**[Part 1: Overview]**

***[Topic]***

*[An overview of what you plan to do in your project. Concentrate on the Big Picture and outcomes.]*

*[ - Two paragraphs]*

We plan to design a unique and fulfilling moral choice game. We hope to establish a standard of the genre which avoids the clichés and tropes which is the cause of a lot of similar games of the genre to fail to live up to their potential. Clichés such as good-bad dichotomies, linearity disguised as branching paths, and obvious ‘correct’ endings which enforce the artistic vision of the game designer rather than offering multiple conclusions depending on the choices made by the player throughout the game are some examples of frequent design errors when implementing a moral choice system which feels organic and immersive.

***[Motivation]***

*[What are the motivations for this project? Why is it interesting? How does it fit in with current IT trends? What would it show to a future employer if you were able to work on this project?]*

*[- One paragraph]*

***[Landscape]***

*[What similar systems or products are available? What competitors are there? What points of difference are there about your project compared to what exist now?]*

*[- One paragraph]*

**[Part 2: Detailed description]**

***[Aims]***

*[The topic description gives a general overview. However, it is usually helpful to have a specific aim for your project, as well as some smaller goals which will be helpful for achieving your aim. Describe these as best you can. Each project should have a single aim.]*

*[If things don't go as expected, this is the part of the plan that you would fall back on to answer questions such as “What are the most important parts of the project? Which parts should have priority over the others? If we have only enough time or resources for one of our goals, which one should it be?". One paragraph for the aim and one for each goal is expected. Each paragraph should include a description of the aim or goal, and a justification for it.]*

*[- Main aim (one paragraph)]*

*[- Goals (one paragraph per goal)]*

**[Part 3: Plans and Progress]**

*[Here you should give as much detail as you can about what your project will do, and how you will do it. This should also include how far you have got with developing any features or outcomes from your project. Tell us about the “story" of your project – how it began, how it has progressed, and what stage of the plan you are up to. Include any dead-ends you may have followed, decisions made, and changes that have been made to the project plan. This will need to include a significant amount of detail, so that it is easily seen what precisely you have done and are planning to do. If it helps, imagine the information that would be required if you were to hand this project over at the end of the semester to a new team to complete the job. What would you want to know, if you were one of the people taking over?]*

*[ 3 to 4 pages]*

**[Part 4: Technicalities]**

***[Roles]***

*[It is sometimes useful to define roles for particular participants, such as Lead Developer, or Technical Designer, or User Interface Designer. It is also possible that roles are changed from week to week, depending on what needs to be done next. Have you defined any specific roles for your project? If so, describe and justify these. If not, describe your process and justify why there are no specific roles.]*

*[ - Define roles for the projects (one paragraph - no minimum length)]*

***[Scope and Limits]***

*[One of the more difficult parts of project planning and execution is to define the scope and limits of the project. As mentioned above, you never really complete project like these; all you can ever do is your best in the time available. Part of that involves setting priorities and accepting that there will be features that will take too long to develop. This means that it is important to set a scope for your project, as a means of ensuring that you make the most of the time available.* ***For example, if you are developing a game, you might consider only producing one level and two or three characters, in order to show a proof-of-concept, rather than develop three levels and ten characters.]***

*[The scope is probably the most crucial part of your plan, and also the most difficult to define. One way to define the scope is to think of the deliverables for your project, i.e. what outcomes would you be able to show to someone who asks you to see the results of your work. This will also include several statements about what will not be part of the project. For example, if you are using Open Street Maps to show the location of all your favourite shops, the deliverables would include the updated map, but not the Open Street Maps technology itself. It would also not include many other features of Open Street Maps, or other interesting location -- just those which show your favourite shops.]*

*[Also, be aware of the phenomenon of `scope creep', which is the tendency for projects to incorporate more and more features. There is nothing wrong with being ambitious, but you only have a certain amount of time. At least one paragraph is expected]*

*[ - One paragraph]*

***[Tools and Technologies]***

*[*What software or other tools are required by the project? Are there any software licenses needed? Is there any hardware needed (beyond a standard laptop or something similar)? This needs to be precise (e.g. Windows Movie Maker Version 45.3) but needn't be long. You should also include a brief description of any prior experience any group members have had with the tools and technologies you list.*]*

*[ - One paragraph (no minimum but be concise)]*

***[Testing]***

*[How will your test your project? How will you know when you have succeeded? Testing is not something that you should leave until the very end; often it is far more useful to have a quick and dirty “mock up" of a project and then do some (limited) testing, to find out whether you are building the right product. If your project involves user testing, you should describe in your plan how you will find the test users, approximately what number of people you will need, and what background (if any) is required.]*

*[ - One paragraph]*

**[Part 5: Timeframe]**

***[Timeframe]***

*[Another difficult aspect of project planning is knowing how much time to allow. You will have something like 36 hours per person for this assignment. In order to develop a plan for further work beyond the end of this course, let us assume that you will have an extra 10 hours per week per person for 10 weeks in addition to this time in order to develop your project. This means that you will have six weeks (Weeks 7 to 12) of the semester to work on your assignment, with a further 10 weeks after that. This means that your plan will be for a total of 15 weeks, with the first 6 being on this assignment.]*

*[You will clearly not have the extra 10 weeks to work on the project; this is intended to give you a feeling for how much you would be able to achieve in that time. This means that the first 6 weeks of your timeline will end up being your actually progress on this project, with the remaining 10 weeks being your plan for the next stages.]*

[**This should be presented in the form of a table, with one row for each week, specifying as best you can the work for each person for each week**. *This means that the first six rows of the table will describe your progress so far, and the remaining 10 your best guess at how the remaining time would work.]*

*[This will no doubt change as you work on your assignment, as it will give you a more precise idea about how long it will take to get things done. This is not an unchangeable contract for exactly how things will work; that is unrealistic for just about any project. The idea is to get you thinking about how exactly your time should be allocated to the various tasks involved. It is a good idea to have a milestone (i.e. a specific outcome) for each week of the project. This may include getting familiar with tools, or reading up on a particular technique or technology. You should also include time for writing up the final report and any other documentation. Writing reports always takes longer than you think, especially as you should expect to re-write any piece of writing that you do at least three or four times.]*

*[ - Table format]*

**[Part 6]**

***[Risks]***

*[What risks can you identify for your project? There will always be some generic risks (such as computers breaking down the night before a deadline, health and family issues, and institutional changes). Do not include generic risks such as these. The idea is to be as specific as you can to your project. For example, if your topic is to develop a game, there may be a risk that the software you choose to work with may be very difficult to learn, poorly documented, or not turn out to have the features that it claims it has. These properties are often only discovered once you have started working with the software, and so unless you have had lots of experience with the particular tool, there is always a risk that it may not work as well as you believe it should, no matter how much prior research you do. Similar comments apply to hardware.]*

*[ - One paragraph]*

***[Group processes and communications]***

*[Communication between group members is arguably the most important aspect of your project. Past experience has shown that communication breakdowns between group members is the most common cause of project failures, so it is vital that you specify at the outset the means and expected frequency of communication between group members. How will your group communicate? How often will meetings take place? Will these be face-to-face, or using technologies such as Skype? Or Facebook? Or email? Or text? Or ... ?? What will you do if you have a group member who does not respond to communications? You should expect contact between group members at least twice a week. You can always make contact more often if you wish, but you do need to know what minimum  
frequency is expected from all members of your group.]*

*[ - One paragraph]*