

Mano

NEA Survey Response

The student

Name	Mano
School Email	kulcm003.209@student.foresthillschool.co.uk
Programming Level	4 / 10

Student's project

Description	The Hungarian card game Makaó which I intend to add multiplayer functionality and non-player opponents.
List of languages	C#
List of technologies	Unity
Experience using languages/technologies	Unity - around 2 months
Client	
Client's identity	My father
Client fictional?	No

Student's Progress

Current section	Technical Implementation
List of completed sections	Analysis
Current page count	Analysis - 3, Design - 6
Progress by section	
Analysis	75% < x < 100%

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

Other

Implementation concerns	Multiplayer using unity netcode and the non-player opponents
Anything else? (Misc)	

Louis' Comments

General Comments	<p>Given the self-assessed programming skill level and the time constraints of the NEA, Mano's decision to write their project in C# (using Unity) is one which I can only describe as a bold move.</p> <p>Despite this, Mano has made better progress than the majority of their peers. I do note, however, that Mano still has the majority of their design section to complete and has only three pages in their analysis section.</p>
Next steps	<p>I reckon it would be worth checking whether it is feasible for Mano to create the project in Unity using C#. If it is not, Mano should strongly consider pivoting to using Python as the window in which this would be possible is rapidly closing.</p> <p>I can't help with Unity stuff like multiplayer.</p>
Complexity	<p>This project could reach the top complexity band, however, it could be at risk of falling short (putting it in the middle band).</p> <p>While making Unity work is complex, I am concerned that much of the complexity in the final project may be handled by the game engine (and thus not be eligible for marks).</p>

See the next page for detailed complexity band information.

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