G51FSE COURSEWORK 2014

Dr. Julie Greensmith
University of Nottingham
11/2/14

Coursework Overview

Your Task: You are tasked to find a partner and to create a 2D Arcade game. You will have a choice over what game you wish to make and a choice of implementation language. Through this process, in your pairs you are to develop the game from its initial specification, through design, protytping, refinement, implementation, testing, and evaluation stages. You are required to deliver a demo for a prototype, a completed game and a portfolio. This document outlines specifically what is expected of you in this coursework. This coursework is worth 30% of the total marks for this module.

Part A: Prototype demo (15%) - Due Date: Lab 6 (7/3/14)
Part B: Full Game and Portfolio (80%) - Due Date: **Wed 14th May, 4pm**
Part D: Attendance (5%) - assessed throughout the course via the lab register

Coursework Synopsis:

In this coursework we aim to emulate an agile development team working for a small game development company, called NG2 Games. You are tasked to find a partner and to develop the concept for your game and to produce it in a professional manner. You are required to produce an inventive 2D game, including a prototype and larger full version and all supporting documentation in the form of a development portfolio. NG2 Games prides itself on developing inventive and imaginative games, but focuses on quick software delivery.

Coursework Learning Objectives:

- To perform the process of software engineering from start to completion, demonstrating evidence of ability and aptitude for each stage of the process.
- To be able to program and develop effectively in pairs.
- To demonstrate original thought and creativity in software production.
- To show use of the appropriate software engineering methodologies and diagrams.
- To comply with the attendance procedures specified for the lab sessions.
- To gain experience in working with external code and libraries
- To apply programming knowledge to a new language
- To be able to manage source code through version control systems

Deliverables:

Part A:
15% -A prototype of your game to be demonstrated in Lab Session 6
Due: 5/3/13 assessed in the lab.

Part B:
40% - Completed game implementation
40% - Development Portfolio (3000 words approx)
Due: 14th May 2014, 4pm as a zip file via Assignment on moodle.

Part C:
% - Full attendance (for the 1hr duration) for **at least 6 out of 10 FSE lab sessions**

Expected Content of Design Portfolio:

There are various sections required for the completion of this component of the coursework. All of these components are essential for you to obtain good marks in this section. We recommend you work on at least one section of the portfolio per week., following the plan provided in the next section:

- Game concept, requirements and specification (functional and non-functional requirements)
- Design of your game including any rules, controller behaviour, interaction design, scoring mechanism and game overview
- Evaluation of prototype and design refinements
- Implementation notes and description of the development methodology
- Evidence of testing and debugging including a write up of test cases and /or use of a test framework
- Evaluation of the game through user evaluation

Help Producing your Games

Python: there are many excellent python tutorials available online, some on our moodle page and via youtube, it is worth taking the time to run through the basics. If you are having trouble understanding python please contact one of our lab demonstrators for additional assistance. The best tutorial is available at http://www.python.org

Pygame:

Again many good tutorials exist for pygame, this one is particularly useful :
http://inventwithpython.com/

Need immediate help? Post your question on the FSE moodle discussion forum or tweet using the hashtag #fselabs on twitter.

Advice and Considerations:

- Firstly you must chose between using the Greenfoot and the python/pygame development frameworks. Use your experience in labs 2 and 3 to guide your choice. Higher marks (75+) cannot be acheived in Greenfoot, so bear this in mind when you are making your choice.

- No other language or framework will be accepted, due to having to ensure marking consistency between assignments.

- If you spend all of your time developing your game and do not complete your portfolio you will not pass this coursework! The documentation and evidence you can go through a process of software engineering is the point of this coursework.

- Diagrams are a good way of expressing design ideas, there are many formal varieties but even simple block diagrams can help get your point across clearly.

- You might not want to try to make a scrolling game in 2D - this is not usually trivial and may detract from the point.

- If you need inspiration check out some retro 2D platform games available on the web at http://www.classicgamesarcade.com/

- Pick your partner wisely - you will be awarded identical marks for this coursework!
-
- If you have severe problems working with your partner speak to the module convenor who may be able to help you sort out your management issues.
-
- If you work on this coursework consistently it should not take you more than 30 hours between you.
-
- Try not to leave this coursework to the last minute, we give you the facility to complete the coursework throughout the semester, make good use of the time.

Reminder: Plagiarism between groups of pairs or code downloaded from the internet will not be tolerated.

Confused? Not sure how to manage your project? Read on....

Lab Plan

Because you guys are not used to doing project management we have broken down the tasks for you into weekly chunks. Each week describes the tasks you should have done in order to complete the coursework with the minimum amount of stress, starting from Week 4.

If you dont want to follow this schedule, that is fine, make sure you hand in all components. Lab 2 & 3 are tutorials. Please make sure you have covered all the material before starting the coursework.

Lab 4 - Constructing Prototypes

Your first task is to chose the development framework. Once you have made this choice start to work on your prototype. To do this make a small provisional set of requirements from which you can work.

Lab 5 - Testing prototypes

Todays task is to complete your prototype. By the end of today you should:
complete your prototype ideally demonstrating two or more features of your game (without awesome graphics or super complicated multilevel game play)
write a brief game synopsis 500 words

Lab 6 - Demo and Feedback

By this point you should have your prototype completed and ready to demo. You will be asked in this session to present your work to date to a FSE demonstrator. This demo will be assessed. Use the remainder of the session to update and improve your requirements based on your feedback.

Lab 7 - Code Improvements

This week you should start the full implementation of your game. You should complete unit tests of your code as you implement, and keep a record of your testing for use in your design portfolio. Hint: python has a built in unit testing tool called unittest. OR use another standard python test harness. Evolve your game to include improved playability, graphics and/or multiple levels

Lab 8 - Game Refinement

Continue working on your final game. At this point start writing up your game portfolio. Dont forget to go back and refactor your code.

Lab 9 - User Evaluation

In this lab session, use other students to evaluate your game and record their feedback. Make any final improvements to your game.

Lab 10 - Assemble Portfolio

Work on completing your game and your write up.

Over The Spring Break

The final tasks for the project are designed so that it is possible to complete over the easter break:

- complete your python game
- start writing up your Development Portfolio
- perform an evaluation of your game with at least two participants (yes, it can be your Nan).

Marking Criteria

The Game Element:

40: Fully complete 2D game with multilevels; uses external libraries and version control; complex game mechanics; good coding style; addictive and playable; original game idea.

30: Good quality game, some errors, high game complexity and original ideas demonstrated; consistent coding style and good code decomposition.

20: Simple game, decent implementation, but apparent bugs; somewhat playable

10: Some game elements implemented; lots of mistakes; poor implementation

<10: Not actually a game, extends basic scenarios.

The Portfolio Element :

40: Well written; all parts complete; systematic testing and user evaluation performed; well written; uses UML in design.

30: Good report, minor mistakes or issues; weaker evaluation section; all parts complete.

20: Satisfactory report, with some omissions and design details not present.

10: Unsatisfactory, not enough detail, inconsistent design, incomplete or missing evaluation.