Task 35 Doc: Custom Project Plan

"You must create a short plan document that describes what you plan to do in your custom project, and in particular identifies what artefacts you will create, including documents, working code or other items such as videos, that demonstrate what you have achieved. This plan document will count towards your credit outcomes, even if you do not implement a custom project."

The purpose of asking you to create a custom project plan is to help you have a realistic goal while presenting what you need at the end for a good unit result. We strongly encourage you to talk to us as much as you need to refine your plan. For tutors - this is the fun stuff!

Key features we like to see in a custom project is a combination of AI techniques for a solving game-related problems or related applications.

Note:

- You do NOT need to build a full game! (DON'T – at least not to start with.)
- Doesn't need to be C++ and/or SDL (but why not and it's a safe domain to use).
- Can be from another unit / other work, but MUST be your work for recognition in this unit.

For most people, thinking of a game-related or pattern-related project is not a problem. Especially keep in mind that you do not need to create a full or complete working game. Demonstrating just a key part of a game system is enough for an excellent result.

Don't forget about supporting your code with documentation!

Note: A good code project that doesn't have good documentation will only line up with the Distinction result, which is a shame if the quality of the code is suitable for a HD result. Don't neglect documentation and supporting evidence in your custom project work! Not sure how to document it? Ask us!

What ideas can you put in the plan document for this task?

- If you are creating a game (even if just a part of one) create a game design document (GDD - mini version) or a game specification document. Remember that we are looking for evidence of AI application in the assessment. If you create a GDD do keep it simple. Typically, a GDD could include basic sketches of the interface, descriptions of the game entity interaction and game rules, how the game is played. References to existing games is a great way to clearly describe your design.
- From the design, you can create a feature list (extraction) - all the things you would need to code.
- For a list of features, you can do clustering of components, or a cost-benefit matrix, to figure out what is important to you to do first or what is interesting. (Your value system - it's up to you.)
- Present the architecture / design / approach (UML? Module/class/sequence? But only do what has value - don't just create it if it doesn't really show anything interesting or of value to you.)
- Implementation details (log, commit notes, release notes, testing notes?)
- Validation / evidence / outcome (particular if testing to see if your program works)
- Perhaps a post-mortem document for each "iteration" of work you do?

Again … it is up to you but talk to your tutor to check and for ideas!

Your custom project plan document needs to also be stored in your repository.

On the next page is a suggested plan document structure. You can of course vary this to suit your plan.

Suggested Project Plan Structure:

You do not need to follow this exactly. Create a document that you think best describes your plan.
Use the headings, replace the italic text with your own content.

Custom Project Plan
Your name and ID
COS30031 – Games Programming

Introduction
Explain the purpose of the document (not your code) … just a good general habit for writing.

My Project Description
Describe what you are interested in creating. It can be a good idea to explain what your interests or motivations are too. For example, "I have seen some examples of procedural terrain generation, and I've always wanted to try that myself. So, I would like to … ". Diagrams and figures are nice to have.

Include several paragraphs that outline what you want to do, and possibly the language/libraries/tools and approaches you think you will use to create your custom project. (Alternatively list these details later.) You can also put URLs to possibly relevant things – even if you don't' use them later. Capture the planning that you have already done and thought about.

If you are planning on creating a UI or (part of) a game, it's fine to include a sketch/photo of your design.

Next section below, you could have a single list of all the things you will create, or break it up into documents/code/at the interview. You could also make a rough plan for the order of things at the end.

What I Will Document
For documents, you might break that up into sections too for different stages. Common documents include: overall game design doc (basic UI, gameplay description, game rules, levels), requirement/features-lists (based on the design), architecture/design docs (UML, sketches etc), implementation/work iteration/stage plans, evidence/outcome documents, post-mortem document(s) typically done after different stages of a project (if you want), release notes, commit notes, readme files, etc.

What I Will Make
Describe the program/code/system. Try to clearly point out the techniques you will be using. If you haven't done so already include the language/framework/tools etc that you are planning (or might) use.

What I Will Present At The Interview
It's great to think about what you might be able to show at the interview. Remember that it is a show-and-tell exercise for the fun stuff you have achieved. Suggest a "Plan A" and a "Plan B" depending on how far you get.

Conclusion
Repeat the key points of your plan. Make sure it matches.