institute.

ITDGM-506-2303

Procedural Content Generation

Assignment: Procedural Content Generation

Home Assignment

Assignment Guidelines

Read the following instructions carefully:

- The assignment coversheet should be the first sheet in your assignment. Moreover, the coversheet should be fully completed with all the necessary details.
- You are required to answer **ALL** questions.
- You are required to use Unity and C# for Section A of this assignment. All tasks should be developed through code and not created through any designer tools unless instructed otherwise.
- All text/code must be properly referenced. In the absence of proper referencing, the assignment will be regarded as plagiarised.
- **Copying is strictly prohibited and will be penalised** in line with the College's disciplinary procedures.
- The deadline to submit all deliverables is **21/05/2024**
- You need to submit all your assignment deliverables (as explained in the assignment brief) via the links on Moodle/VLE.
- You need to commit all your C# scripts and project files to a Git repository for **Section A (only)**. It is important to commit your code in stages to show your progress whilst working on this assignment. Include clear comments and notes to explain key concepts in your code.

- You are required to record a (video) demonstration of all your deliverables for **Section A (only)** and upload your recording to any streaming service of your choice such as Microsoft Stream or YouTube. Ensure that you set the correct permissions and share it only with your lecturer. Kindly limit the recording to not more than 20 minutes.
- You are required to submit **Section B (only)** as a PDF document via the link on Moodle/VLE.

Section A - Development

Task 1 – Procedural City Generation

Task 1.1 – [SE2.4] Generate meshes and shapes that can be generated through code in games – 10 marks

In this task, you are requested to programmatically develop a city generator that generates a city consisting of randomly positioned generated buildings and a road network adjacent to the buildings out of primitives by using a mesh generator that procedurally generates the primitives (such as cubes). All objects should be assigned materials programmatically. Develop at least 5 buildings within the city.

Task 3.1 - [AA4.2] Produce and develop code that generates random levels – 7 marks

In this task, using the city generator developed in Task 1.1, you are requested to programmatically generate levels consisting of at least 2 levels – i.e. 2 different random programmatically generated cities consisting of different randomly positioned generated buildings and a different generated road network adjacent to the buildings. Each city should have at least 5 randomly positioned generated buildings. All materials should be assigned to the objects programmatically.

Task 3.2 - [SE4.3] Generate random levels in games – 10 marks

In this task, you are requested to turn the generated levels developed in Task 3.1 into an explorable game. You are requested to programmatically add a player controller to the game. The player controller can be any car controller freely provided by the Unity Asset store. A random start position should be programmatically added to each city and the player controller should be positioned at this starting position. A level ends when the player explores the whole city until the player controller reaches an end position. Once a level ends, the next level (i.e. city) loads. The game finishes once all 2 levels (i.e. 2 cities) have been explored. You are not required to add any opponents within the game.

Task 2 – Procedural Terrain Generation

Task 2.1 - [AA3.2] Produce and develop code for generating trees and vegetation – 7 marks

In this task, you are requested to develop a random landscape programmatically with different heights as mountains along the landscape. You are also requested to programmatically add vegetation and trees at random locations along the landscape.

Task 2.2 - [AA3.3] Produce and develop code for generating natural elements – 7 marks

In this task, you are requested to programmatically generate random natural elements (such as but not limited to water, river, rain, wind, clouds, fog, sky etc.) to the landscape developed in Task 2.1. You are requested to programmatically generate at least 3 natural elements of your choice at random locations along the landscape.

Task 2.3 - [SE3.4] Generate random terrains – 10 marks

In this task, you are requested to add final touches (such as more mountains, more trees and/or vegetation, and more natural elements) to your landscape developed in Task 2.1 and Task 2.2. Programmatically add a random path to your landscape and turn your landscape into an exploration game. Programmatically add a player controller that could navigate and explore the terrain. The player controller could be a primitive game object (such as a cube) procedurally generated. The game ends when the player quits the game. The player should start at any random position within the landscape.

Section B – Research

Task 3 - Research Question 1

Task 4.1 - [KU3.1] Describe how noise, Voronoi tessellation, midpoint displacement, splat maps and L-systems can be used to generate terrains. – 5 marks

In this task, you are requested to answer the following questions:

i. [Q1] Explain *noise* in terms of procedural content generation. **(1 mark)**

ii. [Q2] Explain *Perlin noise* and how it can be used for procedurally generating terrain. **(1 mark)**

iii. [Q3] Explain *Voronoi tessellation* and how it can be used for procedurally generating terrain. **(1 mark)**

iv. [Q4] Explain *midpoint displacement* and how it can be used for procedurally generating terrain. **(1 mark)**

v. [Q5] Explain *splat maps* and how they are used to add textures to terrain. **(1 mark)**

Task 4 - Research Question 2

Task 5.1 - [KU4.1] Describe different techniques for generating random levels – 5 marks

In this task, describe the process for procedurally generating dungeon levels. Your answer must include:

i. [Q1] Define what are dungeon levels, **(1 mark)**

ii. [Q2] State in which game genres dungeon levels are used and provide examples of existing games that use these types of levels, **(1 mark)**

iii. [Q3] Define what is required to construct dungeon models, **(1 mark)** and

iv. [Q4] Provide an overview of how dungeon models are procedurally generated. **(2 marks)**

Grading Scheme

Task	Assessment Criteria	Marks Awarded	Task Marks
1.1	• 5 Generated Buildings (1.5 marks each) – 7.5 marks • Generated Road Network – 2.5 marks		10
3.1	• 2 randomly generated cities (3 marks each) – 6 marks • Assign materials programmatically – 1 mark		7
3.2	• City Level generator – 4 marks • Car controller – 2 marks • Random start position of car controller – 2 marks • Level management (transitioning from one city to another) – 2 marks		10
2.1	• Landscape generator – 2 marks • Vegetation generator – 2 marks • Tree generator – 2 marks • Randomly locating vegetation and trees along the landscape – 1 mark		7
2.2	• 3 natural elements generator (2 marks each) – 6 marks • Randomly locating natural elements along the landscape – 1 mark		7
2.3	• Finishing touches – 4 marks • Random path generator – 3 marks • Exploration game – 1 mark • Player controller – 2 marks		10
4.1	• [Q1] – 1 mark • [Q2] – 1 mark		5

	• [Q3] – 1 mark • [Q4] – 1 mark • [Q5] – 1 mark		
5.1	• [Q1] – 1 mark • [Q2] – 1 mark • [Q3] – 1 mark • [Q4] – 2 marks		5
Total			61