**IMAT3451 Project Contract Template**

**Student Name** Connor Rowland

**P-number** P16204361

**Programme** Computer Games Programming

**Email address** P16204361@my365.dmu.ac.uk

**Project Title** Dobble

**Project Proposer**

Simon Coupland, Department of Computer Technology (Games/CompSci/SoftEng), simonc@dmu.ac.uk

**Supervisor**

Simon Coupland, lecturer.

**Introduction**

A real-time, multiplayer, simulated card game.

**Project Background**

The project arises from my personal interests of gaming, networking, real-time simulation and mathematics, and wanting to create something that complements each of these interests and that many people can play together, across the “internet”, instead of having to be in the same room to play; like you’d have to be for the original version of the game. I like the idea of being able to share the experience with anyone, and not just people you know, which is another reason why I believe the game must have a multiplayer component, to allow individuals to play with anyone, and meet new people online.

**Aim/Objectives/Deliverables**

**Aim:**

I aim to build a real-time, multiplayer, simulated card game that a couple of people can play together, over the “internet”.

**Objectives:**

* Create clear and concise design documentation, for my project.
* Create and engineer a suitable test plan.
* Research the mathematics behind the card game Dobble.
* Research how to implement the multiplayer side of the game.
* Design and implement an elegant interface.
* Implement a network based real-time simulation game.
* The project will include an in-depth investigation of the context and literature, and where appropriate, other similar products.
* Ensure the final report contains a clear description of the steps taken, in the projects life cycle.
* The final report will contain a description of the use of tools to support the development process.
* Ensure the report includes a description of the tools used in the project.
* Ensure the report contains a critical appraisal of the project. This will include the rationale for any implementation/design decisions made, any lessons learned through the development of the project, and an evaluation of the project outcome and the operation of the creation of the project (including a review of the plan and any deviations from it).
* Ensure all sources used are completely referenced.

**Deliverables:**

|  |  |
| --- | --- |
| **First Submission**  Week 4 | * Project contract * Ethics form * Project plan (Burndown chart/Gannt chart) * Global checklist * BCS checklist |
| **Second Submission**  Week 10 | * Literature review * Prototype (of my current progress) |
| **Final Submission**  Week 27 | * Design documentation (dOxygen) * Test plan * Use Case Diagrams/Use Case Descriptions/Class diagrams/ER model/State transition diagrams * Software (The Dobble game) * Final report, including my critical evaluation * Appendices |
| **Viva Examination**  Week 31-33 | * Oral examination (demo of my work) |

**Resources and Constraints**

**Resources:**

There will be no issues with resources because I have the following: Visual Studio 2017, GitHub, GitHub’s Kanban board, Internet connectivity, OpenGL graphics library, a computer system, and Windows 10.

**Constraints:**

There might be a lack of time due to there being a bigger learning curve than anticipated, when dealing with certain aspects of the project.

**Sources of Information**

Here’s a list of sources I tend to use:

* Game Programming Patterns (<http://gameprogrammingpatterns.com/>).
* Game Engine Architecture (<https://www.gameenginebook.com/>).
* C++ Primer (<https://www.amazon.co.uk/C-Primer-Stanley-B-Lippman/dp/0321714113>).
* Computer Networking - A Top-Down Approach (<https://www.amazon.co.uk/Computer-Networking-Top-Down-James-Kurose/dp/0133594149/ref=sr_1_1?s=books&ie=UTF8&qid=1539879045&sr=1-1&keywords=9780133594140>).
* Software and API documentation.
  + SFML (<https://www.sfml-dev.org/documentation/2.5.0/>).
  + OpenGL (<https://www.opengl.org/documentation/>).
  + GitHub (<https://guides.github.com/>).
  + Visual Studio (<https://docs.microsoft.com/en-us/visualstudio/?view=vs-2017>).
* The Web.
  + Gefferongames (<https://gafferongames.com/post/udp_vs_tcp/>).
  + Wikipedia (<https://en.wikipedia.org/wiki/Real-time_simulation>).
  + Dobble (<https://www.asmodee.co.uk/featured-product/dobble/>).
  + Stackexchange (<https://math.stackexchange.com/questions/464932/dobble-card-game-mathematical-background>).

**Risk Analysis**

I could run out of time. If that happens then I will prioritise the application-critical tasks, to get the most important features implemented before the inessential ones. And I’ll ask for an extended deadline. Or I’ll revise the deliverables.

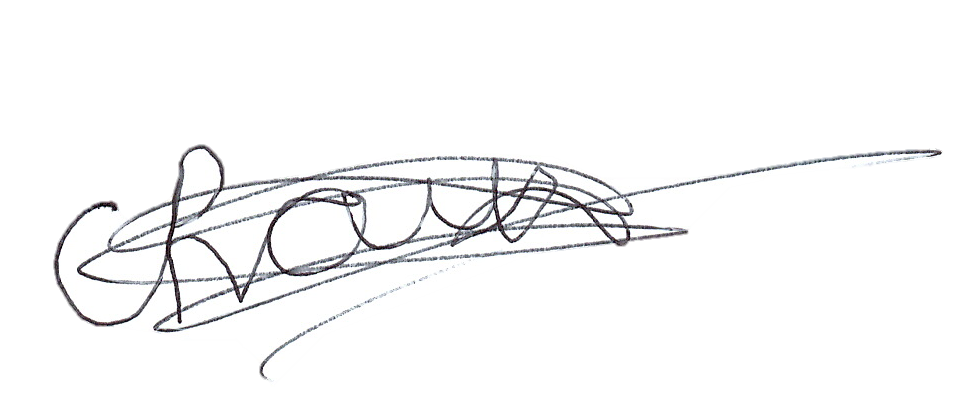
I might become ill, which will cause development to slow down, or come to a halt. I will plan and prepare by creating gaps in my sprints I can use to catch back up, if necessary, or I can plan to finish the project development a sprite early, giving me a whole sprite to catch up with anything I’ve missed or go over the project checking everything.

The main project I’m working on could become corrupt (damaged hardware). I can use a source control program to manage my backup[s], and if anything becomes corrupt I will have a backup and can carry on from that point.

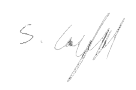
There’s a possibility that I underestimate a task, and it takes me a lot longer than I initially thought. To minimise this risk, I’ll use a risk level for each object, which indicates how risky the objective is – based on my confidence of the task.

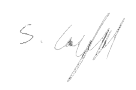
**Schedule of Activities**

I will create a Burndown chart and a Gantt chart to assist me with keeping on track and staying organised, helping me maintain a strict schedule.



**Student:**  **Date:** 16/10/2018

****

**Proposer Date:** 24/10/2018

**Supervisor\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:** 24/10/2018