Department of Computing and Networking
Games Degree Year 3

Project Guidelines

Version 1.0

September 2013

Table of Contents

Introduction

You are required to develop your project in C++ using the 'Ogre' engine. The project is divided into three sprints as indicated in the marking scheme & schedule on the next page.

Your project can be either a game or a game development tool as described below

Game Project

You are free to choose the genre and style of the game, but it must be in 3D and have a fixed camera position. Your project is expected to incorporate at <u>least one</u> of the following three elements unless you have agreed otherwise with your tutor.

- **Physics system:** develop your own routines for collision detection/response and any other requirements.
- **AI system:** develop a system for controlling agent logic and/or a movement system.
- **Turn-based system:** develop a system to manage turns on *n*-player games on a single machine.

Additionally, your game project <u>must</u> incorporate the following:

- **GUI:** use widgets from any compatible GUI library to build a professional looking interface.
- **Audio:** use any compatible audio library for sound effects and any other audio requirements.

Game Development Tool

The project can be a game development tool (GDT). A GDT is a specialist piece of software which aids in the building of a game.

Examples of GDTs are

- Level editor
- Resource file convertor
- Procedural asset generation utility
- Material creation tool

For a GDT project it is required that your has the following elements;

- interactive 3D visualisation (using OGRE) of the task(s) being carried out
- sophisticated graphical user interface
- output is in a format which is usable in a game development context

Component Marks & Schedule

You may not submit a manual or other deliverable before its predecessor has been accepted by your tutor.

	Component	Marks	Due
Sprint One (16/9 to 25/10)	Project Proposal	5	20th Sep
	Proof of concept demo	10	25th Oct
	Iteration Report #1	5	25th Oct
Sprint Two (26/10 to 6/12)	Project Design Document (sprint 2)	5	1st Nov
	Iteration Report #2	5	6th Dec
	Sprint 2 Demo & Presentation (week beginning 10th Dec)	15	TBA
Sprint Three (9/12 to 14/2)	Project Design Document (sprint 3)	5	12th Dec
	Iteration Report #3	5	14th Feb
	Technical Manual	5	14th Feb
	Final Demo (Week beginning 17th Feb)	30	TBA
	Project Management	10	

General Guidelines

These guidelines are guidelines. However, departing from them is not advisable without the prior consent of your project tutor

Formatting

All reports submitted must be in MS Word format and be printed and bound. They must have a title page with project name, document name, date, student name(s) and student userid(s). The IT logo should appear in the top, right-hand corner of the title page. Use the front page of this document as a template.

Each page (other than the title page) of each document should have a header with the project title and document name and a footer with student name and userid. The footer should also have page number and page count (e.g. 3/15)

Any document more than 3 pages long should have a table of contents and should be presented suitably bound, with a transparent sheet over the front cover.

Document text should normally be font size 10 or 12.

Plagiarism

Plagiarism is not permitted. Material from the work of others (whether code, text, diagram or other) should be fully acknowledged. Material based on the work of others (research, for example, phrased in your own words) should include a bibliography in the standard format (see research). The Institute's official declaration on work origin must be included in your project report, and signed in any printed copy of the report.

Copyright

Project ownership and copyright belongs to the Institute. You should put no indication of copyright, assumed or otherwise, on any of your project documentation, except where you are including material, which is the copyright of others. Note that, in the event of you wishing to further develop the project for commercial or personal use, the Institute may consider granting you a licence to do so.

Project Management

You will work as part of a team, but the management of your project will be assessed individually and continuously throughout the academic year:

- Using Jira to plan and track your work in each session (you are expected to have at least one work session per week).
- Committing to github after each session
- Presenting your progress in weekly meetings

You must be present for a project meeting in order to receive credit.

Sprint planning

You are required to write user stories and possibly developer stories as part of the sprint planning process (see Appendix D, section 3 User and Developer Stories). Stories must decomposed into subtasks (see below) and both must be recorded in Jira. With your supervisor, prioritise stories which are then frozen for the sprint.

Subtasks

In the task list, identify all the tasks that are needed to complete this sprint of the project. This list should include all work required, e.g. research, report writing, implementing features, testing & debugging, creating assets, etc.

Estimate how long each task should take to complete.

Break large tasks into smaller ones (task should normally be between 1 and 5 hours in duration).

Assign each task to ONE team member; making sure that the work is fairly distributed. If you find it absolutely necessary for two team members to work on a task together, that information can be recorded in the Jira.

Sprint One – Outline and Deliverables

The project is to be developed over three sprints or iterations.

During the first sprint, you will work towards producing a proof of concept demo to test the feasibility of gameplay or functionality ideas you have planned for your project.

You have to produce a number of documents during each sprint.

Sprint One Deliverables

1. Project proposal document (see appendix A or B).
2. Proof of concept demo: You will demonstrate to the panel of project supervisors a working prototype of the gameplay you have planned for your project.
3. Iteration Report #1(see appendix C)

Sprint Two - Outline and Deliverables

With your initial proof of concept completed, you should now have a clearer picture of the features you will be able to implement in the available time. The work you have planned will be completed over the next two sprints of the project. You will use the project plan to create a new task list in the project management document for this sprint.

Sprint Two Deliverables

1. Project design document (see appendix D or E).
2. Iteration Report #2 (see appendix C)
3. Sprint 2 Demo & Presentation

Should be prepared in PowerPoint (or similar) and delivered as a short talk to your peers. Allow time for questions at the end, but don't allow interruptions. The demo should be incorporated into your presentation as an embedded video that showcases the main gameplay elements of your project.

Sprint Three - Outline and Deliverables

At this stage you should have a basic working game with. The final sprint will see you adding more features, improving gameplay and preparing the project for final demonstration.

You will use the project plan to create a new task list in the project management document for this sprint.

Sprint Three Deliverables

1. A game design document – present the document from sprint 2, but highlight any changes or additions for sprint three.
2. Iteration Report #3 (see appendix C)
3. Technical Manual
    a. Institute Declaration on Originality (see Appendix F)
    b. A UML class diagram illustrating the relationship between classes in your project.
    c. Class Documentation (this should be the formatted output from Doxygen). Every class and public member function must have a summary description. Method parameters and return values must be properly documented.
    d. Detailed references to tools or resources used;
    e. Detailed references to 3rd party code used;
4. Project Software

    Your project code should be delivered on CD/DVD or flash media (non-returnable) as appropriate. It should generally be auto-installing. There should be a separate directory on the disk called "Docs" which should include copies of all documentation submitted during the year, presentation PowerPoints, source code etc. If non-standard additional software (e.g. components for compilers, tool kits etc) are required to build your project, copies should be included in an appropriate directory.
5. Final Demonstration

    Sometime after the submission date of the project materials you will demonstrate your project to the panel of tutors. The demonstration will normally commence with a short outline of the goals and purpose of your project, as most of the panel will not be familiar with it. Part of the demonstration will normally include installation of your software on the target platform from the CD (or other media) you have submitted. By prior arrangement with your tutor you may be permitted to work with pre-installed materials. Following your demonstration, during which you will not be questioned (except for requests for demonstration of particular features, clarifications etc) you will be questioned on your project by the panel.

Appendix A: Game proposal document

The game proposal document is intended to be a short document (no more than 3 pages) which describes the overall vision for your game. It should have the following sections:

What:

- What is this game? (the four sentences that explains the main idea of the game)
- What is the overall scenario?
- What is the story? (if applicable)

Why:

- Why should someone play this game?
- What is it that makes this game unique?
- What principles is the game exploring?
- What is it that drives the game forward?

Where:

- Where will the game take place, what is the game environment?

Who:

- Who are the main players, which characters do they represent?
- Who are the target groups of this game?

How:

- How to play?
- How to win, gain?
- How to lose?
- How to die, be hurt?

Functionality planned for proof of concept demo

List and describe the core features

Appendix B: Game development tool proposal document

The game proposal document is intended to be a short document (no more than 3 pages) which describes the overall vision for your GDT. It should have the following sections:

What

- What is this tool (four sentences that explain the main purpose of the tool)?
- What tasks is it used for?
- What genre of game development will it be applied to?
- What will be produced by the product?

Why:

- Why would someone use this as opposed to some other tool/approach?

Who:

- Who will use this product. What role do they have in the game development process.
- Is it targeted to typical end-users, modders, amateur developers, specialists etc.
- What skills and knowledge will the user be expected to have

How:

- How will the user interact with the tool.
- How will the end product be used elsewhere in game development

Functionality planned for proof of concept demo

- List and describe the core features
- Write one or two user stories outlining the typical workflow using the GDT

Appendix C: Iteration Report

Table of contents.

Work completed

- Stories and associated subtasks implemented (if only partially implemented, discuss here)
- Stories and associated subtasks not implemented.
- Additional work completed.
- Work breakdown between team members. Use screenshots from Jira to illustrate.

Reflection

- What you learned.
- What you would do differently if starting again.

Appendix D: Game Design Document Template

1. Functionality planned for current sprint.

List the stories and associated subtasks planned for the current project sprint (clear screenshots from Jira will suffice).

2. Game mechanics

The next section is a detailed description of the game mechanics which are the procedures and rules of the game.

2.1 Space

Defines the various places that can exist in a game and how those places are related to one another. Spaces can be continuous or discrete.

- What are the boundaries of the space?
- How many dimensions does it have?
- Are there sub-spaces? how are they connected?
- Is this world better than the real world?
- Is this world simpler that the real world?

2.2 Objects (all elements that the player can engage with)

Objects, Attributes and States - a space has objects (car) in it, objects that have attributes (maximum and current speed), each attributes having a current state (200km/h, 90km/h).

- What are the objects in the game?
- What are the attributes of the objects?
- What are the possible states for each attribute? What triggers the state changes for each attribute?
- What states are known by all players?
- What information does the player need that isn't obvious just by looking at the game world?
- When does the player need this information?
- How can this information be delivered to the player, so it doesn't interfere with player's interactions?

Game objects usually have many attributes and states, so it is often useful to construct a state diagram for each attribute to make sure you understand which states are connected to which, and what triggers state changes.

2.3 Actions

Actions - the "verbs" (protect, build, move, jump, shot, avoid) of game mechanics, representing the base actions a player can take.

Some verbs may act on multiple objects;
Goals can be achieved more than one way;
What does it mean to make progress in the game?
What are the operative actions? What are the resultant actions?

2.4 Rules

Rules - define the space, the objects, the actions, the consequences of the actions, the constraints of the actions and the goals.
What is the problem the game asks the player to solve?
Are the rules fair enough for the player to continue to solve the problem? (Inconsistent game rules may unfairly impede the player's progress).
Do the players feel in control and powerful? (feeling in control is one of the characteristics of *flow*, see: http://jenovachen.com/flowingames/foundation.htm#flow)
What are the rewards and risks this game delivers?

2.5 Skills

Skills - every game requires players to exercise certain skills (physical, mental, social skills).

What skills does the game require from the player?
Are the required skills real or virtual?

2.6 Chance

Chance - is essential part of a fun game because chance means uncertainty, and uncertainty means surprises. Chance requires interactions between all the other mechanics.

What in the game is truly random?
Do players have the opportunity to take interesting risks in the game?
What is the relationship between chance and skill in the game?

3. User and Developer Stories

The details of all user and developer stories should be contained in this document which will evolve throughout the project.

The description of the user stories with storyboards should contain enough detail that if you gave the description to another developer, he/she could develop the feature without any further input from you.

Each user story must be presented with a title, description, conditions of satisfaction. It must also have an associated storyboard (developer stories may not need a storyboard). Sample stories are presented below.

Sample User Story

This story is taken from a multiplayer card matching game…

Title: Positive reinforcement
I want to receive encouragement during play so that I am motivated to continue playing.

Conditions of satisfaction

- I occasionally see positive reinforcement messages when I do not match a pair of cards.
- I occasionally see positive reinforcement messages when I successfully match a pair of cards.

Storyboard:

1) The current player fails to match a pair of cards and sees a positive reinforcement message in the form of a short-lived notification:

2) The current player matches a pair of cards and sees a positive reinforcement message in the form of a short-lived notification:

Sample Developer Story

This story is taken from a browser-based multiplayer card matching game…

Title: Correctly formed client URL
As a server developer, I want the client browser URL to contain a valid game instance identifier and an access token so I can determine what players are participating in this game.

Conditions of satisfaction

- The client URL will be of the form
  `http://memory_game_server_address/?giid=<gameInstanceIdentifier>&access_token=<access token>`

4. User Interface

User interfaces should be specified loosely here (possibly screen shots of prototype user interfaces - you will not be held to using these UIs they are just to assist in explanation of your project functionality)

Appendix E: Game Development Tool Design Document Template

1. Functionality planned for current sprint.

List and describe the core features planned for the current project sprint.

2. Tool mechanics

The next section is a detailed description of the mechanics which are the procedures and rules of the GDT.

2.1 Views

Defines the various ways that the user can visualise and interact with the data.
What views are available.
Are the views spatial or non-spatial?
For spatial data, what projections are used, what camera control is available?

2.2 Objects (all elements that the user can engage with)

Objects, Attributes and States - a space has objects (car) in it, objects that have attributes (maximum and current speed), each attributes having a current state (200km/h, 90km/h).
What are the objects in the tool?
What are the attributes of the objects?
What are the possible states for each attribute? What triggers the state changes for each attribute?
When does the user need this information?
How can this information be delivered to the player, so it doesn't interfere with player's interactions?
Tool objects usually have many attributes and states, so it is often useful to construct a state diagram for each attribute to make sure you understand which states are connected to which, and what triggers state changes.

2.3 Actions

Actions - the "verbs" (select, build, move, modify, interrogate, etc) of the mechanics, representing the base actions a player can take.
Some verbs may act on multiple objects;
Goals can be achieved more than one way;
What are the operative actions? What are the resultant actions?
What are the consequences of the actions, the constraints of the actions and the goals.
Feedback, how will the user know of the result of an action

3. User and Developer Stories

(See Section 3, Appendix D).

4. User Interface

4.1 Controls

What controls will be available to the user and how do they correspond to the allowable actions?
How will the controls change depending on context
A mock-up of the user interface should be supplied

4.2 Editing modes

Can the user enter different editing modes? How does the interface change

5. Data Design

5.1 Data formats & sources

Detail the format of any data which will be input or output from the tool.

Where will the data come from/be sent (file, stream, internet service...)

5.2 Data structures

Describe any internal data structures to be used.

Appendix F: Declaration on Originality

DECLARATION

I declare that the work I am submitting for assessing by the Institute examiner(s) is entirely my own work, save for any portion(s) thereof where the author or source has been duly referenced and attributed.

I further state that I have read, and am familiar with, the Institute Examination and Assessment Regulations and that I am not, so far as I am aware, in breach of any of these regulations.

Student Name:…….Student ID Number:……

Course of Study:……..

SIGNED:_____DATE:_____