Kshan

NEA Survey Response

The student

Name	Kshan
School Email	jeyak027.209@student.foresthillschool.co.uk
Programming Level	2 / 10

Student's project

Description	Its a multiplayer fighting game, where the land is procedurally generated using Perlin noise
List of languages	C#
List of technologies	Unity now and blender later on
Experience using languages/technologies	Unity for like 2 month and c# for the same
Client	
Client's identity	me for now i will change later
Client fictional?	No

Student's Progress

Current section	Technical Implementation
List of completed sections	Analysis
Current page count	im not sure but like 5-8
Progress by section	
Analysis	75% < x < 100%

Design	0 < x ≤ 25%
Technical Implementation	25% < x ≤ 50%
Testing	Not started (0%)
Evaluation	Not started (0%)

Other

Implementation concerns	Multiplayer and even though i have done the procedural generator I follow a tutorial so Im a bit iffy on editing the code
Anything else? (Misc)	Ok for the procedural generator because of my lack of talent I had to follow this "https://www.youtube.com/playlist?list=PLFt_AvWsXl0eBW2EiBtl_sxmDtSgZBxB3" tutorial for creating the land, is this legal to do for my NEA as I am using his code for it, but I am adding other stuff like Multiplayer functionality and stuff so idk

Louis' Comments

General Comments	As I've noted about other projects, the choice to use C# and Unity is a little strange.[1] That said, Kshan's progress is not awful, even if the page count is a bit low. Though I have stumbled across the YouTube playlist mentioned above before, I am not familiar with the specifics regarding procedural terrain generation. I may have enough knowledge to have ideas bounced off me, but I don't have time to learn enough to be properly helpful. Kshan mentions that they plan to use Blender "later on". Unless they already have experience using Blender and/or are only planning to make simple models, I'm concerned that they could waste time creating models that won't earn them any marks.
Next steps	Given the page count is a bit low (especially seeing as Kshan believes the analysis section is nearly complete), it might be worth getting them to create a checklist of stuff included in the exemplars. I am concerned that Kshan seems to have moved on to technical implementation without completing much design work. This may mean that they don't have a full understanding of the complexity involved in the project or an idea of what

[1] Especially if the 2/10 programming ability rating is accurate.

	order they need to implement different components. I understand why they feel pressure to start implementing, given that the submission date is fast approaching. However, given how complex their project is, prematurely beginning implementation before having a good idea could lead to a - and I'm going a technical term here - bit of a sh*tshow. As part of Kshan's design work, he should ensure that his project is sufficiently modular. This is so he can prioritise certain functionality to ensure he can create a minimum viable project by the submission deadline.
Complexity	If completed, this project feels like it would fairly clearly be in the top complexity band.

See the next page for detailed complexity band information.

			Kshan
BOTTOM MARK BAND	**Algorithms**	Simple mathematical calculations	Must Have
		Linear search	Not Sure
	Databases	Non-SQL table access	Not Used
		Simple data structures	
MIDDLE MARK BAND	**Algorithms**	Simple scientific/mathematical /robotics/control/business model	Could Have
		Bubble Sort	Not Sure
		Binary search	Not Sure
		Simple user defined algorithms	Must Have
	Databases	Single table database	Not Used
		Simple data model in database	Not Used
		Writing and reading from files	Could Have
	File Access	Text files	Could Have
		File(s) organised for sequential access	Not Sure
	Web *Stuff*	Calling Web service APIs	Not Used
		Simple client-server model	Not Used
	Data Structures	Multi-dimensional arrays	
		Dictionaries	Should Have
		Records	Should Have
		Simple OOP model	Must Have
TOP MARK BAND	**Algorithms**	Complex scientific/mathematical/robotics/control/business model	Could Have
		Hashing	Not Sure
		Merge sort	Not Sure
		Advanced matrix operations	Not Used
		Recursive algorithms	Could Have
		Graph/Tree Traversal	Not Sure
		Complex user defined algorithms	Must Have
	Databases	Complex data model in database	Not Used
	File Access	Files(s) organised for direct access	Not Sure
	Web *Stuff*	Server-side scripting using request and response objects	Not Used
		Complex client-server model	Not Used
	Data Structures	Hash tables	Not Sure
		Lists	Should Have
		Stacks	Could Have
		Queues	Could Have
		Graphs	Could Have
		Trees	Could Have
		Complex OOP model	Must Have
		Linked lists	Could Have