**Project Plan (5%) – Due 8pm Monday Week 4**

Before the substantive development of the game can begin, each group will need to submit a project plan, the substantive part of which should not exceed two pages (you can have an appendix with diagrams). If there are any issues, you may be called in for an interview to discuss progress with the lecturer(s). It should include:

* Brief background about the developers (e.g. previous relevant experience)
* An outline of the system to be implemented

What you expect your product to be able to do, in particular, extra features beyond the minimum specifications (if necessary, the client may come back and request that you modify or remove some of your proposed features)

* Provisional schedule (when you expect to do things, interim milestones)
* Foreseen challenges and how you plan to overcome them
* Diagrams (e.g. class diagrams, sequence diagrams), sketches, screenshots, or other pictures that you can include in the appendix to show that you have thought about how the game might be structured (e.g. classes and interfaces) and also how the game might visually look (i.e. graphics).

Brief background:

Sylvain is in his third year at the University of Auckland, working towards his Computer Systems Engineering degree. He has been familiarised with object orientated programming, Eclipse and C++ as part of his COMPSYS202 paper which focused on these aspects. Sylvain worked on chatbot development as part of his practical work experience throughout last summer, actively coding in C# and following object orientated programming principles. He uses Git as a version control system quite actively, whether to maintain the websites he created or for organising code-related, university group projects. Sylvain also has a personal interest in game immersion and the factors which impact on and result in a better (or worse!) game immersion.

Mark is studying a conjoint Computer Systems Engineering and Computer Science degree at the University of Auckland. Earlier in the year he worked on a Microsoft Imagine Cup project which consists of a Windows Forms application and a Xamarin cross platform app, both coded in XAML and C#. His team’s extension features for the ELECTENG 209 project last year incorporated a real time graphing Android app, coded in Java. Both projects incorporated Git and object oriented programming. Last summer, Mark undertook a summer studentship for the University of Auckland’s Department of Electrical and Computer Engineering, supervised by Dr Nasser Giacaman. The project involved designing and creating a selection of educational videos, a website, and lab exercises to help future engineering students familiarise themselves with software such as Ubuntu, C++, and Eclipse.

Outline of the system to be implemented:

[(Brief?) Description of the game and what it's supposed to do]

Features:

[Basic and additional features we want to include]

Provisional schedule:

[When are we releasing what features]

Foreseen challenges:

Of the many foreseeable challenges which may be encountered, some of the major ones include the AI implementation in the game, closely following object orientated programming principles and the creation of an original storyline which is appropriate to the game. AI implementation may get complicated if we want it to be up to a certain standard of quality, and not a basic AI following the ball moving on the screen. Following object orientated programming principles is a actually more complex than it seems for students. This is because basic programming papers encourage learning the language and the syntax, rarely focusing on code efficiency and programming principles such as object orientated principles. As a result, students have very few opportunities to develop the ability of following these coding principles, which explains why it may be one of the major challenges which will be encountered. Finally, the creation of an original storyline relevant to the game may prove to be an issue. Between the fact that the game is very specific when it comes to the player number and game mechanics and the fact that the environment never really changes, the story which has to encompass all these details will be complicated to develop. This is especially true if the story is to be original and captivating to the player, while not reusing too many clichés or the storyline from the original game.

Diagrams and stuff (technically the appendix):

[All the stuff we drew on paper and such] 🡨 Scan and upload or redraw in PP? Redraw + Scan