University of Westminster
School of Computer Science & Engineering

6CCGD002W Advanced Maths and Game AI – Coursework (2021/22)	
Module leader	Markos Mentzelopoulos
Unit	Coursework – AI Implementation
Weighting:	60%
Qualifying mark	30%
Description	Demonstrate knowledge of advanced mathematics applicable to both games and software engineering using both written and programming evaluation approach Create a game environment that demonstrates a comprehension of a variety of game AI techniques for a proposed scenario, and a report justifying the selected methodology for the design solution and techniques used in the implementation.
Learning Outcomes Covered in this Assignment:	LO4 Critically evaluate design methodology related to Game AI and be able to select the methodology relevant to the given scenario. LO5 Be able to identify relevant techniques needed for the production of terrain and game intelligent agents and apply them to produce a reasonably complex terrain with realistic agent behaviour.
Handed Out:	Tuesday 8th February
Due Date	Wednesday 6th April by 1pm
Expected deliverables	Submit on Blackboard a zip file containing: **1) zip folder with development code elements, assets, executable and testing document** **2) Individual report describing your justification of the choices made your designs** **3)Video Link from YouTube with video walkthrough /explanation of your prototype and approach**
Method of Submission:	Submission online Blackboard
Type of Feedback and Due Date:	• Verbal feedback will be provided in the tutorials as the assessment progress, • Verbal feedback for the submitted coursework will be provided directly during the Demonstration • Written feedback and marks within 15 working days (3 weeks) after the submission deadline – Feedback will also be submitted through rubric during the Demonstration session. • Students that will miss the demonstration cannot obtain more than 35%

6CCGD002W Advanced Maths and Game AI – Coursework (2021/22)

	All marks will remain provisional until formally agreed by an Assessment Board.
BCS Criteria	2.1.1 Knowledge and understanding of facts, concepts, principles & theories 2.1.3 Problem solving strategies 2.1.8 Knowledge of management techniques to achieve objectives 4.1.2 Knowledge and understanding of mathematical principles 4.1.3 Knowledge and understanding of computational modelling 4.2.2 Defining problems, managing design process and evaluating outcomes

Assessment regulations

Refer to section 4 of the "How you study" guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:**http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**

Coursework Description

For this section you will need to develop a procedural generated environment following some requirement (see description below). The programming languages and game engine is free of choice. The arena can be tile-based, pixel-based or point-based (depending on the environment you will use to simulate the result).

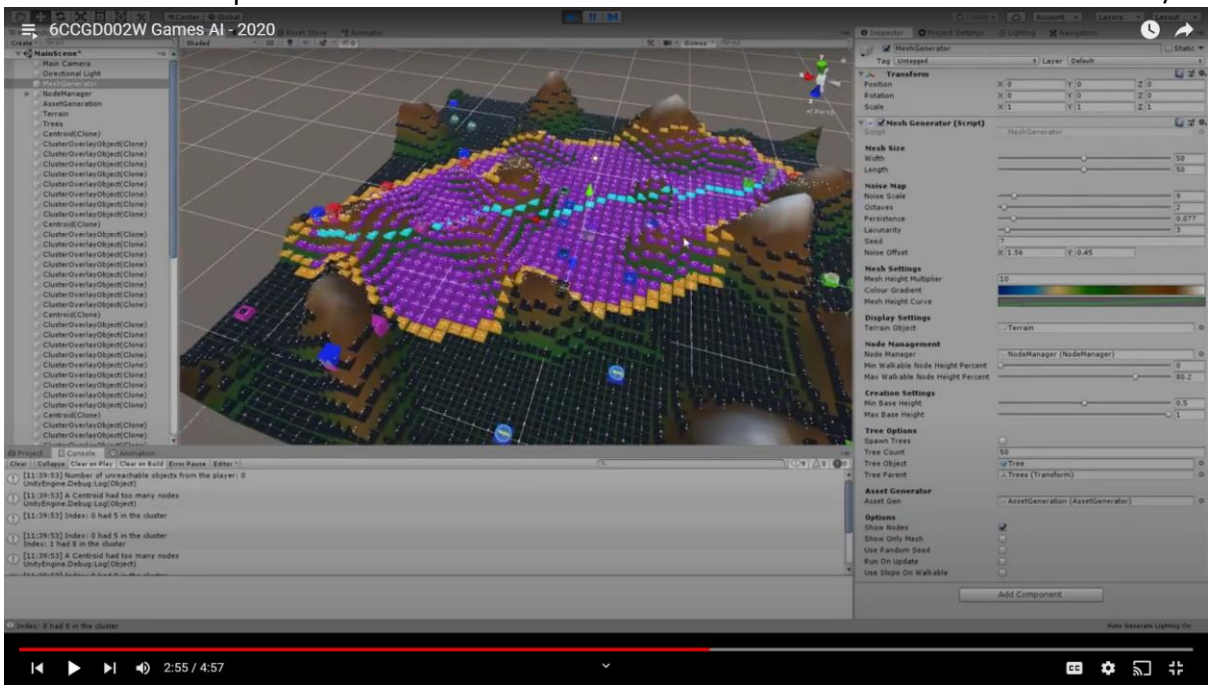
The terrain can be either 2D (top/down – depending on engine and language) or 3D.

[Visualisation appearance is not going to be marked based on graphical appearance rather than information representation]

1 – Main Terrain Implementation: 70 Marks

For the implementation you will need to incorporate features such as:

- Generating content: Based on your choice of environment set distinctive rules for asset incorporation breakdown – Use a set of 6 unique assets [e.g.: Health position, coin, weapon, magic spell, shield, trap etc.) which can be clustered under 4 categories. For example can be two for each category (pickables, traps, enemies, interactive), define set of rules for keeping track of the number of distinct triggered interactions e.g.: $\% \boldsymbol{Asset}_{\boldsymbol{Appearance}} = \frac{\boldsymbol{Artifact\ (Unique)}}{\boldsymbol{Total\ Number\ of\ assets}}$
- Use methodologies of either cellular, arbitrary shapes or noise (or hybrid) to generate your terrain (for noise you can use Perlin – extra marks will be provided for own implementation method – i.e., not Unity or UE4 integrated version)
- Add spatial filtering if needed for smoothing terrain
- Use path-finder techniques to check assets accessibility prior to generating them (add visualisation for success generation of each object for reaching between player and object location
- Use Clustering outcome and apply any technique technique to ensure smooth environment appearance and object generation optimisation – Your project should look for each cluster areas that incorporate assets from all categories with close distributions (at least 1 asset from each category) as well as been reachable from users.



Example of Terrain generation using Unity. Includes adaptive Perlin noise with smoothing option, clustering for 4 regions and pathfinder [cyan color]

2 - Project Report & Evaluation: 25 Marks

Submit a critical report of maximum 3000 words for your project implementation. The report must include critical evaluation of the choices you made for the terrain, the methodologies you used, and critical evaluation – accuracy and performance.

The project must be evaluated over a gradually added number of interactable assets (pickables, doors, enemies (stationary at this point) – procedurally generated.

Diagrams and tables can be added as part of Appendix.

3 - Project Video: 5 Marks

Finally create small 2-3 min video capture of your demo (Can use Panopto). Provide the URL of your deployed video on (YouTube.)

Coursework Marking scheme

The Coursework will be marked based on the following marking criteria:

Criteria	Mark per component	Mark provided	Comments
Section 1: Main Terrain Implementation	**70**		
Terrain generation using appropriate methodologies ***[If only Unity/UE readymade Perlin is used only 10 marks will be provided]***	20		
Applying correctly Clustering for correct asset distribution	10		
Clustering Visualization	10		
Applying Pathfinder for each asset to verify accessibility for player	10		
Pathfinder Visualization/asset (option selection)	10		
Advanced: Terrain & Asset Calibration methodologies	10		
Section 2: Project Report & Evaluation	**25**		
Critical analysis of approach used to develop your terrain and assets. Clearly describe your choices. [Include figures and experimentation stages]	15		
Critical Evaluation of project [Testing outcomes under different constraints [time/ time/ amount of clusters/number of objects] – can be presented with graphs.	10		
Section 3: Demo Video	5		
Total	100		

Marking Grade descriptors**: **For every section the following grade descriptor will be followed. For example, on Maximum number of 5 marks: Excellent =5, Good =4, Average =3, Basic =2, Insufficient =0 or 1

Excellent: demonstrates an outstanding understanding of topic principles (both in mathematical and implementation) with exceptional initiative, creativity, originality. The solution is well-structured and demonstrated. Documentation is in professional level and overall complexity in outstanding level – close to publicity level.

Good: demonstrates a good overall knowledge behind the topic. Covers fundamentals taught within the module and applied accurately to demonstrate a solution. Communicates ideas clearly and succinctly with good standard of presentation.

Average: demonstrates a production and documentation to a reasonable level of implementation. Minor errors exist and overall performance is close to demonstration. The mathematical solution follows correct approach with minor mistakes in both written and development of the algorithms. The report covers important topics of the implementation but lucks of justification and analysis.

Basic: demonstrates a basic understanding behind mathematical solutions and algorithmic interpretation to implement the project solution. Report is very basic without clear justification for the given approach followed towards the implementation of the solution and without any comparative results.

Insufficient: demonstrates a poor understanding behind mathematics and/or programming for generating a given solution. Report is insufficient and is missing critical components.