A5   –   Complete Implementation

Lewis Edwards   –   708830
Callum Hazelton   –   709161
Gavin Tsang   –   658679
Daniel Squires   –   709547
Ben Golightly   –   659715
Geraint Howard   –   710909
Jiaming Dong   –   742299

Contents

Contribution Overview

This section describes who contributed to what classes and sub-systems as well as unit and integration testing.

Name: Lewis Edwards
Classes Contributed to: SaveManager
System Contributed to: N/A
Subsystems Contributed to: SaveManager
Tests: N/A
Manager Role: Planning & Quality Manager

"Lewis Edwards contributed to the SaveManager class. This class is responsible for saving the current game as a .txt file named "SaveGame". As planning and quality manager, he was responsible for making sure the group met often enough to keep the bond between group members through the course of the assignment. He also took minutes for all formal meetings. He ensured that he commented his code correctly with doxygen. Tests were carried out on the SaveManager through the LoadManager class."

Name: Callum Hazelton
Classes Contributed to: Clock
System Contributed to: N/A
Subsystems Contributed to: Timer
Tests: Clock: main method test
Manager Role: Doxygen Manager

"Callum Hazelton implemented the Clock class. The clock class was part of the GUI subsystem meaning he had to work with other people on the same subsystem, to make sure it linked and worked well. There weren't many tests that could be made on the class, apart from to check the timer worked, which he did in terminal, and then check to see if it shows up in the GUI. He also helped with coding conventions."

Name: Gavin Tsang

Classes Contributed to: Load Manager
System Contributed to: N/A
Subsystems Contributed to: Game Logic, AI, Save and Load, Timer
Tests: Load Manager
Manager Role: Implementation Manager

"Gavin Tsang designed and programmed the LoadManager class which provides a platform to recreate a game via data retrieved from a save file. As implementation manager, he was also responsible with providing knowledge and guidance on the implementation of the entire program. As such, he contributed to the majority of classes such as the AI subsystem, load and save manager and timer. Furthermore, in order for the newly created AI, load and save manager and time subclasses to provide input and output to the original game, Gavin altered the original code to provide a link between these subsystems and the program. Gavin also improved on the original design of the program by eliminating the use of String type variables in order to indicate players and pieces via the use of Color type variables."

Name: Daniel Squires
Classes Contributed to: Animation, Connect4Animation and Connect4AnimationPane
System Contributed to: N/A
Subsystems Contributed to: Game subsystem, Animation Sub-System
Tests: Connect4aAimation, Connect4AnimationPane
Manager Role: Implementation Manager

"Dan was responsible for adding animation to the game of Connect Four to meet the specification, he was also responsible for Othello, but this was given to Ben when he returned from illness, and for implementing the Animation Superclass. This included coordinating with Ben to get the animations working with the implementation that we got. His implementation has delivered responsive, attractive animations."

Name: Ben Golightly
Classes Contributed to: OthelloAnimation, OthelloAnimationPane, OthelloAnimationPoint, and enhancements to Othello and ConnectFour to highlight winning peices
System Contributed to: N/A
Subsystems Contributed to: Game subsystem, Animation Subsystem
Tests: OthelloAnimationTest and OthelloAnimationPointTest
Manager Role: Customer Interface Manager

"Ben was responsible for adding animation to the Othello game to meet the requested features. This included coordinating with Dan (who implemented animation for Connect Four and the Animation superclass). His solution delivered attractive animations that are performed in sequence when pieces are captured, as specified. He also updated the Othello and Connect Four classes to meet the requirement that the winner's pieces or winning move is highlighted when the game ends."

Name: Geraint Howard
Classes contributed to: AIEasy, AIEasyTest
Systems contributed to: N/A
Sub-systems contributed to: Player
Tests: AIEasy
Manager role: Test Manager

"Geraint was responsible for the implementation of the AIEasy class and the AIEasy test. This is a class that makes the computer choose any random legal move for both of the games. As test manager, he made sure that every class had tests defined, building on the feedback that he received from the assessment 4 demonstration."

Name: Jiaming Dong
Classes contributed to: AIHard
Systems Contributed to: N/A
Subsystems contributed to: Player
Tests: AIHard
Manager role: Design Manager

"Jiaming Dong contributed to the AIHard class. This class could let the program able to make a move with logic. As the design manager, although we are completing a project from some another group, he still tried to improve the designing of it."

Here, the designated managers answer relevant questions on their role.

Planning & Quality Manager – Lewis Edwards

Is the team meeting often enough?
Yes, definitely. In comparison to last assignment, the team has met (and recorded minutes for) on 8 occasions, in comparison to 3. The team also met on other occasions to do group programming.

Is the team following the minutes protocol?
We are following the minutes protocol more closely than last assignment. We have used individual names for the things discussed in meetings and for things to do, so members had a more clear vision of what was discussed, and what needs to be done.

Does the team use version control?
Yes, we used Github in this assignment. We found it very useful for getting rid of clashes of clashes in different versions.

Are code inspections being carried out?
Yes. With Github, all members were checking the code as updates were being implemented. 7 people's eyes are far superior than 1, and this helped us keep errors to a minimum.

Do quality of all classes and sub-systems meet the team's quality standards? If not, what is recommended to fix this?
We ensured that the quality of our code is high. By carrying out code inspections with each update that was made, all members checked the code, and made suggestions for changes, or made small corrections themselves.

What is our plan in case member A does not contribute?
Our plan was to give all members equal amounts of the class that would have gone undone. This would spread the workload between all members equally.

Doxygen Manager – Callum Hazelton

Is implementation properly documented using doxygen?
All classes have sufficient comments on the code.

Implementation Managers – Gavin Tsang

Are all team members competent in Java? If not, what remedial actions are recommended? Everyone should have a programming reference book. Not all team members were very confident in programming in java. In order to help improve their programming, I as the implementation manager, sat down with them and helped instruct them on any concepts they were unsure about and help bug test. As such, all team members were able to implement code that is bug free and feature complete to the best of our knowledge.

Is design documentation clear enough to begin implementation?
The design documentation was not very clear on exactly how the program should be implemented. Furthermore, the original code base didn't consistently follow their design documentation either and thus we couldn't follow design during our implementation.

Are all team members following Bob's Concise Coding Conventions?
The team members have generally been following the Coding Conventions to the best of my knowledge. Occasionally magic numbers and directly accessing global variables would crop up, but most violations should be mostly eliminated.

Are team members taking advantage of shared and or reusable code where possible?
Yes, areas of code such as the qualityMove method makes use of code in our group's original program. Through the use of abstract superclasses, any code that is consistently reused is moved up the hierarchy.

Implementation Managers (cont) – Daniel Squires

Are all team members competent in Java?

A couple of members have shown to be very strong in Java and have needed no/little help in programming their classes, the rest of the group have ranged from pretty competent to fairly competent, and needed help from either myself or Gavin at times to complete their classes.

Is design documentation clear enough to begin implementation?

The other group did not follow their own design documentation so we were not able to do so either.

Are all team members following Bob's Concise Coding Conventions?

Yes, Gavin and I have made sure that everyone has followed Bob's Coding Conventions when they have been coding, and we as a group are rigorously going through the classes, both ours and the other groups, to get rid of as many convention breaks as we possibly can.

Are team members taking advantage of shared and or reusable code where possible?

Most team members have taken advantage of shared and/or reusable code when and where possible, the only time this has not happened has been the AI classes where a superclass may have been a good idea due to a fairly large amount of duplicate code, but apart from that, shared and/or reusable code has been used. The Animation hierarchy is a good example of this as where-ever code is shared between the 2 sub-classes, it was moved into the superclass for better access.

Customer Interface Manager – Ben Golightly

Are we being responsive to customer requests? We have been ruthless in going through the requirements and making sure that everything is implemented that is practical to implement in the timescale – see the next point for one feature that could not be met.

Is quality of requirements documentation sufficient to guide development? The requirements documentation is split across several PDFs but with occasional questions for Bob and careful reading, augmented by feedback from the partial implementation, we are confident that we will have implemented every required feature.

However there is one exception: allowing the user to choose the colour of their own piece was not possible in the code base we inherited. This problem was not solvable without a complete re-engineering of the product, as this was hard coded across dozens of files. To do so would require renegotiating the time-scale for the product release.

Do all team members understand requirements? I have gone through the requirements for things we overlooked, for example the ability to highlight pieces on victory, raised these points, and made sure the features were understood and implemented.

Testing Manager – Geraint Howard

Are test plans produced when the process needs them?
Yes, test plans are produced when the process need them.

Do all classes have unit tests defined? Are all methods tested by unit tests?
Yes, all classes that are able to be tested have had tests defined for them. The only classes that does not have a test defined is the saveManager class which we were unable to test

Are test plans complete and thorough?
Yes, test plans are complete and thorough.

Does each team member understand how to produce tests? If not, what remedial action is recommended?
No, every team member did not fully understand how to produce tests for their classes. However, with the help of the other group members this problem was rectified and tests were defined for every class.

Design Manager – Jiaming Dong

Do all team members understand how to use design methods?
We got a good understanding of using design methods and use them when designing.

Is the team's design work of high quality?
We did some improvements to the project we received to make them easier to be used.

Does software design consider future product evolution?
We try to make our design as simple to be understood, and our code more re-usable which could let the program easier to be evaluated in the future.

Is each team member contributing to design?
Every team member has re-designed the class relate to their work.