G54GAM – Coursework 2 & 3

Summary

You are required to design and prototype a computer game. This coursework consists of two parts – describing and documenting the design of your game (coursework 2) and developing a working prototype (coursework 3). Your documented design and prototype game should submitted no later than:

- **4pm on Friday 18^{h} May 2012**

Submissions should be made electronically via WebCT (http://webct.nottingham.ac.uk). Standard penalties of 5% per working day will be applied to late submissions.

Coursework 2 – documentation – is worth 30% of the mark for this module
Coursework 3 – prototype – is worth 40% of the mark for this module

Assessment Criteria

The coursework will be assessed against the standard criteria:
https://workspace.nottingham.ac.uk/display/CompSci/Marking+Criteria

Demonstrating knowledge of the area

Quality of the concept, including appropriateness and novelty

Quality of the technological design, including appropriate use of game design concepts, and appropriate good coding practice (abstraction, commenting, naming)

Quality of the realization, including how well it works and elaborations over and above the basic requirements

Including all of the above aspects, clarity of structure, quality of argument / evidence, and insight / novelty

N.B. the following are NOT assessed unless specifically demonstrating a relevant concept:

- **Quality or quantity of media assets (sprites, sounds, images etc)**
- **Quantity of levels / quantity of similar game objects / size of application**

Hints and Tips

You should design your game considering **one or more, but not all** of the following aims (feel free to explore areas outside of this list!)

- Creating a series of well-defined challenges
- Demonstrating the use of a difficulty curve
- Demonstrating balanced relationships between game objects
- Demonstrating a novel premise
- Demonstrating the use of narrative
- Proving the technical feasibility of a novel design
- Provoking a certain aesthetic (horror? thrill?)

Your game should have a beginning and an end that allows the player to play towards a clearly defined objective, comprising of representative challenges, for example, exploration, pattern recognition, knowledge etc. The game should be appropriately balanced, and demonstrate an appreciation of the concept of progression and increasing difficulty and challenge.

The emphasis of this coursework is about thinking about structured game play, and demonstrating your understanding of the course material, so try not to get too bogged down designing assets and sprites - you'll get more marks for a simple looking game that has a well thought out game design than for an aesthetically beautiful game that is shallow.

Begin by brainstorming a number of different game ideas, and use the concepts discussed in the lectures on game design to develop one that you think will be interesting, enjoyable and realistic in the time frame.

If you are stuck for inspiration, then feel free to use the simple games that we made in the lab sessions as a **starting point**. Coming up with a unique form of game play is very difficult, even for the professionals, so you may also draw inspiration from existing games. However, you should make sure that your game is suitably unique, whether this is in terms of core mechanics, premise or challenges.

NB – games that are blatant derivations or clones of the Game Maker tutorials are not acceptable, and will be heavily penalized. Similarly, it is not acceptable to submit just one of the lab session exercises. You are being assessed on your understanding of the course content, and you should specifically attempt to introduce particular challenges, difficulties and balanced relationships into your game that demonstrate this.

Start by selecting a **game genre**. Here are a few examples that we have made in the lab sessions, or could easily be made by adapting the core mechanics of the prototypes that we have made:

- Platformer – Jumping the ball between platforms
- Action – Shooter - Scrolling shoot-em-up

- Action – Shooter – Multi-player tanks
- Racing – Could the tank game be adapted into a Micro-machines style racing game?

Next think of a **premise** for your game. What is the story behind the game? Defending the earth from aliens or ghosts? Racing a tank or a car? What is the core mechanic?

Finally, consider the **formal elements** of game play. What are the goals and objectives, rules, resources, conflict and challenges, amongst others?

Coursework 2 - Documentation

Your game documentation (coursework 2) should consist of three sections, as follows. There is no word limit, although as a guide you should put a similar amount of effort into it as you would a 2500 word essay!

The following sections give an overview as to what you could consider putting in each section of your documentation; these are just examples that you are not required to adhere to, although your sections should have the same overall aim.

Concept or Pitch

Necessarily brief, simple and to-the-point. 1 page that is used to sell your game concept – imagine that you are trying to get the reader to invest money in your game.

- Introduction
    - The most important part of the document that sets this game apart from other games. Include the title, genre, premise and snappy overview in a couple of sentences
- Description
    - Describe the game to the reader as if they are the player; an exciting narrative of the player's experience. Include the key elements that define the core game play, describing what the player sees and does.
- Key features
    - A list of items that set this game apart from others (e.g. "hoards of AI controlled enemies")
- Genre
    - Define the game genre, core mechanic and style, using existing classifications (e.g. "a futuristic fire-person shooter")
- Concept art
    - A screenshot or mock-up helps to sell the idea.

Design and Functional Specification

The features and functions of the game – what goes into the game and what it does. This is not a technical specification, but it is a "bible" that could be given to a developer who could then use it to produce a technical specification or a working prototype. Diagrams and bullet points should be used rather than flowing prose to make the specification clear and readable.

- Game Mechanics – the game play in detailed terms
    - Core game play
        - Objects, rules, procedures and resources
    - Game flow
        - The progression of challenge, balance and difficulty
    - Characters
        - Controlled by the players or AI
    - Game play elements
        - Weapons, switches, traps, items, power-ups
    - Physics and statistics
        - Movement and speed, collisions and reactions
    - AI (if applicable)
        - Desired behaviour
    - Multiplayer (if applicable)
        - One on one, cooperative

- User Interface
    - Flow chart
        - Navigation through the screens and windows, the HUD
    - Functional requirements
        - How the player interacts with the user interface

- Level Requirements
    - Level relationship diagram
        - Different levels or rooms in the game and how they are linked (linear, branching, nested)
    - Asset revelation schedule
        - When the game's assets are revealed to the player (bosses and enemy types, weapons, power-ups, objectives, challenges)
    - Target difficulty level
        - Easy, medium then difficult

Prototype Walkthrough

A set of instructions or a transcript describing one way of successfully playing your prototype.

NB – this is not just a restatement of the game design, instead describe the player's experience in a way that could be used for a walkthrough or FAQ for the game.

Here is an example of how the walkthrough for a simple shoot-em-up might be structured:

- Intro
    - Press “start game” on the intro screen to begin
- Setting
    - Fly your ship through a scrolling landscape of floating islands
    - Steer left and right using the arrow keys, and use the space bar to fire a bullet
- Challenge 1
    - Two small red enemies appear and fly towards you, shooting. Move left and right to avoid the bullets, while shooting to try and hit the enemies
    - Shoot the two enemies approximately ten times each until they are destroyed. One drops a power-up, which you can collect, giving you a weapon that fires faster bullets

…

- End game
    - The boss enemy moves in a circle above you. Hide on the left hand side to avoid its bullets, then quickly dart out to shoot it. Repeat until it is destroyed.
    - Enter your name into the scoreboard with the new highest score

Coursework 3 – Prototype Implementation

The prototype should be a snapshot of the game, as if a slice were lifted from a complete game. If your game is a platform game, then create one or two levels and settings and illustrate them in the documentation. If it is a racing or fighting game, then select two or three vehicles or characters and settings to implement to demonstrate the game play and game mechanics. Focus on a small game that demonstrates the mechanics, progression and balance of the game play.

Your prototype should be built using Game Maker, unless you have discussed it with me first. You should submit the following:

- A compiled executable file (.exe)
- Your Game Maker project file (.gmk or other, depending on version)
- Any associated resource files
- Any source code if using a different platform / language

NB – You are required to submit a prototype of your game, not a complete game. Again, remember that you are being assessed on your understanding of the course content. It is better to submit a prototype that demonstrates a few points well, than attempt to construct a full game. No additional marks will be given for large games that do not demonstrate specific game design elements (e.g. multiple similar levels that do not change in difficulty)

While the emphasis of this coursework is on implementing a prototype of your game design, you should also consider the technical implementation of your prototype:

- Clearly label your objects and resources
- Objects, events and actions should be well thought out and arranged in appropriate hierarchies

    - Make use of object orientation, parents and children

    - Think about which object is responsible for which functionality

    - Use controller objects and finite state machines as appropriate

We've only used a subset of the events and actions available in Game Maker so far in the lab sessions. If you get stuck, or don't know how to implement a particular feature, then the manual is a good source of information:

- http://www.yoyogames.com/downloads/docs/gmaker80.pdf

NB – Use of third party assets (sprites, images, graphics, sounds etc) is acceptable, however these MUST be credited or referenced. Failure to do so amounts to plagiarism and will be penalized as such – e.g. potential failure of coursework, module or even degree. If you have any doubts, then ask me. I cannot stress this strongly enough – do not submit work that is not your own.