Coursework Two - Documentation

Gan Cao, Michael Cooper, Cheng Jiang, Olivia Ruston, and Kat Wojna

December 2018

Introduction

This document presents an overview of the 7 week process of creating a game for the Software Engineering unit using agile methodology. The document is comprised of seven chapters that correspond to the seven sprint cycles completed and includes the aims and goals for each cycle and the entire project. Each sprint cycle begins on a Thursday and ends on the Wednesday. Customer interviews occurred weekly on Wednesday afternoon and formed the basis for the start of each sprint. As a result of this, some documents, such as meeting minutes or programming logs, might appear twice in two different sprints depending on their relevance to work being carried out.

Each section starts with an overview for the entire sprint cycle. It then goes on to analyse the customer interview from user stories are written. The implementation of the new changes are then discussed, including pair programming practise and the testing of the product at the end of each iteration.

In addition, this document records the process of creating the game; from the concept to the final product, the implementation successes and failure, the wide range of problems we faced and the solutions we came up with to counter them.

The document contains an appendix which consists of all the documentation that was completed during the development process. These documents provide the reader with a framework for task and resource allocation, attendance records, and the design and iteration processes. The appendix is divided by sprint chapters and is referenced throughout the report as a justification of our methods.

Chapter 1

Sprint One: 18/10-24/10

The first sprint cycle was mostly focused on organizing work, project management and deciding upon the tools we would use to support cooperative work. This included discussion about potential task allocation and a conversation about how we will accommodate the coursework within our different schedules. We spent this time to getting to know our strengths and weaknesses and how to make the best of them in order to create a well balanced team (see appendix A.1.1).

During the first week our team focused on the brainstorming the ideas for the game-play and game mechanics for the second course work. Our goal for this sprint was to loosely define several game ideas (see appendix A.1.2), which were to be presented to the customer during first meeting, and to establish how we would manage the collaborative work (see appendix A.1.1).

Project management

In the first week, we focused on deciding upon our approach to managing the project. This included identifying potential risks that could impact the project and a discussion about the creation of a Gantt chart to provide an overview for each sprint (see appendix A.1.2). A shared folder on Google Drive was used as the central location for all of the documents relating to the project, such meeting minutes and notes from cutomer interviews. We agreed to use Trello as our task allocation tool, using Kanban methods to keep track of work in progress. It was decided that GitHub would be used to for code storage and version control (see appendixA.1.1). Eclipse was suggested as one team member as a suitable IDE with the ability to easily commit changes to GitHub repository.

Game concepts

During our brainstorming session we used the whiteboard as our main tool to present the ideas to the group. This way of creating the game ideas was taken from the ego-less programming concept, and it was done to encourage the group collaboration and ownership of the game. We discussed how the game will be played, how the difficulty is going to increase to keep player challenged, the potential collaborative parts of the multiplayer, the view the player will be presented with, and game objectives (see appendix A.1.3.

1.1 User Stories

No user stories were created during this sprint cycle.

1.2 Pair Programming

There were no pair programming activities during this sprint cycle.

1.3 Tests

As above, no tests were done during this sprint cycle.

1.4 Customer Interview

We conducted the first customer interview at the end of our sprint cycle. The discussion was driven by the work carried out in the brainstorming session. We created three distinct game concepts. Each concept had a different approach to game feedback, mechanics and genre. We encouraged the customer to choose freely from the board with a pick and mix approach. The interview was recorded and then transcribed. The aim of this interview was to gain insight into customer requirements which then inform the development of user stories in the following cycle (see appendix A.2.1.

Chapter 2

Sprint Two: 25/10-31/10

In sprint two, the majority of the work concentrated on developing user stories and scoping of the project. Following the initial customer interview, it was decided that during the project we would develop a game in which the player programmed collaborative bots that sort out collectible resources. There was a lot of discussion as to what would be achievable within the time allowed bearing in mind the skill and experience of the team (see appendix B.1.2). It was decided that full bot programmability would be quite challenging and therefore we would aim to allow the players to execute blocks of commands, e.g. "move left, move up, move up" in one turn as opposed to "move left". If time permitted, we would then consider implementing a programmable element. We also discussed more basic aspects of the game such as how the game spawns, number of turns, and the mechanics of resource collection (see appendix B.1.3).

During this sprint, we refined some aspects of project management. We reviewed our initial Gantt Chart and finalised the risk assessment document (see appendices B.1.1, B.6.1 and B.7.1). One team member demonstrated how to set up the digital workstations to be used during programming. This included how to link Eclipse to GitHub, how to make commits and how to manage version control (see appendix B.1.2).

At the end of the second sprint cycle we had created small map with a randomly spawning bot. This formed the foundations for development in sprint three.

2.1 User Stories

Having transcribed the customer interview from last week, we analysed the transcript and used this to develop a series of user stories (see appendix B.2.1).

2.2 Pair Programming

During the second sprint cycle we managed to create the basic map that prints out with a randomly spawn bot in it (see appendix B.4.1). This work was not completed through pair programming but was later presented an explained to the rest of the group in a meeting (see appendix B.1.2).

2.3 Tests

At the end of the cycle we started writing the test cases from the user stories and then expand them for this version of the game and for the next iterated version. Some of the test cases we wrote during this sprint cycle we aimed to pass in later cycles. B.3.1 B

2.4 Customer Interview

In this customer interview, we wanted to update the customer on our revised product scope. As said previously, it was decided that we would aim to develop a game which would allow the player to execute blocks of commands in one turn. We also presented the customer with a visual concept for the final game (see appendix B.8.2), to support discussion and provoke the customer to ask questions about game play. The aim was to collect user stories about how the customer expected to interact and receive feedback from the game (see appendix B.9.1).

Chapter 3

Sprint Three: 1/11-7/11

During the third sprint, we produced a basic game program. The code written in sprint two was re-factored to allow for more versatility in the future (see appendix C.1.1). During meetings we continued to have discussions about the basic mechanics of the game, including how the bot should move and collect objects. These were decided in practise during programming sessions.

3.1 User Stories

The interview from the previous sprint was transcribed and analysed. From this more user stories were created and added to the backlog (see appendix C.2.1). This allowed us to prioritize tasks, decide on how to allocate time and resources, and how to develop the basic game framework.

From User Stories we started developing a diagram of Use Cases to build a mental model of how the game would operate, what are the goals of each action and what to do to achieve them (see appendix C.3.1). We used use cases to think about testing; as in wrong input/output, the working condition of the system or unresponsive bot.

3.2 Pair Programming

As said previously, in this sprint the focus was on developing a versatile base game. Work was done on creating class to represent a single map tile within a map. The map tile class included information about the stored objects within the environment, such as bots or collectible objects. This allowed different types of object to be in the same tile at the same time. For example, a bot and a coin can be in the same place, allowing the bot to collect the coin (see appendices C.5.1 and C.5.2).

In addition to this, work was also done to create an arbitrary sized map, allow for unlimited number of bots and collectibles within the map, and re-printing the map after the bot made a movement (see appendix C.5.1).

In preparation for sprint four, we decided to split our team into two teams working simultaneously on two different versions of the game (see appendix C.1.3). Each version explored a different approach to turn taking and move making in the game. In one version, the each turn

allowed the player to make a set of moves, e.g. left, left, up. In the second version, the player decides upon a set of moves at the start of the game and watches as these the bot makes a move incrementally. The idea being that the second version could form the foundation for further development into bot programmability. We agreed from the beginning that one of the versions would not be included in the final project. However, we treated it not as "wasted work" but rather as a back up plan for our risk assessment. If we were not able to create the programmable bot, we would still have a working product with less of the functionality.

3.3 Tests

Whitebox Testing and Code Review

At this stage, any code testing was made "on the spot" since it was easy and reasonable to fix the problems with code as they emerged. When pair programming, we looked at the code we had written and tried to understand any issues together (see appendices C.5.3 and C.5.5).

Blackbox Testing

We mostly based our testing on the User Stories during this sprint cycle. We created the scenarios for the game and checked the game play against customer expectations (see appendices C.4.1 and C).

3.4 Customer Interview

We presented our first basic game, asked for the feedback and discussed the solutions for existing issues (see appendix C.6.1).

Chapter 4

Sprint Four: 8/11-14/11

During this sprint cycle we worked on how the player interacts with the bot. As mentioned in the previous chapter the team worked on two different approaches to player commands and turn taking. One version was for the player to have multiple turns and within each turn the player could make multiple commands (see appendix D.5.1). The second version was one turn in which the player can make multiple commands and these commands are then made incrementally. Progress was reviewed and compared during a meeting just before the end of the sprint (see appendix D.1.2. The decision between the two was simple to make as one team had gone beyond the task and developed a simple language and interpreter for controlling bots within the game.

Having made the decision to move forward with programmable bots, we began discussing other aspects of the game mechanics. This included which bot starts first, interface for the user, how to keep the player engaged during the game, increasing the difficulty, and keeping track of scoring (see appendix D.1.2).

In addition to development, time was spent collating the documentation and agreeing upon presentation. We reviewed previous sprints and made notes on what needed improving (see appendix D.1.1).

We also discussed planned for the following sprint and how to improve out current approach to testing (see appendix D.1.3.

4.1 User Stories

From the interview transcript and analysis, we adjusted our user stories to incorporate the new, more sophisticated player control and to reflect the current state of the game (see appendix D.2.1).

4.2 Pair Programming

As mentioned in the previous chapter, in this sprint cycle we had two teams working on different tasks. We decided to move forward with work completed by the second team as they had developed a way to program bot behaviour (see appendices D.5.2, D.5.3, and D.5.4). Having

decided on version, work was done to improve other aspects of game play. This included fixing the number of the bots to two, initiating each new turn by hitting "ENTER", adding an idle statement and implement allowing white spaces as a part of the split function (see appendices D.5.5, D.5.6, and D.5.7).

4.3 Tests

Like with other sprint cycle, most of the code testing was done during the pair programming (Whitebox and Code review) sessions.

It was done by simply running and rewriting the code, until we got the desired outcome for the code.

Similarly the usability testing was done by checking the customer requirements from the interview analysis and running the game through the user scenarios and playing the game (black-box testing). D.1.1. D.2.1 D.3.1 D.4

4.4 Customer Interview

Following the same procedure, we presented the existing game with additional iterations and functionality, asked for the feedback and suggestions for the improvement (see appendix D.6.1).

Chapter 5

Sprint Five: 15/11-21/11

During sprint five our team focused on two important aspects of the project. We started focusing more onto creating the properly structured report, reviewing the existing documentation, uniforming the minutes and programming logs to fit report template (see appendix E.1.1).

The main goal for this sprint was to create a graphical user interface (GUI) for the game. We achieved this goal and were able to present the game to the customer using an executable file (see appendix E.1.3). We also started improving our game by creating a more enjoyable game-play experience. We refined the amount of the bots, the pace of the game and how instructions are presented to the user (see appendix E.1.1).

Our aims for the next sprint are to allow for players to have teams of multiple bots and to improve upon the user interface. In the current sprint, the game exists in two versions; one that includes the GUI and one features the programmable bots. In the next sprint, we will merge the two (see appendix E.1.2).

5.1 User Stories

From the interview transcript and analysis, we reevaluated the user stories and added new ones. In addition to this we checked the compatibility of the existing progress against the new and old requirements (see appendix E.2.1).

5.2 Tests

We kept testing our game with refernce to user stories, doing the pair programming and by Black-box testing the game. Most of our testing focused on resolving the problems while they occurred (fixing bugs on the spot). We kept focusing on the making sure each version is error-free before we decided to progress with another change (see appendices E.3.1 and E).

5.3 Pair programming

The fifth sprint cycle was focused on improving user interaction and creating GUI. We have created a window for the player to put the number of bots or change the size of the map. We have created the GUI windows that the player can interact with the bot by using the buttons and fixing the syntax code (see appendix E.4.1).

5.4 Customer Interview

This is the first interview where our game started looking a little more like a game with the addition of the simple GUI. During this interview session we presented the customer with the two branches of the game which we planned to merge by the next sprint (see appendix D.6.1).

Chapter 6

Sprint Six: 22/11-28/11

In this sprint cycle we focused heavily on project management. Two members of the team will not be available in the last week of the project and therefore we had to adjust our deadline to account for this. In addition to this we established aims for the current and following sprint, prioritised individual tasks, and continued to work on the documentation (see appendix F.1.2).

During this sprint cycle we merged the two versions of the game (GUI and programmable) into a a single version of the game. We improved upon the GUI by adding a scroller for the map and adding more interesting graphics. In order to improve player experience, a 'mini-map' was added to allow the player to track bot position within the map. Another aspect of game play that came up repeatedly in customer interviews was turn taking and the number of moves the player could make per turn. In this sprint we addressed this by setting a maximum number of moves per turn (see appendix F.4.1).

In preparation for the following sprint, we started thinking more about the play-ability of the game. Particularly focusing on how to measure the enjoyment and engagement of the user during the game-play. We have created a Likert Scale to get the feedback from the game-play during the next sprint cycle (see appendix F.3.2).

6.1 User Stories

As before, the customer interview from the previous interview was transcribed and analysed as a group. From this, we developed new user stories to add to the backlog (see appendix F.2.1). The closer we come to deadline the fewer new user stories there are to add.

6.2 Tests

The testing done in a similar way to the previous cycles. During this sprint cycle, we failed to incorporate most of our planned interaction, therefore we focused on making the code better and more functional (see appendices F.3.1 and F).

6.3 Pair programming

Due to other commitments within the team, we did not manage to complete as much development as we had anticipated for this sprint. However, our team members managed to complete the high priority aims for the cycle; this included merging the two versions and improving the GUI (see appendices F.4.1 and F.4.2).

6.4 Customer Interview

This sprint cycle we presented the customer with functioning version of programmable game with graphics and GUI (see appendix E.5.1).

Chapter 7

Sprint Seven: 29/11-5/11

During this sprint cycle we presented a further improved version of the game. Before play, the player can now select the number of bots and coins within the game. The user interface now includes provision for programming multiple bots. We had aimed to add dynamic map sizing and to increase the number of players. However, due to an unexpected illness within our team, tasks and resources were reallocated meaning we were unable to achieve these goals. While we were still be able to present the game to the customer, after this point no further development was possible and the game had to stay in its existing form (see appendices G.1.1 and G.1.2).

While we decided not to complete an eighth sprint, it is worth noting that we did continue to do some work. We focused on unifying, proofreading and cross checking the documentation. We also worked on additional documentation, such as game tutorial, project evaluation and the maintenance guide (see appendix G.1.3).

7.1 User Stories

We did not create any new user stories during this sprint cycle. however, we tried to resolve the ones from the previous sprint cycle (see appendix G.3.1).

7.2 Pair programming

The key development for this sprint was to add multiple bots per team and to add a screen title, where the game can be restarted or shut down. The other additional improvements were carried out to improve playability and make the interface easier to navigate. This included greying out areas when the text boxes are not editable or adding separate text areas for commanding each bot (see appendices G.4.1 and G.5.1).

7.3 Tests

In this sprint cycle, work was carried out to ensure the product passes validation tests. Considerable meeting time was spent cross checking the game with the previous test cases, evaluating

any small changes that could be made within this short period of time and what is no longer feasible to add (see appendices G.4.1 and G.5.1).

7.4 Customer Interview

In the final interview with the customer, we presented our final version of the game, which included multiple, identifiable bots per team and additional GUI changes. We played the game with the customer, and asked for the feedback (see appendix G.5.2).

We discussed what could be improved if there was more time, and how the game would benefit from further developments. However, we did manage to create a working product that met the requirements. With this in mind, and considering the little time and few people on the team, we have decided not to develop the game any further (see appendix G.6.1).

Chapter 8

Project Evaluation

This was one of the most difficult projects our team members had worked on. The seven weeks duration with additional half of the week was not a sufficient amount to finish the project to the standard we had hoped. However, it is worth noting that the real life software engineering teams often face similar problems with time management, team working and presenting a product that is not fully developed.

We decided to create a fairly ambitious and sophisticated product, with the "bot programability" as a main feature. As a result, we did not focus on the aesthetics and graphics for the game as much as we probably should. Making the game programmable was a big challenge, especially with majority of our members having no significant background in programming.

We did not face major disagreements regarding the shape of the game or the directions we should take as a team. Despite not being able to achieve all of the goals such as dynamic size map, we managed to preserve the general idea we started with.

With the frequent meetings, we were able to keep all members informed about the project progress and tasks for completion. We did not need to use any external tool for the group management, beside the Whats App group, which we used for scheduling meetings. Agile methodology is very friendly in terms of the group and task management. Without the need to create unnecessary documentation and with pair programming as our main way of creating this product, we could focus on resolving the emerging issues as they arose. This includes both development and team related issues, e.g. illness or other commitments.

Exception Handling

Our team had two major issues that required to be taken into account when planning and developing the project. One of the issues was the shortage of one team member within our group. This required us to allocate a greater workload to each team member than other teams. As a result, this might have led to producing lower quality work compared to other teams.

The second issue relates to project management and planning. We had to bear in mind that two of our team members would be unavailable for the last three days. This forced the deadline to moved from the 14th December to 11th December.

During the last sprint of our team member fell very ill, which required hospitalization. Taking

into account we already were one person short, this added additional pressure on the team. This affected our ability do planned last sprint testing and implementing additional fixes to the game.

Project Iteration Process

Each cycle started with the analysis of customer requirements, based upon conversations had during customer interviews. We recorded each interview, transcribed and analyzed it as a team to work out the requirements for the game. We then worked on development, based upon user stories. After each sprint cycle, we checked our test cases to validate the current build. The code testing was mostly done in an informal way, on the spot with either pair code review or just by running the code and testing if it works. We followed this structure through each cycle to ensure the product we were building was "the right product" not just "a working product".

It was clear from the final customer feedback session that there were improvements that could be made. Many features were not clearly explained, there were some syntax errors in the programmable side of the game and the game itself not easy to "understand". However, we received positive feedback regarding the difficulty of the game and customizability of the game.

We managed to fulfill the basic requirements set out in the coursework specification (multiplayer, bots, dungeon) and added a twist of our own (programmable bots, multiple bots per team). As a result we believe the implementation of the game was successful in terms of creating the products and following the agile methods.

If there was a possibility for further iterations (more team members, more time), we would try to implement intelligent agents to do the game, make bots aware of their environment, work as a team, and switch the game to the online multiplayer mode. This would definitely increase the amount of fun the players would have with the game. However, with our set of skills, amount of time and team members, we deem the project to be fairly successful.

Appendix A

Sprint One: 18/10-24/10

Return to chapter 1.

A.1 Meeting Minutes

A.1.1 19/10

Meeting Minutes - 19/10

Attendees: Kat, Alex, Michael, Olivia, Andy

Introductions

- What everyone does
- Skills
- Previous Experience

Communication

- Shared folder on Google drive
    - Project documentation, e.g. meeting minutes
- GitHub for version control

Next Meeting - Tuesday 23/10

- Bring game concepts to present customer
- Discuss questions for customer interview
- Project Management

A.1.2 23/10

Meeting Minutes - 23/10

Attendees: Kat, Alex, Michael, Olivia, Andy

Discussing game concepts

- Kat: single/multiplayer, gets progressively harder, players build a score
    - Beating monsters
    - Score = how long you live
    - Death is outcome of game
    - Output in plain text
    - Control with WSAD
- Michael: single/multiplayer, collaborative, return on investment
    - Move up the tower block
    - Office tools
        - Loot from filing cabinets
    - Aim is to kill the boss
    - Top down view
    - View is static - overall view of floor
    - Defeat all enemies on each floor
    - Control with WSAD
- Olivia: single player, princess escape from palace
    - Collect objects
    - Use objects to meet some goal
    - Keypad control / on screen control

Edit concepts slightly to show variation in games - offer options

Project management

- Kanban
    - Use Trello
- Ganntt Chart - overview of sprints
- Take some elements from various agile methodology
    - Scrum type
        - What has been done
        - What will be done before next meeting
        - Problems faced

Customer Interview

- Get familiar with question
- Finalise game concepts
    - If possible scan and upload to drive for everyone to see.

Risk Assessment

- What are we doing to prevent failure

Next Meeting - Friday 24/10 but use some time Wednesday 26/10

- Plan iterations

- User stories
- Scope project

A.1.3 24/10

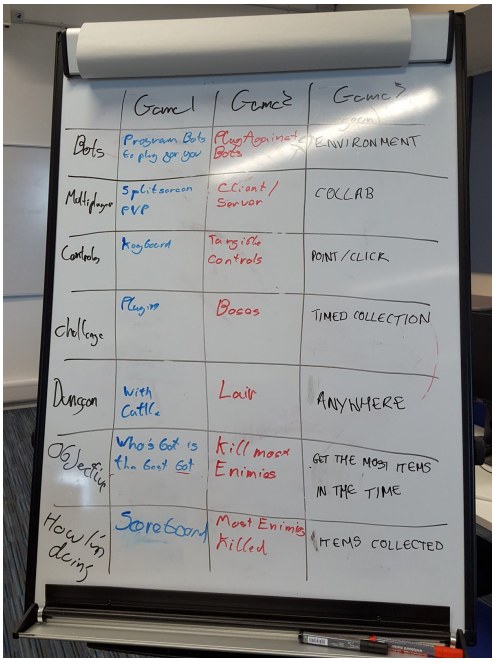
Meeting Minutes - 24/10

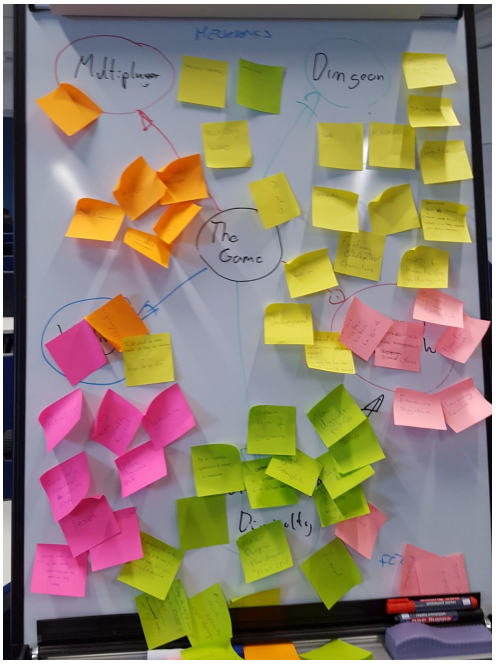
Attendees: Andy, Alex, Kat, Michael, Olivia

Brainstorming session

- Building on from previous game concepts
- Ideas as a group
    - Encourage project ownership
- Five categories
    - Multiplayer
    - Dungeon
    - Writing bot
    - Change the difficulty
    - Scores/How to win
    - Feedback
    - Mechanics

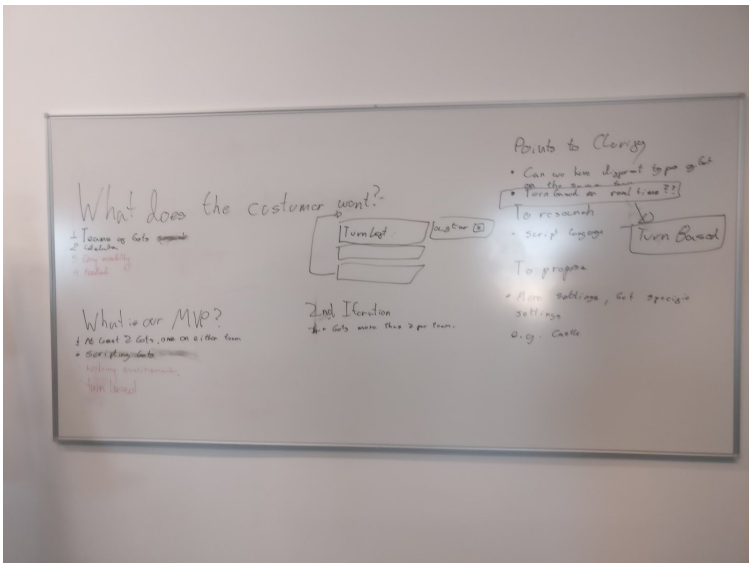
- Use various ideas to create three concepts



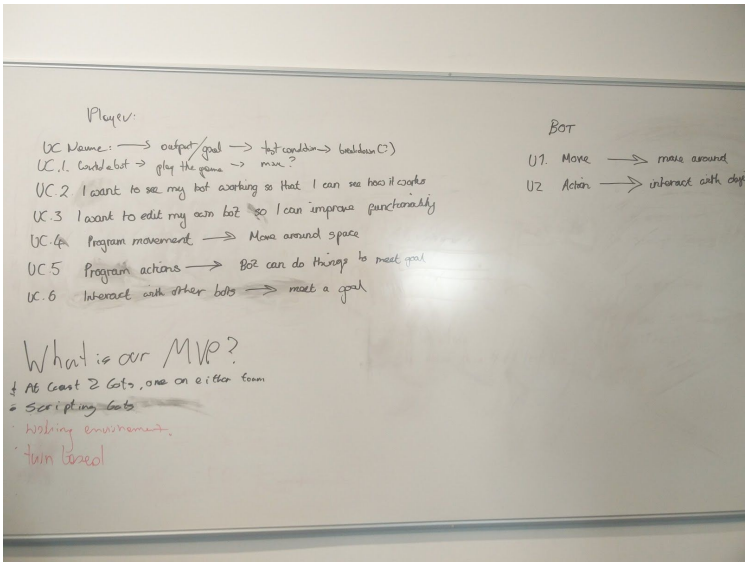


- Customer interview
    - Programmable bots
    - Create teams that complete some tasks
    - Castle theme
- MVP

- See picture



- Basic Use Cases



A.2 Customer Interview Documents

A.2.1 Interview Transcript

Interview Transcript - 24/10

Introduction to game concepts.
Skipping forward to input from the customer.

O: So which bits do you like the sound of?
J: I like the sound of the bots that you plug in that play for you. That's different, that's interesting.

Pause to retrieve whiteboard markers

J: It would be interesting to possibly have the bots collaborating with one another.
O: Okay.
J: That would be something, because that requires communication.
O: What kind of task do you want them to complete? Do you want them to...
J: They could be roaming an asteroidal planet couldn't they? No, does that bring back bad memories?

Laughter

O: So do you them to collect things or complete tasks or do you want it to be a bit more?
J: I'm liking the sort of story scenario on the first column. So it could be that there are multiple teams of bots...
M: Okay so more perhaps, instead of programming an individual bot programming an army persay..?
J: Not necessarily, it could be that you program your bots and then a bunch of bots get thrown together to work together as particular team.
O: This sounds so familiar.

M: I tell you what, er, ooh...
O: We are writing notes and people are recording.
M: I mean it would be nice to write it down...so...er...have bots work together.
J: We've got collab on the other side that has been circled.

M: Okay.
J: Erm so, I'm not sure where the controls come in if we're going down the rows here. In that case because...
O: You're programming.
J: Yeah that's right so there's going to be the question of how are you going to allow...how are you going to allow me to program it?
M: I mean with regards to that, would you like a sort of programming scenario where you're using a formal programming language in running scripts or would you like something a bit more high level. So have you come across Scratch for instance?
J: Yes.
M: So something a bit like that where you're taking modules and you're sort of layering modules so your user interface...
J: I can see that being more accessible...
M: Yeah.

J: Than writing it in conventional programming language. So, stepping aside from being the customer, I'm worried about the amount of work being involved with all of those things. So I think trying to find a sufficiently low effort way through to delivering programmability is going to matter.
M: I think, you know, what we can do is interacting with you going forward is say on our MVP, our minimum viable product, we simply say you can only program with a script and then try and later on add that as a feature potentially just keep that in the back say. And say, you know if its possible to add on the user interface based way of interacting that will be great but we can simply run scripts. Do you think that's a useful thing?
J: You might also be able to find a suitable, more accessible scripting language like you said. There is scratch but you know scratch is actually pretty painful in terms of you've got no abstraction mechanisms. So you can't build procedures. I think that would be quite problematic. So I'm not sure what else, I'm sure there are plenty of scripting languages available. It just requires a bit of researching.
M: Okay, excellent.
J: Then we've got the environment. And so I think visually the first one offers the more opportunity to do something than... I mean anywhere is...
O: You may have had a specific idea that you wanted

Digression into previous years where one group did a Jane Austen world.

J: So for the moment I think I find the castle option the most attractive, it doesn't mean you have to go with that you can have something else.
M: So say if we come next week, it would be good for you to see potentially more specific options for environments then?
J: Yep, yeah.
M: Sure, that's something we can certainly do.
J: Erm yes. So who's bot is the best bot. Well if they're collaborating though that may not be the case, which team is the best team. I think that follows naturally.
M: Okay.

Timer goes off

J: Well we've still got the bottom row, erm...
M: In reality they follow naturally on.
J: Yeah I think so. Are there any critical questions that you want to ask?
O: I don't think so.
M: I don't think so, does everyone else agree?
O: Right okay. Thank you.
M: Thank you very much.

Appendix B

Sprint Two: 25/10-31/10

Return to chapter 2.

B.1 Meeting Minutes

B.1.1 26/10

Meeting - 26/10

Attendees: Alex, Andy, Kat, Olivia, Michael(present for first half)

Improving user stories

- Store them on trello
    - Can login with Google account
- Created some user stories - they can be edited

Review Gantt Chart

- Happy with the Gantt
- Gives us something to follow / Keep us on track

Work by next meeting

- Kat: Risk assessment
- Olivia: Develop starter program
- Andy, Michael & Alex: Attempt CRC
    - Useful CRC example: https://goo.gl/images/xxdq1y

B.1.2 30/10

Meeting - 30/10

Attendees: Olivia, Kat, Andy, Andrew

Discussing the requirements change regarding the gameplay

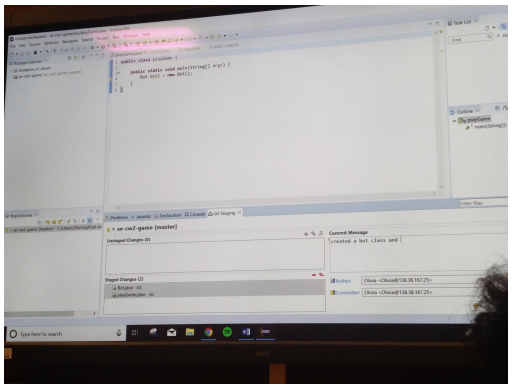
- First proposal for the iteration of how the player interacts with the bot
    - With the keyboard, not programming AI for the bot

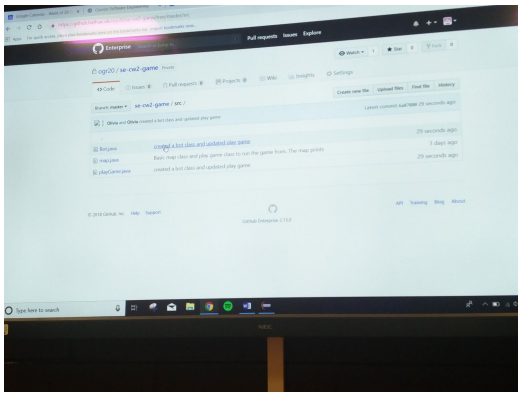
Managing Project

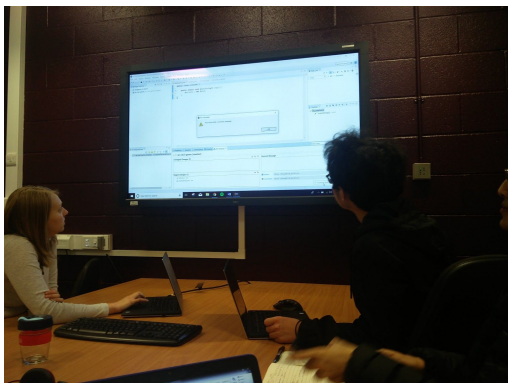
- Discussing pair programming
- Discussing progress of the project
- Discussing the schedule
- Setting up the digital workstations for the project
    - (eclipse to github)

10 minutes interview

- Change for the requirements







B.1.3 31/10

Meeting - 31/10

Attendees: Michael, Alex, Andy, Kat, Olivia

Michael pitches to keep programming bot game aspect
- Use interpreted language over compiled
    - Python over Java
- Create own interpreter
    - Takes user input and runs as set of functions
- Attempt runtime compiler

Solution - allow user to input commands as a block rather than one at a time.

Create “mock up”
- How the bot moves
- How it picks up resources
- Where it puts resources
- Turn based
    - Turn limit
    - Who can pick up the most resources within the same number of turns
    - Resources are just collected, no need to deposit
        - Collection automatic as you move over resource

Pair programming slots
- Friday 10:15 - 12:15
- Monday 14:15 onwards

To do for Next Meeting:
- Everyone is set up to program
- Transcribe the interview
- CRC finish them and upload to Google Drive

B.2 User Stories

B.2.1 25/10 31/10

SPRINT 2

2.1 - As a player I want to move around a map - make a map

2.2 As a player I want a bot so I can play the game - make a bot

2.3 As a player I want to program bot movement - allow player to make changes to bot movement

2.4 As a player I want to be able to see my bot working so I can improve it - create feedback loop

2.5 As a bot in the game, I want to collaborate and communicate with other bots so that we can work as a team - interactive

2.6 As a player, I want to see multiple teams of bots

2.7 As a player, I want to get bots to work together as a team

2.8 As a player, I want to program my bot so that I can control it

2.9 As a player, I want my programming language to be more accessible than a conventional one

2.10 As a player, I want to program functions so that I can make my bot functional

2.11 As a player, I want the game to be set in a castle

2.12 As a player, I want to know which team is the best - scoreboard

Backlog - Sprint 2	**In Progress**	**Completed**
1.1 - As a player , I want to move around a map - make a map	/	/
1.2 - As a player I want a the bot so I can play the game - make a bot	/	/
1.3 - As a player I want to program bot movement - allow player to make changes to bot movement	/	/
1.4 - As a boat I want to be interactive so i can interact with other bots/environment	/	/
1.5 - As a player, i want to plug in bots that can play for me	/	/
1.6 - As a bot in the game, I want to collaborate and communicate with other bots so that we can work as a team - interactive	/	/
1.7 - As a player, I want to see mutiple teams of bots	/	/
1.8 - As a player, I want to get bots to work together as a team	/	/
1.9 - As a player, I want to program my bot so that I can control it	/	/
1.10 - As a player, I want my programming language to be more accessible than a conventional one	/	/
1.11 - As a player, I want to program functions so that I can make my bot functional	/	/
1.12 - As a player, I want the game to be set in a castle	/	/
1.13 - As a player, I want to know which team is the best - scoreboard	/	/

B.3 Test Cases

B.3.1 25/10 31/10

SPRINT 2

2.1 - As a player I want to move around a map - make a map	pass
2.2 As a player I want a the bot so I can play the game - make a bot	pass
2.3 As a player I want to program bot movement - allow player to make changes to bot movement	pass
2.4 As a player I want to be able to see my bot working so I can improve it - create feedback loop	pass
2.5 As a bot in the game, I want to collaborate and communicate with other bots so that we can work as a team - interactive	fail
2.6 As a player, I want to see multiple teams of bots	fail
2.7 As a player, I want to get bots to work together as a team	fail
2.8 As a player, I want to program my bot so that I can control it	pass
2.9 As a player, I want my programming language to be more accessible than a conventional one	pass
2.10 As a player, I want to program functions so that I can make my bot functional	pass
2.11 As a player, I want the game to be set in a castle	pass
2.12 As a player, I want to know which team is the best - scoreboard	pass
2.13 As a player I want the game to end. NOTE THIS WAS ADDED ON THE 4th DEC AS IT WA OBVIOUSLY MISSING.	pass

2.1 - As a player I want to move around a map - make a map pass by sprint 2
- The map grid prints to the command line. pass
- The bot is visible on the map that was printed. pass
- The wasd keys move the bot on by one move on a single turn. pass
- Pressing enter finalises the turn and the game proceeds to the next turn. pass
- The map then prints to the command line again. pass
- The bot is also printed again and has moved in the correct direction. pass

2.2 - As a player I want a bot so I can play the game - make a bot pass by sprint 2
- There exists a bot. pass
- I can control this bot with wasd. pass
- The bot is visible on the map. pass

2.3 - As a player I want to program bot movement - allow player to make changes to bot movement pass by sprint 5
- Scripts can be written in an editor. pass
- There is a specified language for the script pass
- The game can read this script. pass
- The game assigns the script to a bot. pass
- When the game runs, the bots move according to the script. pass

2.4 - As a player I want to be able to see my bot working so I can improve it - create feedback loop pass by sprint 2
- Each players bot is identifiable. pass
- Each move is visible. pass
- Each players bot is always visible. pass
- The score for each bot is also visible. pass

2.5 - As a bot in the game, I want to collaborate and communicate with other bots so that we can work as a team - interactive
- There exists teams of bots. fail
- There exists multiple bots per team. fail
- Each bot has some way of communicating with other bots in the their team. fail
- Bots can act on information given by other bots. fail

2.6 - As a player, I want to see multiple teams of bots.
- There exists teams of bots. fail
- There exists multiple bots per team. fail
- Players can identify who's on their team, and who's on opposing teams. pass
- There are at least two teams. fail

2.7 - As a player, I want to get bots to work together as a team.
- Each bot has some way of communicating with other bots in the their team. fail
- Bots can act on information given by other bots. fail
- The core of an individual bot counts towards team score. pass

2.8 - As a player, I want to program my bot, so that I can control it
- Potential duplicate

2.9 - As a player, I want my programming language to be more accessible than a conventional one. pass by sprint 5
- The language is simple, with a limited syntax. pass
- The language is transparent, such that it is always obvious what is happening. pass
- The language is written in natural language. pass
- The language has very little abstraction. pass
- The language can have whitespace. pass
- Despite being simple the language should still be able to offer maximal functionality.
- It should resemble contemporary programming languages. pass

2.10 - As a player, I want to program functions so that I can make my bot functional.
- The programming language should allow for the creation of functions and methods. fail

B.4 Programming Logs

B.4.1 27/10

Programming Log	
Participants	Olivia
Date	27-10-2018
Time Started (nearest 15 mins)	12-00
Time Finished (nearest 15 mins)	15-00

Programming Objectives

- Create main running class
- Create a map class
- Print map to command line
- Create a bot class
- Randomly place bot in map

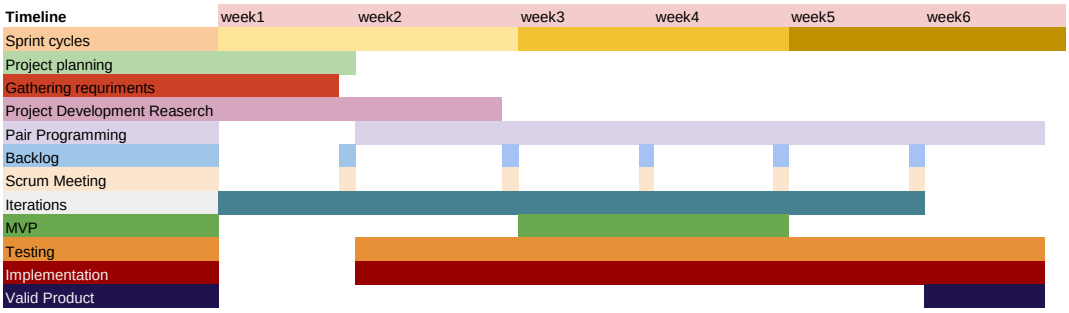
Outcomes

- All completed

B.5 Design Documents

B.6 Gantt Chart

B.6.1 26/10



B.7 Risk Assesment

B.7.1 29/10

Ca(s)tle massacre risk assessment

Risks identified:

No.	Date	Risk	Probab -ility	Impact	Possible effects	Risk Reduction Actions	Triggers & Actions
1	29.10.18	Loss of a team member: Due to illness Due to lack of commitment	High medium	**high**	Inability to complete tasks	Team meetings to keep the track of the morale's and situation within the group Encourage group work Self motivate	**Triggers** Reports of absence, or the lack of work from the team member. **Actions** Identify available resources. Relocate the work within the group. Keep the log of incident.
2	29.10.18	Changes in user requirements: Slight Significant	High Medium	Low Medium	Exceeding deadlines Increased work load	Verification against the detailed specification with the client during the meetings	**Triggers** Reports of the potential problems. **Actions** Discussion within the team and client depending on severity of the issue. Project iteration.
4	29.10.18	**Often changes/iterations of the code.**	**High**	**High**	Delays.	Prioritise tasks. Focus on functional requirements by consulting with the client.	**Triggers** Reports of not meeting the goals in time. **Actions** Potential redesign of the project. Reduction of project features. Focus on MVP. Resource relocation the most important task.
5	29.10.18	Lack of agreement within the team.	Medium	High	Inability to find a solution for the problem. Inability to complete some or all of the tasks.	Team meetings focused on communication between team members. Encourage of group work.	**Triggers** Informal problems with the tasks. **Actions** Informal communication between the team members to find a solution everyone will agree upon. Formal mediator from the University. Relocation to the task the team member agree with, Removal of a team member (last resort). Backlog of the incident.
7	29.10.18	**Lack of ability to iterate the project or implement customers desired functionality.**	High	High	Failure to produce the project to satisfy a customer.	Inform a client of inability to deliver some of the functions. Find an agreement.	**Triggers** Feedback from the client **Actions** Attempt to find a solution to implement required functionality. Finding help outside of the team members. Agreeing upon swapping some functionality for other.
8	16.10.18	**Failure to capture user requirements.**	**High**	**High**	Failure to achieve minimum requirements.	Recorded and transcribed interview with a client.	**Triggers** N/A **Actions** Recorded Interview.
9	29.10.18	**Lack of required skills within team members to create MVP.**	**High**	**High**	Inability to program/create the project.	Pair programming. Additional self-training.	**Triggers** Reports from the team member. **Actions** Relocate resources to help within the task. Additional help from outside of the team members (self-training).
11	29.10.18	**Poor code writing**	**High**	**High**	Code that doesn't work.	Pair programming to revise existing code.	**Triggers** System does not perform as required. **Actions** Rewriting the code.
13	29.10.18	Inability to present the MVP in time.	Medium	Low	Client will not be able to assess the existing progress.	Stick to the schedule.	**Triggers** **Actions** Monitor the activities against the schedule.
14	29.10.18	**Lack of resources to relocate in a case of emergency.**	**High**	**High**	Inability to create MVP on time.	Ensure the MVP is truly just MVP. Focus on the most basic functionality and build upon it.	**Triggers** Not meeting the scheduled goals. **Actions** Iterate the MVP. Cut off even more of the tasks.
15	29.10.18	Different schedules of team members. Inability to meet to work togheter.	Low/med ium	high	Poor cohesion. Erratic work schedule.	Monitor the tasks. Time management with all the team members in mind.	**Triggers** Reports of problems with the time schedule. **Actions** Focus on better time management. Find the time schedule that suits everyone.
16	29.10.18	Insufficient testing.	Medium	High	Hidden bugs.	Schedule enough sessions extensive testing through the team members and the client.	**Triggers** N/A **Actions** On-going unit and system testing.

B.8 CRC

B.8.1 26/10

environment	
collectable items able to move scene	bot board

b
add/deduct scores

player	
interactive control the bot	bot environment

bot	
able to move interactive able to attack/collect	environment board player

B.8.2 Game Visual - 24/10

B.9 Customer Interview Documents

B.9.1 Interview Transcript

Interview Transcript - 31/10

Michael: So with the previous versions that we presented to you, we said that we would quite like to add, you would quite like to have some programming functionality. So, we've gone away and researched what is feasible. We think we have come up with quite a nice way of doing it. So within our section what we're looking at doing is essentially creating a program that you compile. So as a user I go about, I click the different kinds of button in order to move my character. So I might click turn left, turn right and it would compile each different command within this box and then you click run. So if you're ever familiar with the turtles, it's a very similar...

Julian: Yup the first word that came into my head was, ''Turtles!''

M: So we're looking to do a very similar concept. In terms of actually developing the product its going to be the easiest one for us to implement and consequently we'll be able to deliver a better product ultimately. We would like to look at additional scope as well for further iterations. So we'd like to be able to look at bringing variables into this and also we'd like to look at adding loops as well to create further functionality. Although with this method it's unlikely to... it would be very difficult to create the full flexibility of having a proper programming language. It is...we believe it will be easier for the user and certainly to develop as well.

J: Okay.

M: So what we've done is, we've designed a current look of how we want our first product to look like, the phase one product. Alex will go through that and then after that we are going to describe the different phases we would like to go through in the iterations.

An: Okay so, what we have here is a bot which is able to move around by clicking the buttons on the right. And what he will do is to collect the coins which are spread randomly in the map. This game is turn based, so, erm, whoever collects the most coins in a given time, wins.

J: So who else is collecting the coins?

K: Yeah that's the iteration we think for the next week. This week we are trying to focus on how... the basics of the game, how its going to walk so we want to have a player and we want to have a bot that has some sort of resources like coins and we want to have some sort of way of showing how well the collection went and maybe create the first sort of architecture for the turn base. And if that works, hopefully it will, then think about how the multiplayer works. And that's probably going to be played on the same device. As Andy said, its going to be turn based and it will be the same amount of time. Maybe the coins going to be randomly spawning or maybe they're going to be in the same place – we don't know this yet. And basically that is the plan for this and next week.

M: So yeah just expand on that a little bit with our phases. So within phase two, which is what we're just describing is all about having the multiplayer. So we're just using the click buttons at this stage. And then once we've got past phase two and we've got two people going against each other, competing against each other, the next step, phase three, is to have people running their program as opposed to directly controlling their character.

J: Right...so the thing I'm not understanding yet is if we've got players running their programs, and I know you described using the stuff that was on the whiteboard, is that say we've got this scenario here, which way is the player facing? So if I wanted to get it to pick up the gold, that's one up and to the left, would I have to get it to turn left twice? For example?

M: Yes, that could be an option.

J: What I'm asking is, given where the player is now and where the gold is now, I wanted to make it go up one and left one... Because if I said forward at the moment it would go that way wouldn't it?

M: That way. So yes you click turn left, forward, turn left, forward. And this creates...

J: Then what?

M: And then you would want to try and pick up the other coins, so you might say turn right...

J: Yeah but how do you pick them up?

O: So when you move over a coin, you automatically pick them up.

J: Ah. Okay. Right, so there isn't a pick action?

M: No.

J: Okay.

O: This was discussed but the consensus was that you would just pick them up as you moved over them.

J: Okay, okay. Erm so that means that all of the perception part of this is achieved by the player viewing the map and working out for example the shortest path.

M: Yeah.

J: From where they are to picking up the gold. So if you had a multiplayer game and they both unleashed their bots at the same time then clearly a bot, I'm referring to this character thing there as a bot, may well being going to a place where there is no gold.

K: But its still turn based so you would wait for the first player to collect as many as they can, then you would be allowed to...

M: Well, sort of so so. The idea currently is that one player has a move. Then the next player has a move and so that's part of the challenge is to try to anticipate what the other player is going to do.

J: So when we talk about turn it is in fact one click of one button.

M: One click of a button, exactly. One click equals one turn and then we move to phase three with the programming then one move or one step in the program is a turn. And so part of the challenge is to anticipate what the other player is going to do.

J: Yes, okay...

M: If that makes sense?

J: Yes, yes it does. And that certainly ups the difficulty. However, it's still...okay maybe that is the right way to do it though.

M: And we feel, especially when we move onto our final phase with programming, then you can't anticipate what they're going to do halfway through you've got to try and think ahead of your opponent at the very beginning before you run the program which makes it that much more difficult.

J: It certainly makes it difficult because it means that you're essentially programming for something that might not happen.

M: Mm.

J: Which could be frustrating. Erm what I'm worrying about is the need for or the desire for the program to make decisions. There is no decision-making capability in the programming going on here. It only moves in turns.

M: So you what you would like to see is potentially, when you're actually running the program some kind of if-statement or conditional statement. So if this player was here, I do this, for instance.

J: That's what I'm thinking.

M: Okay, sure.

J: So that requires...that introduces then the capability for the language, or it means that the language has to capture the idea the bot can look and see what is in its...

M: Vicinity perhaps...

J: In the box yeah...

M: Err yeah, I think that would be something really nice to introduce but that is certainly something we have to look at for additional scope.

J: Oh yes it's something that comes...

M: Its later. I think we want to make sure we have our minimum viable product then we want to move through the phases to make sure that we build from the bottom up. So its something I think would be really nice to have, conditional things, but its something we can't, might not necessarily be able to deliver within our given time frame.

J: Yeah.

M: I hope you understand that.

J: Yeah, yeah because as it stands it means that because you've got a dynamic environment the program that you construct may no longer be fit for purpose when it comes to a certain phase of its execution.

M: For sure.

J: You know all about this because you did intelligent agents last year. Didn't you?

O: I did, yeah.

J: So I guess you know what I'm getting at?

O: Yeah. I'm not sure whether we as a team have the skillset to achieve something like that.

J: Yeah.

O: And this is the concern. I mean, we could definitely do it if we pushed ourselves to the limit but its weighing it up whether its worth pushing ourselves to the limit considering the other obligations we've got.

J: So it's a question of whether the fragility, the relative fragility of a fixed program against a changing environment provides enough of an interesting game play.

M: Sure. Sure.

J: Or whether you need to introduce the capacity for the entity to sense its environment and make decisions based on that.

M: Sure. Sure. Yeah, I don't think it…yeah. I agree that it would be a really good thing to do and I think we need to think about, I think we'll try and think about having the ability to implement it later. SO ensuring that our architecture is not designed so that we can't do it. But, erm yeah, I think I agree with Olivia here. Its something we leave until the end. For sure.

J: Well not necessarily the end but down the stream from phase one.

Appendix C

Sprint Three: 1/11-7/11

Return to chapter 3.

C.1 Meeting Minutes

C.1.1 02/11

Meeting - 2/11

Attendees: Olivia, Kat, Andrew, Andy (late), Michael (late)

Objectives for today's meeting:

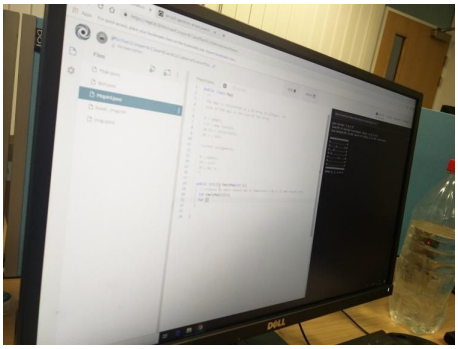
- Set all group members with working Eclipse on their preferred machines,
- Integrate Github with Eclipse software,
- Discussing the map/bot presentation in arrays (char vs int),
- Discussing next iterations of the game characteristic,

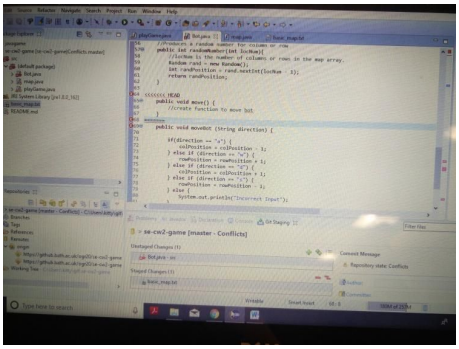
Programming:

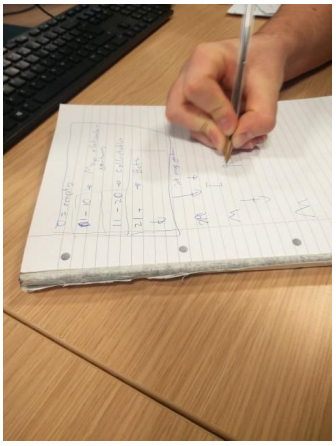
- First attempt of pair programing
- Discussing implementation of the game mechanics through the programming language;
    - Discussing the roles within the team; disusing the global variables;
    - Existing code explanation (bot, map)
- Get the bot moving around the map using wsad;
    - First attempt to program bot; discussing logic behind the code; input scanner function into the bot class; print the user instruction;
- Michael suggests major refactoring
    - Auto generator for the map, the method that saves the map and loads the map:
    - Creating class to represent a single map tile. To endure flexibility of the stored objects within the environment. Arrays with map features, collectibles and the bot. allows the different types of items to be in the same tile at the same time (ensures the ability to collect the coin)

Outcome:

- Refactoring the code.
- Next meeting on Monday.







C.1.2 06/11

Meeting - 06/11

Attendees - Andrew, Alex, Michael, Kat

Objectives for Today's Meeting

- Talking about the next interview with the customer
    - Showing the progress of the game
    - Asking for the feedback
- Agreeing on MVP

Sprint One Outcomes

- Researching the development of the game and how to implement it;
- First iteration of the project (from the programmable bots to more simple commands)
    - First code refactoring (from character based to integr based game)
    - Having a working environment for the game (map, bot and collectible);
    - Having a movable bot that a character can play using WSAD;
    - Having randomly positioned bot and collectibles for the players to interact with;
    - Ability to collect the collectibles;
    - Ability to win a game;

Aims for Sprint Two

- Scheduling the next meeting for the customer interview
- Looking into adding a new player(s)
- Looking at the game mechanics for the players to win

C.1.3 07/11

Meeting - 07/11

Attendees: Andy, Michael, Kat, Olivia

Objectives for today's meeting:

- Meeting with a customer
  - asking about the customers priorities
- Planning for the next pair programming session; considering two teams working on different tasks
- Discussing the way the player interacts with the bot(s)
  - interacting through the UI via buttons
  - commanding a bot through the pre-programmed command lines
- Discussing the game mechanics
  - programming how to do the players turns

Brainstorming Ideas

- commanding a bot at the beginning of each turn,
  - turn based with feedback
  - limited amount of commands per turn,
- Discussing the programing solutions for the how to command the bot;

Plan for Pair Programming

- Discussing programing 2 approaches at the same
  - present the two options for the customer with the intentions of keep just one:
  - 1 turn: 1 block of codes vs. multiple block of codes;

Outcome:

- Merging the game to the master branch
- Fixing a bug related to Unicode (display)
- Getting the working (running) MVP game
- Scheduling meetings for two programming teams

C.2 User Stories

C.2.1 01/11 - 07/11

SPRINT 3

3.1 - As a bot I want to be dynamic so i can move within the environment - make interface to program bot

3.2 - As a player, I want to know how multiplayers work in the game

3.3 - As a player, I want to know which way the player is facing so that I can decide how to move the bot - make the character consistent with bot movement

3.4 - As a player, I want to have a "pick up" action

3.5 - As a player, I want the bots to be unleashed at the same time

3.6 - As a player, I want the program to be capable of making decisions

3.7 - As a bot in the game, I want to read a dynamic environment so that I can make decisions based on that

3.8 - As a player, I want to program functions so that I can make my bot functional

3.9 - As a player, I want the game to be set in a castle

3.10 - As a player, I want to know which team is the best - scoreboard

3.11 - As a player, I want to move around a map - make a map

3.12 - As a player I want a bot so I can play the game - make a bot

3.13 - As a player I want to program bot movement - allow player to make changes to bot movement

3.14 - As a player, i want to plug in bots that can play for me

3.15 - As a bot in the game, I want to collaborate and communicate with other bots so that we can work as a team - interactive

3.16 - As a player, I want to get bots to work together as a team

3.18 - As a player, I want to program my bot so that I can control it

3.19 - As a player, I want my programming language to be more accessible than a conventional one

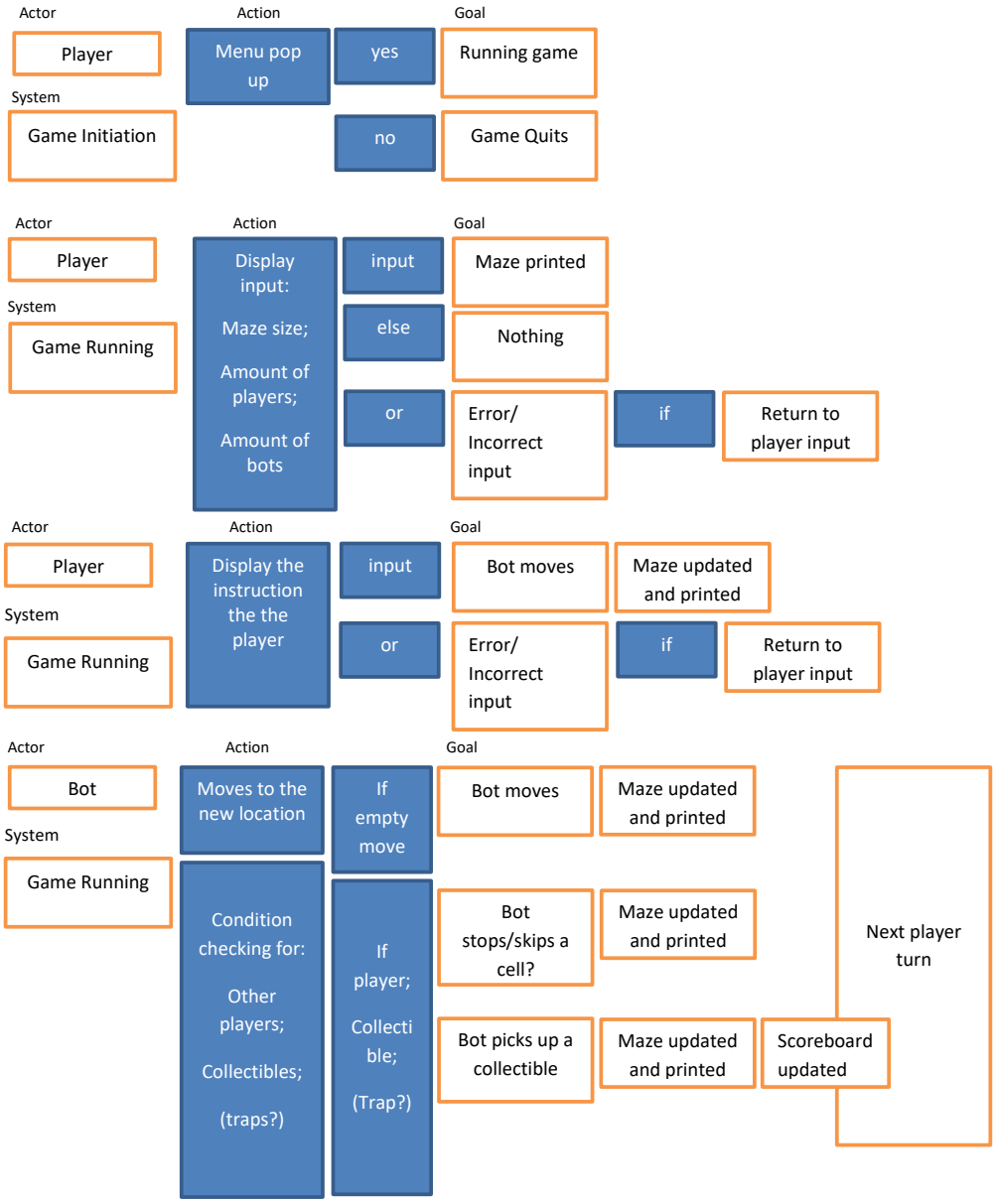
3.20 - As a player, I want to see multiple teams of bots

3.21 - As a boat I want to be interactive so i can interact with other bots/environment

Backlog - Sprint 3	**In Progress**	**Completed**
2.1 - as a bot I want to be dynamic so i can move withing the environment - make interface to program bot	1.1 - As a player , I want to move around a map - make a map	/
2.2 - As a player, I want to know how multiplayers work in the game	1.2 - As a player I want a the bot so I can play the game - make a bot	/
2.3 - As a player, I want to know which way the player is facing so that I can decide how to move the bot - make the character consistent with bot movement	1.3 - As a player I want to program bot movement - allow player to make changes to bot movement	/
2.4 - As a player, I want to have a "pick up" action	1.4 - As a boat I want to be interactive so i can interact with other bots/environment	/
2.5 - As a player, I want the bots to be unleashed at the same time	1.5 - As a player, i want to plug in bots that can play for me	/
2.6 - As a player, I want the program to be capable of making decisions	1.6 - As a bot in the game, I want to collaborate and communicate with other bots so that we can work as a team - interactive	/
2.7 - As a bot in the game, I want to read a dynamic environment so that I can make decisions based on that	1.7 - As a player, I want to see mutiple teams of bots	/
/	1.8 - As a player, I want to get bots to work together as a team	/
/	1.9 - As a player, I want to program my bot so that I can control it	/
/	1.10 - As a player, I want my programming language to be more accessible than a conventional one	/
/	1.11 - As a player, I want to program functions so that I can make my bot functional	/
/	1.12 - As a player, I want the game to be set in a castle	/
/	1.13 - As a player, I want to know which team is the best - scoreboard	/

C.3 Use Cases

C.3.1 29/10



C.4 Test Cases

C.4.1 1/11 7/11

SPRINT 3

3.1 - As a bot I want to be dynamic so i can move within the environment - make interface to program bot	pass
3.2 - As a player, I want to know how multiplayers work in the game	pass
3.3 - As a player, I want to know which way the player is facing so that I can decide how to move the bot - make the character consistent with bot movement	pass
3.4 - As a player, I want to have a "pick up" action	pass
3.5 - As a player, I want the bots to be unleashed at the same time	pass
3.6 - As a player, I want the program to be capable of making decisions	fail
3.7 - As a bot in the game, I want to read a dynamic environment so that I can make decisions based on that	pass
3.8 - As a player, I want to program functions so that I can make my bot functional	fail
3.9 - As a player I want to program bot movement - allow player to make changes to bot movement	pass
3.10 - As a player, i want to plug in bots that can play for me	pass
3.11 - As a bot in the game, I want to collaborate and communicate with other bots so that we can work as a team - interactive	fail
3.12 - As a player, I want to program my bot so that I can control it	pass
3.13 - As a player, I want my programming language to be more accessible than a conventional one	pass
3.14 - As a boat I want to be interactive so i can interact with other bots/environment	pass

3.1 - As a bot I want to be dynamic so i can move within the environment - make interface to program bot pass by sprint 3

- I want to make changes to the movement of bot in the middle of the game pass
- I can program the bot within a text area pass

3.2 - As a player, I want to know how multiplayers work in the game pass by sprint 3

- Different teams of bots are visible in the map pass
- There are different text area for players to program for their bots pass
- I want to know the scores of other players pass
- There are different ways to control the bot for different players pass

3.3 - As a player, I want to know which way the player is facing so that I can decide how to move the bot - make the character consistent with bot movement pass by sprint 6

- The bot is visible in the map pass
- The bot is graphically presented pass
- The bot has a head facing different directions pass

3.4 - As a player, I want to have a "pick up" action pass by sprint 3

- There are coins in the map pass
- The bot is dynamic pass
- The cell contains the coin pass
- There is a “pick up” button fail
- The player can decide whether to pick up a coin or not fail

3.5 - As a player, I want the bots to be unleashed at the same time pass by sprint 3

- There are at least two bots on the map fail
- The bots can be unleashed at the same time fail
- Do they all run at the same time pass

3.6 - As a player, I want the program to be capable of making decisions fail

- The bot can be programmed to make decisions fail
- The bot can scan the environment fail
- The bot can respond to other bots or coins fail

3.7 - As a bot in the game, I want to read a dynamic environment so that I can make decisions based on that

- **Duplicated**

3.8 - As a player, I want to program functions so that I can make my bot functional fail

- The programming language should be simple and powerful enough to program functions fail
- The programming language should be natural pass

3.9 - As a player I want to program bot movement - allow player to make changes to bot movement

- **Duplicated**
- I want to make changes to the movement of bot in the middle of the game
- I can program the bot within a text area

3.10 - As a player, i want to plug in bots that can play for me

- **Duplicated**
- There exists a bot
- The player can program the bot with a simple language
- The program contains the basic AI and is capable of decision making
- The player can make the bot do what he wants it to do

C.5 Programming Logs

C.5.1 02/11 - Part One

Programming Log	
Participants	Michael, Andy, Alex
Date	02-11-2018
Time Started (nearest 15 mins)	11:30
Time Finished (nearest 15 mins)	13:00

Programming Objectives

- Refactor the code to an map that contains a 3D array with map objects stored as integers

Outcomes

- Created a map tile object, which is contains integer numbers representing map objects, collectable and bots. This allows for an unlimited number of unique bots. Furthermore this allows for multiple kinds of map object and collectable.
- Made methods to both create and print arbitrary maps.

C.5.2 02/11 - Part Two

Programming Log	
Participants	Alex
Date	02-11-2018
Time Started (nearest 15 mins)	Alex to Confirm
Time Finished (nearest 15 mins)	Alex to Confirm

Programming Objectives

- Add a bot to a random location in the map that does not contain a coin.
- Add random coins to the map that do not contain a bot
- Create a scanner which reads input from the user in wasd terms.
- Allow the user to move the robot
- Print the map again

Outcomes

- All complete

C.5.3 05/11

Programming Log	
Participants	Olivia, Andy, Kat, Alex
Date	05-11-2018
Time Started (nearest 15 mins)	14-15
Time Finished (nearest 15 mins)	16-15

Programming Objectives

- Pair programming.
- Create a map with randomized collectibles and bot(s).
- Creating one method that recalls randomized numbers (x,y - location) and passes them towards the bots/collectible objects in order to spawn them inside of the array. The method checks if the assigned number within the array is a bot (with "for loop") or if it's an empty space, and assigns a collectible to this (empty) space. It does it for x times. (fixed problems with overwriting bot with the collectibles objects).
- Creating the method for the bot to move around the map. – this is done by taking the location coordinates of the bot, and storing it inside of the bot class.
- Creating a scanner function that translates the keyboard input and passes it to the location of the bot. .

Outcomes

- Created a randomised bot and collectible objects within the map environment.

C.5.4 06/11

Programming Log	
Participants	Michael
Date	06-11-2018
Time Started (nearest 15 mins)	10:00
Time Finished (nearest 15 mins)	12:15

Programming Objectives

- Refactor the previous version to create version 001
- Change to code to simplify it, removing .get methods, instead simply calling up variable.
- Modify code to allow for arbitrary numbers of bots and coins to be specified before the start of the game.

Outcomes

- All complete.
- Also allowed for arbitrary sized maps.
- Allowed for the map size, number of coins, and number of bots to be specified at the beginning of the game.
- Furthermore allowed for the game to run continuously, such that after finishing one game you can specify a new game or tell it to stop.
- Moved the program to version 001 for the end of the first code sprint.

C.5.5 07/11

Programming Log	
Participants	Michael Andy
Date	07-11-2018
Time Started (nearest 15 mins)	16:00
Time Finished (nearest 15 mins)	17:00

Programming Objectives

- Make a start on implementing an interpreter for the bots.

Outcomes

- Managed to work out exactly how to implement an interpreter, specifically in how to manage while loops and if statements.
- Worked out the syntax to use for the programming language to be interpreted.

C.6 Customer Interview Documents

C.6.1 Interview Transcript

Interview Transcript - 7/11

Michael: So we have no got a functional game. So at the moment this is very much kind of the next sprint but on this laptop we have a functional game. So, at the moment its only command line based, so our next steps for the next sprint will be to try and create a GUI and implement the basic running of code from the bots. So far we have managed to get a game which has a multiple... or an arbitrary number of bots and an arbitrary number of items to pick up, keeps a tally of your score and has an arbitrary size grid. Would you like to have a look?

Julian: Sure.

Olivia: Okay, I'll run the game again.... Are you happy to follow the instructions on the laptop?

J: Okay.

M: Do you want us to...

J: So I think I type 'y'.

M: And then, I would suggest ten.

J: Okay, that's a factor of ten smaller than I was thinking but never mind.

M: Err, two sounds reasonable. And then, maybe, 20 coins.

J: Okay.

M: And so here you can see at the moment we're representing the coins as a dollar sign. Then each bot has its number. So we've got bot one which is here and bot two which is here.

J: And where am I?

M: Well if you're bot one and I can be bot two if you like?

J: Oh I see, okay. Ah, I didn't realise that.

M: Yeah so if you want to have a quick go.

J: So I can do, a will take me...?

M: Down.

J: Down. And then do I have to...?

M: Enter, yeah, because unfortunately it's command line. Yeah so, ooh... So yeah, you move...

(Plays game)

M: They're working yeah.

J: Okay, fine. Thank you.

M: Thank you very much.

O: Thank you.

J: I didn't start the timer, so I'll start it at seven.

O: Okay, so that's the game in its current form. You know how it works, you've interacted with it. So our plan for the next sprint is to create a GUI for the game and simultaneously implement the idea of block commands.

J: Oh yes, this mechanism for kind of programming which we were talking about last week.

O: Yeah so, the idea is that we will split off and one team will create the GUI and one team will try and implement the basic code so we can pair program at the same time, which we haven't done yet. We've basically just had two sets; you know one pair programmer or pair programming unit programming at one time so we haven't had the opportunity two developments going on and then merging them back in. That's why we want to do it that way.

J: When we talked about the notion of constructing the programming last week, did we actually talk about...

M: Conditional statements?

J: We talked about that. We talked about trying to make it both more challenging but also possibly more sophisticated. The conditional statements obviously bring in one dimension. Another dimension is how far ahead you are able to program.

M: Sure.

J: How many steps should one allow the player to program? Because there is only a certain number before you lose track of what's going on...

O: Yeah, and you don't want to program...you don't want to allow a number of steps which means you win before your opponent has even had a chance to.

J: There's that but there is also the problem with the other player's bots also moving and we have to think about...we talked about this already, I think, last week.

M: Yeah.

J: About them colliding with them or trying to pick to pick up something...

M: I think...Sure. So our current...with regards to simply collision and picking up coins which have already been collected, these are actually solved in our current version of the game. If you collide, it simply skips to the next turn, if that makes sense? It's a bit harsh but its functional. And similarly with the picking up coins as soon as you move over the coin. The coin is picked up. So you can no longer pick it up in the next turn. Erm, I think we're exploring several options to have a look at how we can limit how far ahead you can program. One of them is to implement a turn limit. So you might run it for, a hundred turns for instance and after that the game finishes. And whoever's got the most coins within a given time then wins, in a given turn limit as such *(6.24)*

J: Does that mean as a programmer player I could put in a hundred loops?

O: Yeah, this is...I think this is what you mean, we need to program the number of moves that you're allowed per term as opposed to...

J: And then thinking, that could be as small as four or five. Because we can't, it isn't, well, it is a game tree.

M: Sure.

J: And so obviously the branching factor in that game tree depends upon how many players we've got in the space.

M: Sure.

J: And it doesn't take a very great depth of game tree before what you've worked out is completely useless. Because everything was changed too much.

M: Sure.

J: So it could be that, I'm not saying it should be, but the depth of the game, the number of moves you can turn ahead is going to be inversely proportional to the number of players you've got in the game. I'm just saying. I'm saying you should do that...

O: But it needs to be considered.

J: It's not necessarily linearly, inversely proportional, it could be by making that a...uh, a tunable factor,

M: Sure.

J: But that's a way of changing the difficulty aspect of the game.

M: So what you're sort of envisaging is that you would have, you would be able to go, each person would program a little bit of their bot, little bit of their bot. Run them and then after that step you can then review the game board again and then repeat.

J: And also another question would be which bot gets to go first. Because that is going to make a difference. Because the lower numbered bot always goes first. It's a bit like playing noughts and crosses, you know who's going to win, if you make the right move. So you make it...you make it random?

O: I think when we have a GUI, instead of being able to identify your bot by a number we could just have it as a colour and we could have some kind of random number generator behind the scenes which would randomly pick. So we could remove that number...

J: If one thinks about this from a player strategy point of view, knowing whether you get to go first or not would affect the programming of their bot.

O: True.

J: So it could be done randomly which means they would have to write a strategy neutral program. Or it could be that they know that this time they'll get the opportunity to go first and on another occasion they will get the opportunity to go later.

M: Sure.

J: And it might again effect the kind of program that they write and it becomes rather more chess like. You're thinking, if I do this and this and this and the other player does that and that and that, I get a good outcome. Or, oops I forgot about the opportunities they might have at this stage. That's what I mean about thinking through the game play and the game tree.

M: Sure, we could...equally we could feasibly have it that the players move in the same turn so you can have each within a given turn. The players move simultaneously.

J: That indeed is another option and so each one of them making a move and because of our condition rules, as long as they are precisely stated then it's not going to be a problem.

M: Sure, yeah.

J: Unless you…because there will be a predetermined conflict resolution. Okay?

O: Okay.

J: What would you do if one program is shorter than the other program?

O: I was thinking that you would have an upper limit of the amount of moves or actions you can take. And then, that is almost part of your tactics, how many you take but we haven't actually discussed that yet.

J: Another way of playing it could be you play all your moves in one go rather than each one doing one move.

M: So you might say move forward five and then in a singular turn moves all five.

J: Yeah.

M: Okay.

J: I'm not saying it's a good idea.

M: Sure.

J: I don't know if it's a good idea because trying to evaluate these ideas in the space of a short meeting like this, is not going to work well. One might have an intuition about it, but intuition? Do I trust that? You'll only know if it improves the playability of the game when you go through a number of scenarios. One doesn't necessarily make decisions based on well we better stick with because the customer said so. It's better to do it this way because it makes the game more interesting.

M: Sure.

J: Because that's what you're trying to do make an interesting game.

O: In an agile way.

J: Yes. Okay, I think we're done. We've run out of time. Any more questions?

Appendix D

Sprint Four: 8/11-14/11

Return to chapter 4.

D.1 Meeting Minutes

D.1.1 09/11

Meeting Minutes - 09/11

Attendees: Andy, Kat, Olivia, Alex

We have come up with a list of documentation tasks/questions to address

- Olivia is going to transcribe the interviews
- Kat has made a UML diagram
- Ask for clarification on user stories - Wednesday lab
    - Improve our current user stories
- Prepare some testing documentation
- Clarify documentation formatting - Wednesday lab

Outcomes:

- Next meeting at 2:15 12/11
    - Share coding progress
        - Maybe think about which route but we might make the decision on wednesday
    - Go through all current documentation.

D.1.2 13/11

Meeting Minutes - 13/11

Attendees: Olivia, Kat, Michael, Andy, Alex

Organisational

- Creating user stories based on interviews with the customer and game requirements
- Group analysis of interview Transcripts - Andy to write up user stories
- Summary of discussion
    - Gameplay of bot/bots; programmable and/or turn taking
    - They way of interacting with the player and bot; through the series of commands or some blocks of code
    - Feedback/scoring; who wins and how – team or a player
    - Building procedures in programming script; programming language or script
    - Difficulty of the game; how to “step up” a difficulty –“planning for something that might not happen”
    - How far ahead are you able to program the bot (related to turn taking), limiting the amount of moves ahead depending on the amount of player. Giving a player an upper limit but can not use all the moves at once
    - Ability of the decision making for the bots and game (using AI); scanning for the resources (challenging and more sophisticated)
    - A fixed program vs a dynamic environment;
    - The size of the environment and the ability to identify the bot straight away (mostly related to graphic)
    - Instruction for the player of how to play the game; tutorials vs manual
    - Mechanics for the game: how many programmable loops
    - Which bot gets to get first – affects who wins. Make a strategy neutral program that chose who goes first.

Next Meeting

- Tomorrow 14/11 14:15

D.1.3 14/11

Meeting Minutes - 14/11

Attendees: Olivia, Kat, Michael, Andy, Alex

Meeting with a customer:

- Clarification for the document processing (report writing)- narrative writing or the putting the logs together
- How the customer feels about the game so far – break up of the game play
- What kind of guidance the player want (how to prevent player from writing a “wrong” program vs how do I do what I want to do)
- Discussing the potential iteration of the game mechanics as a last possible time slot
- Speeding up the game by blocks of commands for both of the players (keeping the player engaged)

Improvements to Documentation

- Updating the Gantt chart
- Updating risk assessment
- TESTING

Next Iteration

- User interface,
- Potential teams of bots,
- Guidance for the player
- Discussing the likeart scale for the customer to rate our product

Next meeting:

- Tomorrow 15/11 2:15

D.2 User Stories

D.2.1 8/110 14/11

SPRINT 4

4.1 - As a player, i want to interact with a large map

4.2 - As a player, I want to know where the bot is

4.3 - As a player, I want to have a clear instruction of how to play the game

4.4 - As a player, I want the game to be more challenging and sophisticated - conditional statement

4.5 - As a player, I want the game to set some limit to what players can do.

4.6 - As a player, I want the number of players to be proportional to the number of steps one player can take.

4.7 - As player, I want the number of move changes as level goes up - change numbers of turns

4.8 - As a player, I want it to be random that who goes first so that it can't affect my strategy

4.9 - As a player, I want to play all my moves in one go

4.0 - As a bot I want to be dynamic so i can move within the environment - make interface to program bot

4.11 - As a player, I want to move around a map - make a map

4.12 - As a boat I want to be interactive so i can interact with other bots/environment

4.13 - As a player I want to program bot movement - allow player to make changes to bot movement

4.14 - As a player, I want the program to be capable of making decisions

4.15 - As a player I want a bot so I can play the game - make a bot

4.16 - As a player, i want to plug in bots that can play for me

4.17 - As a player, I want to know which team is the best - scoreboard

4.18 - As a player, I want to program my bot so that I can control it

4.19 - As a player, I want to have a "pick up" action

Backlog - Sprint 4	**In Progress**	**Completed**
3.1 - As a player, i want to interact with a large map	2.1 - As a bot I want to be dynamic so i can move withing the environment - make interface to program bot	1.1 - As a player , I want to move around a map - make a map
3.2 - As a player, I want to know where the bot is	2.6 - As a player, I want the program to be capable of making decisions	1.2 - As a player I want a the bot so I can play the game - make a bot
3.3 - As a player, I want to have a clear instruction of how to play the game	1.4 - As a boat I want to be interactive so i can interact with other bots/environment	1.3 - As a player I want to program bot movement - allow player to make changes to bot movement
3.4 - As a player, I want the game to be more challenging and sophisticated - conditional statement	1.5 - As a player, i want to plug in bots that can play for me	1.13 - As a player, I want to know which team is the best - scoreboard
3.5 - As a player, I want the game to set some limit to what players can do.	1.9 - As a player, I want to program my bot so that I can control it	2.4 - As a player, I want to have a "pick up" action
3.6 - As a player, I want the number of players to be proportional to the number of steps one player can take.	/	/
3.7 - As player, I want the number of move changes as level goes up - change numbers of turns	/	/
3.8 - As a player, I want it to be random that who goes first so that it can't affect my strategy	/	/
3.9 - As a player, I want to play all my moves in one go	/	/

D.3 Test Cases

D.3.1 8/11 14/11

SPRINT 4

4.1 - As a player, i want to interact with a large map	pass
4.2 - As a player, I want to know where the bot is	pass
4.3 - As a player, I want to have a clear instruction of how to play the game	pass
4.4 - As a player, I want the game to be more challenging and sophisticated - conditional statement	pass
4.5 - As a player, I want the game to set some limit to what players can do.	pass
4.6 - As a player, I want the number of players to be proportional to the number of steps one player can take.	pass
4.7 - As player, I want the number of move changes as level goes up - change numbers of turns	fail
4.8 - As a player, I want the program to be capable of making decisions	fail
4.9 - As a player, i want to plug in bots that can play for me	pass

D.4 Test Cases

4.1 - As a player, i want to interact with a large map. pass by sprint 5 with GUI
- There is a large map, over 40 tiles square. pass by version1
- The player can move the bots about the map. pass by version1
- The is a way in which the player can still see their bot, even if each tile appears very small. pass by version GUI.
- The map zooms in to players if it is too large. fail

4.2 - As a player, I want to know where the bot is. pass by sprint 5 with GUI
- The player can easily distinguish which bot is theirs. pass by version1
- Even in a large map it is still obvious where the player's bot is. pass by version1

4.3 - As a player, I want to have a clear instruction of how to play the game
- The game is largely self explanatory. fail
- There is a manual on how to program. fail
- There is a manual on how to play the game. fail
- There is a video tutorial. fail
- There are example programs that can be cut and paste. pass by version2

4.4 - As a player, I want the game to be more challenging and sophisticated - conditional statement. Pass by sprint 5 (merging versions with GUI)
- There exists in the programming language conditional if statements such that bots can react to their environment. pass
- There is a conditional "if bot in tile" pass
- There is a conditional "if wall in tile" pass
- There is a conditional "if coin in tile" pass

4.5 - As a player, I want the game to set some limit to what players can do.
- There is a syntax checker on the code to prevent players creating erroneous code. fail
- There is a loop limit to prevent players code entering an infinite loop. fail
- The loop limit has been set to something high enough to prevent it becoming restrictive, but low enough to stop the program from becoming too slow. fail
- If the syntax is wrong, the game says so. fail
- If there is an infinite loop, the game says so. fail

4.6 - As a player, I want the number of players to be proportional to the number of steps one player can take.
- The less players there are, the greater the number of moves between modifying the code. fail
- This is controlled by a factor. fail

4.7 - As player, I want the number of move changes as level goes up - change numbers of turns
- The number of moves between modifying the code is controllable to increase the difficulty. fail

4.8 - As a player, I want the program to be capable of making decisions
- The programming language allows for conditional if statements. pass by version2
- "if bot in tile" fail
- "if wall in tile" fail
- "if coin in tile" fail
- "if var1 = var2" pass by version2
- "if var1 < var2" pass by version2
- "if var1 > var2" pass by version2

4.9 - As a player, i want to plug in bots that can play for me. pass by sprint 5
- There exists programmable bots. pass by version2
- These bots play on the players behalf in accordance with their code. pass by version2

D.5 Programming Logs

D.5.1 08/11

Programming Log	
Participants	Kat, Olivia
Date	08-11-2018
Time Started (nearest 15 mins)	09-15
Time Finished (nearest 15 mins)	10-15

Programming Objectives

- Programming a block of x number of action for the player to take every turn;
- Getting commands split and stored in order to run command line;
- Adding "quit" option for the game; (fixing problems with printing another map after the command quit was entered and game finished);

Outcomes

- All the objectives achieved.

D.5.2 09/11

Programming Log	
Participants	Michael
Date	09-11-2018
Time Started (nearest 15 mins)	14:30
Time Finished (nearest 15 mins)	18:30

Programming Objectives

- Implement an initial interpreter for the language.
- Create a method to read files containing code.

Outcomes

- Created a class called BotProgram.
- Created a method to read program files and store them in a 2D array.
- Created the Code object holding the array and the Variables object which is a dictionary of variables.
- Implemented the basic functions of commands and variable allocation and manipulation.
- Implemented a .next() method within the BotProgram object, this makes the code run until the next command is seen, at which point this is returned by the function.

D.5.3 10/11

Programming Log	
Participants	Michael
Date	10-11-2018
Time Started (nearest 15 mins)	10-00
Time Finished (nearest 15 mins)	12-30

Programming Objectives

- Implement loops.
- Create a print statement.

Outcomes

- Made good progress on implementing loops.
- Created a basic print statement which prints out text written after it.

D.5.4 11/11

Programming Log	
Participants	Michael
Date	11-11-2018
Time Started (nearest 15 mins)	11-00
Time Finished (nearest 15 mins)	13-00

Programming Objectives

- Complete loops.
- Add if statements.

Outcomes

- Managed to implement while loops with different inequalities, specifically with =, > and <.
- Managed to also implement if statements in a similar vein, which check to see if a variable satisfies the aforementioned inequalities.
- Managed to implement both print and printvar statements.
- Managed to implement comments; lines beginning with # are ignored.
- Allowed enter line breaks; empty lines are ignored by the interpreter.

D.5.5 12/11

Programming Log	
Participants	Andy, Michael
Date	12-11-2018
Time Started (nearest 15 mins)	15-00
Time Finished (nearest 15 mins)	16-45

Programming Objectives

- Create maps where the bots start in a specified position, and the map is of a specified size.. This is so that there is consistency allowing meaningful programs to be made.
- Integrate the interpreter with the main program to allow for bots to be moved using programs written by the users.
- Allow if statements relating to the bots, for example “if bot in tile upper left, move right”.
- Avoid refactors for the time being, simply replace human input by program input.

Outcomes

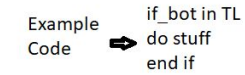
- Deprecated the AddBots function (this creates an arbitrary number of bots in a map) and replaced it with SpecBots (this creates two bots in specific locations, precisely the upper left corner and the lower right corner).
- Permanently set the number of bots to 2. Have changed as little as possible to avoid conflicts.
- Andy has permanently set the map size to 12 because he's awesome.
- Each turn is now simply initiated by hitting enter.
- The program is now running based on two code files for each bot.
- Question, can we have a mp that loops back onto itself?? For example a torus.
- Have now made a few example programs to run the game on!!

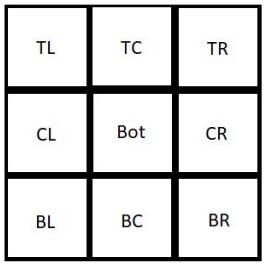
D.5.6 13/11

Programming Log	
Participants	Michael, Andy, Alex
Date	13-11-2018
Time Started (nearest 15 mins)	17:15
Time Finished (nearest 15 mins)	hh-mm

Programming Objectives

- Implement some kind of conditional statement to detect other bots in the vicinity.
- Potentially allow the user to assume direct control in these scenarios?? Suggested by Andy
- Elaborating on andy perhaps have a par system which means every time the user assumes direct control track of score.





Outcomes

- Added conditions to the simple if statement to ensure that it is only called up when allowed.
- The .next() function now has the mainmap passed to it.
- Made some progress into implementing the if bot statements.

D.5.7 14/11

Programming Log	
Participants	Michael
Date	14-11-2018
Time Started (nearest 15 mins)	11:00
Time Finished (nearest 15 mins)	13:00

Programming Objectives

- Complete the implementation of if bot statements.

Outcomes

- Completed implementation of if bot statements.
- Added an idle statement.
- Implemented the allowing of whitespace before statements.
- It now not only treats all whitespace as part of the split function, it also removes any whitespace before a function.
- To remove whitespace from before a function, I implemented a separate function for this.

D.6 Customer Interview Documents

D.6.1 Interview Transcript

Interview Transcript – 14/11

Michael: Sure. So, we come to you with a much more developed game this time. So, what we've done is we've run, currently we've run with the idea of actually being able to write programs to run our robot from. So, this is currently what we have.

Julian: Okay.

M: And we have, with the language which we've created, we've got a pretty...err...complete set of functions so you can do lots of things. Erm, including variables...erm variables, loops, if-statements, conditionals, comments, enters the whole lot. So...

J: From where did you get this language?

M: We made it up on the spot to try and be nice to work with and to write, easy to read and easy for the program to interpret it.

J: So, how did you write the language processor?

M: Erm, made an interpreter. So, this is a full-blown interpreter within Java. And the nice thing about having an interpreter is that it means we can change our programs on the fly. So that each time you start a new game, you don't have to restart the whole program.

J: Okay.

M: So, if you'd like to have a look...have a look at it in operation. So, we've got a program for our player one and we have a program for player two. And we've now set the size and the location of the robots in the game. And so, we've set the robots, one to the top left and one to the bottom right. At the moment we've got a very small map, this is just so that it is easier to demonstrate but in reality, you would have a bigger map to make it more interesting. So, each robot performs one command per turn. So, the current one, this one moving up, and the bot number one has been programmed to scan along each line in a big zig-zag. So, at the moment it is zig-zagging. Now here it runs, the first time, this program currently runs into a problem because it hits two and it can't move anywhere because two is blocking its way. So, this we use to demonstrate another feature of out program which is that currently it can either finish when all the coins are picked up or finish when the turn limit is reached, which is currently set to 125. Erm, would like...that is one of the things we would like to do is make the turn limit be able to change. So, it then finishes at a turn limit. Now what I can do is I can take this one here and I can do a very Blue Peter moment and do a here's one I prepared earlier. So, I can change the file name, err change that to 'c' for instance and change this to player one. Then because its interpreted, without stopping the program, I can now play another game here and it will run. This particular program shows another, nice feature of the language we've got which is it allows the robots to detect using conditional if statements, if there is another bot in front of it or nearby. This time instead of getting stuck, as long as I've loaded up the right program, it should now detect there is a bot nearby, print "I've seen a bot nearby" to the command line and is now actively avoiding the previous robot. And it does it in the other direction too. So, we've also implemented looking for coins as well so you can program the robot to see coins. And this is the current program that's doing that so we've got quite a lot of...its quite a complex lang..., there's a lot of things you can do but its easy to get started in a simple way.

J: Okay.

M: Our vision for the future s to, or our vision for our finished program would be to wrap this in a user interface rather than having it command line based. Erm, to have automatic running features so

you're not having to press enter every time, although you would be able to if you chose to do that so you can diagnose your robot. And be able to control teams of robots so you would have maybe two robots per team.

J: So, you can have more?

M: Yeah, you can have more. So, you might have two robots on one team and two robots on another team, and you write a script for each. And then they run, together.

J: Yup. Erm, so, presumably you're going to see with part of the interface when the number of rows and columns gets larger, what's going to happen?

M: Erm...

J: Presuming it goes outside the realms of the box.

M: This is why we need to add a wrapper on the, add a UI wrapper onto this because it's a limitation of the command line, it's literally just printing out each line...

J: Sure. Yes, I know that. It's a good solution for a first version. So, I'm asking, what's the final solution going to be on that?

M: Err, as in how...

Olivia: So, when you start programming at the beginning of the game and your faced with the map and you need to program your bot, the bit that you will see on the screen won't be the whole thing but the plan would be to have scroll bars so you could the rest of the map. And then, once you run the program then you'll see the bot move and the screen will move with the bot.

J: Okay, good. So that's part of your plan for function?

O: Yes.

J: And err, with any luck you'll get there as well. Okay, that's good. Erm, so, what problems do you see?

O: Well, there are a few things we need to address. We wanted to know what your opinion was on the current game play, how you play the game. So, at the moment, as you know, you program at the beginning, both players program at the beginning and then the game starts. The players passively watch.

J: Mmm.

O: So, the active engagement is at the beginning only.

J: Exactly.

O: It could be that this is changed so that you can improve your program, in the middle of the game. You have x number of moves and then both players take a moment to...

J: Yes, I can...because I can see that, as you were pointing out, it could become a little dull just watching two things progress.

O: Yes.

J: Whereas you might want to see how the behaviour needs to change part way through and so you'd like to go and do something about that.

O: It might be as well that you look at how your bot is moving and think, 'Yeah, I'm actually happy with how it's moving' and you don't have to change it. It might be good to break up the game play so that you actually feel like you're playing it.

J: Yes, that's right, erm yep. As developing the way my bot works, erm, do you think that the current presentation of what goes on is the best way to do that or do you think something faster might be desirable. It could be that the issue you're concerned about is the end game and so you've got to run it god knows how many times in order to get to move 125 or whatever to see what happens.

O: I think that it would probably be more enjoyable if it wasn't so much one person takes a turn, the next person takes a turn. That could speed up in some way. Maybe, erm, instead of the player pressing enter or however the game progresses you could have…see each bot take five moves.

J: Yes, okay.

O: So, you can see them both move at the same time or not at the same time. One moves…then you see that in a faster block, you don't have to sit and press enter through all. The other thing you could do is have it where you take more than one move. So, player one does his five moves, player two does their five moves and then you see them move…

J: And even run to completion.

M: Yeah.

O: I think just to keep the player engaged.

J: Yes.

O: But like, I think you said in the last meeting, we need to try out these scenarios to see which one is best for the player.

J: Mhm, yes.

M: Equally, you feasibly have it so that you can do skip to turn x so it automatically runs at the fastest speed java can run to turn x and then it displays whatever…so lots of options.

O: The other question we wanted to ask was what kind of guidance as a player would you like with regards to the programming. So, how do you want the instructions to be displayed?

J: Well I suppose that, we're talking about some form of…, although his is a game we're talking about a form of IDE. So that means, how do you…what are you going to do about people writing the wrong program or how you prevent them from writing a syntactically incorrect program. Given that the language syntax is quite limited, which is a good thing, that possibly makes it easier to produce a constrained IDE. So essentially there is now way to write a syntactically incorrect program. Which leaves the programmer player only having to deal with the problem it will deal with which is how do I make it do what I want it to do. Erm, and so, erm, so that's the situation I think one wants to be in.

O: Okay.

J: Does that make sense?

O: Yes.

J: Erm...

M: Do you like the fact it is a programming interface game?

J: This is what we talked about in the first week and its definitely different.

M: Yeah.

J: And so, it's interesting to see how that idea is evolving.

M: Mm.

J: Erm, how is your documentation going?

O: We were gonna ask you about this. So, we've got documentation in the sense that we've got meeting minutes and we've got programming logs and we've got some user stories and we've got some design documents. We need to improve upon our testing. Something that we're finding difficult is how to collate it together. Should there be a string of narrative linking them all, or, I know you said about cross referencing, in the lecture.

J: Mhm.

O: We're finding it difficult to decide how to do it. It does feel like they're quite separate and because we don't want to write out, 'We did this and we did this and then we did this' ...

J: No, right... So, you've got all the... It sounds like what you've got is a good number of the artifacts that are documenting the process and you said the collation is the issue?

O: Yes.

J: So, it's a question of what...of how to structure those things together and what I've suggested, I think it was in the, it was that you should do it week by week. I'm sure you know when you've created these things, which sprint the documents are part of.

O: Okay.

J: So, think of it as essentially being like chapters where each sprint is a chapter.

O: Like a weekly diary?

J: Yeah.

O: Okay.

J: Yeah, and so, then within each one of those sprint chapters, have the same structure every week because that makes it easier for me to find my way around. Make sure the sections are numbered so that just if I open a random page, I know which week it is.

O: Yeah.

J: I say open, I'm going to be looking at this on a screen. Hyperlinks would be a good idea as well.

O: Okay.

J: Hmm, yes. And, err, obviously there should be a table of contents at the beginning, hyperlinked. And for each week, there also should be some overview narrative. I think I wrote about this in one of the, I think its actually in the coursework specification. So, you need something like, what happened this week, maybe what the highlights were, what things didn't work and what did work so you can

link from that introduction to the content in this week. And then it should be quite easy going through those sprints, to see that ,'oh their functional increment was achieved this week, this is what the backlog looked like...' Then in your sprint meetings, 'Look we're going to prioritise this one' and then I'll go to the next week and lo and behold, there it is.

O: Okay that makes, that helps, I think.

J: I hope its just a matter of organising what you've got.

O: Yeah, that was the thing. We were looking at like, 'How, how is this going to go together? How is it going to make sense?'. For me, weekly diary, overview; I can suddenly see how this comes together.

J: Right. And that's me trying to put the smallest overhead I can on the production of the documentation. Because as you know documentation is not an important part of agile methods.

O: No.

J: But I need to have something...

O: You need to have evidence...

J: ...to assess.

Appendix E

Sprint Five: 15/11-21

Return to chapter 5.

E.1 Meeting Minutes

E.1.1 15/11

Meeting Minutes - 15/11

Attendees: Olivia, Kat, Michael, Andy, Alex

Review of Interview Transcript 14/11

- Talking about the speed of the game,
- The amount of bots in a team,
- Instructions to be displayed,
- Feedback from the game (as in incorrect code or make the bots do what the player wants),
- UI

Reorganising Documentation

- Ensuring user stories are associated with the right sprint
    - Week 1:
        - Make a map for a player
        - Playable bot
        - Movable bot
        - Controllable bot
        - Programming language
        - Functional bot
        - Environment
        - Scoreboard
    - Week 2:
        - Collaboration
        - Multiplayer
        - Pick up action
        - Timing of the bots
        - Decision making
        - Dynamic env
    - Week 3:
        - Graphical clue
        - Large map
        - Where the bot is
        - Clear instruction
        - Challenging and sophisticated
        - Set a limit for the player
        - Proportional amount of steps
        - Number of turns change
        - Moves on the go
        - Strategy

Integrated testing

Next Meeting:

- Tues 20/11 16:15
- Prioritise GUI for this sprint

E.1.2 20/11

Meeting Minutes - 20/11

Attendees: Kat, Michael

Organisational

- Planning testing
- Progress of the iteration
    - Working GUI

Meeting with the customer:

- Showing a GUI

Plans for next iteration:

- More UI
- Add multiple bots for a team
- Turn based

E.1.3 21/11

Meeting Minutes - 21/11

Attendees: Kat, Michael, Alex, Andy

Planning testing

- User case testing using the trello board

Meeting with the customer:

- Showing a GUI through the executable java file;
- Feedback

Plans for next iteration

- Merging the versions together

E.2 User Stories

E.2.1 15/11 21/11

SPRINT 5

5.1 - As a player, I want to have more than 2 bots per team in the game

5.2 - As a player, I want to know how I can see the rest of the map if the map goes out of the box

5.3 - As a player, I want to improve my program in the middle of the game so that I can adjust my strategy.

5.4 - As a player, I want to skip to a certain turn so that I can speed up or slow down my game

5.5 - As a player, I want to program the game like IDE

5.6 - As a player, I only want to know how I make the bot do what I want it to do

5.7 - As a player, I want to know where I am doing wrong in the program

5.8 - As a player, I want to see my screen move with the bot

5.9 - As a player, I want to program my bot so that I can control it

5.10 - as a bot I want to be dynamic so i can move within the environment - make interface to program bot

5.11- As a player, I want the program to be capable of making decisions

5.12 - as i boat i want to be interactive so i can interact with other bots/environment

5.13 - As a player, i want to plug in bots that can play for me

5.14 - As a player, I want to know where the bot is

Backlog - Sprint 5	**In Progress**	**Completed**
4.1 - As a player, I want to have more than 2 bots per team in the game	3.2 - As a player, I want to know where the bot is	2.1 - as a bot I want to be dynamic so i can move withing the environment - make interface to program bot
4.2 - As a player, I want to know how I can see the rest of the map if the map gose out of the box	/	2.6 - As a player, I want the program to be capable of making decisions
4.3 - As a player, I want to improve my program in the middle of the game so that I can adjust my strategy.	/	1.4 - as i boat i want to be interactive so i can interact with other bots/environment
4.4 - As a player, I want to skip to a certain turn so that I can speed up or slow down my game	/	1.5 - As a player, i want to plug in bots that can play for me
4.5 - As a player, I want to program the game like IDE	/	1.9 - As a player, I want to program my bot so that I can control it
4.6 - As a player, I only want to know how I make the bot do what I want it to do	/	/
4.7 - As a player, I want to know where I am doing wrong in the program	/	/
4.8 - As a player, I want to see my screen move with the bot	/	/
/	/	/

E.3 Test Cases

SPRINT 5

E.3.1 15/11 21/11

5.1 - As a player, I want to have more than 2 bots per team in the game	fail
5.2 - As a player, I want to know how I can see the rest of the map if the map goes out of the box	pass
5.3 - As a player, I want to improve my program in the middle of the game so that I can adjust my strategy.	pass
5.4 - As a player, I want to skip to a certain turn so that I can speed up or slow down my game	pass
5.5 - As a player, I want to program the game like IDE	pass
5.6 - As a player, I only want to know how I make the bot do what I want it to do	pass
5.7 - As a player, I want to know where I am doing wrong in the program	fail
5.8 - As a player, I want to see my screen move with the bot	pass
5.9- As a player, I want the program to be capable of making decisions	fail

5.1 - As a player, I want to have more than 2 bots per team in the game . pass by sprint 1
- There are multiple bots per team pass
- The bots in each team is distinguishable pass

5.2 - As a player, I want to know how I can see the rest of the map if the map goes out of the box pass by sprint 5
- There is a working scroller pass
- The screen of the game has a fixed size pass

5.3 - As a player, I want to improve my program in the middle of the game so that I can adjust my strategy. pass
- I can always see what my bot is doing clearly during the game pass
- My script for my bot should be saved somewhere and I could see it and change it anytime I want. fail
- I can skip in the middle of a turn fail

5.4 - As a player, I want to skip to a certain turn so that I can speed up or slow down my game fail
- There exits a skip button fail
- The game could allow bots to run more than 1 turn in one breath if the players want. fail
- The game can move on to one certain turn if the players input the turn number. fail

5.5 - As a player, I want to program the game like IDE pass
- The game resembles an IDE. pass
- The game has a code editor. pass
- The game has a command line output pass
- The game has a screen to view the bot's movement. pass

5.6 - As a player, I only want to know how I make the bot do what I want it to do fail
- There is a simple and powerful language to program the bot fail
- The player can program functions for the bot fail
- The player can program the bot to move or read environment and make decisions pass

5.7 - As a player, I want to know where I am doing wrong in the program fail
- There is a tutorial before starting showing how to program your bot. fail
- There are tips showing how to improve your program fail
- There is a syntax library for player to check fail
- There are some pre-written code provided for players if they don't know how to program. pass

5.8 - As a player, I want to see my screen move with the bot pass with GUI
- There exists a larger map than the screen of the game pass
- The program needs to have the scrolling function. pass
- A min-map is needed to show the location of every bot ,for a screen is not big enough to show all the bots clearly. pass

5.9- As a player, I want the program to be capable of making decisions
- **Duplicated**
- The bot can read the react to the environment FAIL

E.4 Programming Logs

E.4.1 16/11

Programming Log	
Participants	Alex, Andy
Date	16-11-2018
Time Started (nearest 15 mins)	13:00
Time Finished (nearest 15 mins)	16:30

Programming Objectives

- Create a GUI window to allow users to decide the size of map and the number of coins and bots by themselves.
- Create a GUI window for the real game to allow users to input their command by click the buttons or keyboard rather than command line.
- Figure out a reasonable layout for all the components of GUI to make the interface use- friendly.
- Rewrite part of previous code to meet the syntax for GUI.

Outcomes

- All complete

E.5 Customer Interview Documents

E.5.1 Interview Transcript

Interview Transcript - 21/11

Michael: Ok so, we are just gonna quickly run through the stage that current progress we are at the moment. So currently, we've got version one which was a nonprogramming bot version of the game, but it is not got a GUI, whereas version 2 was still as it was last week, as shows last week provide GUI the reason for this is that we have 2 teams working on different parts of project, and so GUI for version one now became ready to display. So the next scope is to move onto version 3 where we merging the GUI before we discuss the next version. The version one which now has g GUI.

OK. Julian was playing the game…

Michael: That's the current version one that one was showing quite a lot of bots so next one step is going to version 3 so we will be putting on a GUI to what we demonstrated to you last week, which was with the programming bot functionality, we shall, you shall have the ability to edit code after 20 turns. So the game will pause you can change your program, and then it will resume again, you have the option.

Julian: And then How you, you probably going to need to evaluate that 20 turns.

Michael: Yeah, we will fix it for the time being, but with the…just to make programing of this .and so, we will have the option to all to run them, so that you don't have click next go through very quickly. And it will have fixed map size with 2 bots, then moving forwards, and merge the version for beyond, and it's going to be large gather the bot, however you have the option to have multiple bots per team, you have scrolling map, and arbitrary map size, eh, and you will able to scale the number of turns to the map size, and scale the number of turns between more than modifying the code so you are be able to change whether its 20, 40 turns, 50 turns and so forth.

Julian: Do you think that ought to be actually something that is, well, eh…if that.. that can be could be fixed externally, but it could also be that it something that is part of the way the game works, perhaps, the number of turns is an extra dimension to the playing, so that it makes it, you could plan 20 turns ahead, sometimes you find oh well I should go to change your strategy after 10.

Michael: Ok sure,

Julian: And the least thing maybe we are thinking about is investigating, you can see how it ups the difficulty.

Michael: Absolutely

Julian: Ok

Kat: I have one more question on them, first you said you'd like game to be set in a castle, are you still happy with this option, coz now we are sort of thinking how the game should look like since we GUI now.

Julian: I was accepting what you are offering

Kat: Do you have any particular sort of you like, you like it to look like?

Julian: Em, it's really down to what you can if your best of imagining and working with, because I'm interested in eh, I want you to give me a narrative, and en…I'm sort of

providing a view on whether this is the kind of whether this game has got the interesting playability features that I like. I don't really care too much about the narrative. That's not true. I mean I want it to be interesting, I don't want. I'd rather more too obviously derivative, though that maybe hard too, avoid. So if you choose to do castle, castle is fine with me.

Kat: Yeah ok, we will go wild then, I think that's all from our sides.

Michael: Yeah pretty much

Julian: Ok, so how's the.. how's things getting on?

Kat: This week we mostly focused on like giving the documentation coz we have lots of difference sort of pieces you know we are very random like notes from meetings and you know, all the risk assessment and how to bring this thing into one document

Julian: So where is that process now

Kat: Well the process is being in the middle, we set up the document in latex and we started。like sort of segregating all those documents into weeks, and we started putting which user cases we did by what week, and that's the working progress, we already have the segregation of this by weeks when we write, this is going to be in a reasonable flow

Yeah sure

Michael: It's definitely a lot better than it was, and because we've now we've got documentation sorted, we should be able to program a lot more efficiently and faster from now on. So although not much programming once done, a lot I mean Alex a massive ton, these guys did really good on the GUI this week, but we should now be back onto programming again.

Julian: Sometimes one has to sort of feel like one standing still in order to be able to progress faster later.

Julian: So I have a question last week's question was as I was asking about documentation, this week's question is about the user manual, and the specification talks about being a centralized in paper representation, but it occurred to me that well maybe really a paper of representation isn't the most appropriate for this kind of product and would you rather do something more interactive.

Kat: I think that's the idea of game, we wouldn't want to sit and wait about the game, we just want to learn in playing.

Julian: So doing, eh maybe a video, rather than a user manual which sounds terribly sort of classical software engineering might be rather more fun coz that means you can incorporate bits of this is how you do the game along with some commentary and some subtitles if you want to as well and you can join it together, you can say this is how you do in the game , it only has the last minute and you've got that can and you can put something else some sort of frame bricks between it with a.. this chapter tells about and you can allocate by chapters too and this is how to do this aspect of this game and so on. So What I'm asking you about this because I'm kind of changing the specification, suggesting changing the specification should you want to do it like that, so what I said I will accept video instead of a paper document as the user manual if that's what you

would like to do.

Michael: Oh yes, I think it makes more senses as well, add in for instance, examples because this is programming focus, you can add in example scripts for people to use.

Julian: Exactly, yes, except that, Because of that particular property of your game, there's also going to be somebody might want to copy paste that and so you might want to have a library so to speak, of scripts that people can pick up and edit because that's the particular property of your game others won't have, trying to copy and paste out of a video.

Michael: Very difficult, at some point we almost have to have a kind of paper based feature a little bit because we need a short document saying this is syntax of how you write the program so there is something a little bit inescapable.

Julian: This is a bit of paper user guide but going to a chanting you might prefer because otherwise, ganning screenshots and then pulling them into paper document. It is so tedious.

Kat: No one would probably want to read for this anyway.

Julian: ok, that's cool.

Appendix F

Sprint Six: 22/11-28/11

Return to chapter 6.

F.1 Meeting Minutes

F.1.1 23/11

Meeting Minutes - 23/11

Attendees: Kat, Michael, Olivia, Andy

- Discussing the next iteration:
- Trying to merge two existing version, one of with GUI and one with the programmable bots.
- Putting the documentation together.

F.1.2 27/11

Meeting Minutes - 27/11

Attendees: Kat, Michael, Andy, Alex, Olivia

Recap of Sprint Cycle
- GUI
- Scrolling screen
    - Merging versions

This week
- Task prioritising
- Organising team resources against other CW deadlines
- Discussing project deadline
    - Need to hand-in by 11/12
        - Kat and Olivia are away at a conference after this date
        - Includes all code and documentation

Programming
- Aims for this sprint
    - Add graphics to programmable game
    - Change player code during game
    - Fixed map size
    - Fixed number of players
- Aims for next sprint
    - Scalability
        - Dynamic map size
        - Dynamic number of players
    - Graphics
    - Syntax checker
- Game Tutorials
    - Video
    - Document explaining game language syntax

Documentation
- Go through Interview Transcript 21/11
    - Graphics
        - Narrative/theme
    - User manual in form of video
    - Internally fix number of player turns
- Discuss gaps in current documentation
    - Testing, user story backlog, use cases, risk assessments, etc.

Next Meeting
- Wednesday 28/11 14:15
- Olivia to compile documentation to list
    - Share Overleaf edit link
- Kat to work on sprint overviews
- Michael, Andy, Alex to apply graphics to V2

F.2 User Stories

F.2.1 22/11 28/11

SPRINT 6

6.1 - As a player, I want the turn to be fixed externally

6.2 - As a player, I want to have a narrative, but I don't care what kind of narrative it could be

6.3 - As a player, I want to have a video as the user manual

6.4 - As a player, I want to have a syntax documentation so that I can know how to write the program for the bot

6.5 - As a player, I want to know where the bot is

Backlog - Sprint 6	**In Progress**	**Completed**
5.1 - As a player, I want the turn to be fixed externally	/	3.2 - As a player, I want to know where the bot is
5.2 - As a player, I want to have a narrative, but I don't care what kind of narrative it could be	/	/
5.3 - As a player, I want to have a video as the user manual	/	/
5.4 - As a player, I want to have a syntax documentation so that I can know how to write the program for the bot	/	/
/	/	/
/	/	/
/	/	/
/	/	/
/	/	/

F.3 Test Cases

F.3.1 22/11 28/11

SPRINT 6

6.1 - As a player, I want the turn to be fixed externally	fail
6.2 - As a player, I want to have a narrative, but I don't care what kind of narrative it could be	fail
6.3 - As a player, I want to have a video as the user manual	fail
6.4 - As a player, I want to have a syntax documentation so that I can know how to write the program for the bot	fail

6.1 - As a player, I want the turn to be fixed externally.

- The number of turns between being able to modify code is fixed for each game. Pass by sprint 6
- This however can be modified before each game. Pass by sprint 6

6.2 - As a player, I want to have a narrative, but I don't care what kind of narrative it could be.

- There is some kind of story to the game. fail
- There is a backstory. fail
- There is a reason for the bots and coins. fail

6.3 - As a player, I want to have a video as the user manual.

- There exists a video tutorial. fail
- The video tutorial explains how to use the game. fail
- The tutorial demonstrates all game functionality. fail
- Th tutorial shows how you can play the game. fail
- The tutorial demonstrates an example game. fail

6.4 - As a player, I want to have a syntax documentation so that I can know how to write the program for the bot

- There is a document which describes in detail the syntax for the game. fail
- The document shows all included functionality. fail
- The document shows correct examples. fail
- The document shows incorrect examples. fail
- The document demonstrates an infinite loop and presents the concept of the backend loop limiter to prevent the game from crashing. fail

F.3.2 Likert Scale

User feedback	1 Strongly Agree	2 Agree	3 Neither	4 Disagree	5 Strongly Disagree
Gameplay and usability:					
It is easy to understand what is happening on screen	1	2	3	4	5
The environment of the game is easy to navigate	1	2	3	4	5
The game provides clear objective	1	2	3	4	5
I understand how to control/play the game	1	2	3	4	5
The game is easy to “use”	1	2	3	4	5
The game provides feedback (positive or negative)	1	2	3	4	5
Architecture:					
The game play is well designed	1	2	3	4	5
The game is playable	1	2	3	4	5
The game runs without any bugs (smoothly)	1	2	3	4	5
The game needs to be more “customizable” (map size etc)	1	2	3	4	5
The game provides the player status with every turn	1	2	3	4	5
Motivation:					
The difficulty of the game is sufficient	1	2	3	4	5
I would like the game to increase difficulty with time	1	2	3	4	5
Pleasure (fun):					
I had fun playing this game.	1	2	3	4	5
The game kept me engaged	1	2	3	4	5

What was the most enjoyable part of the game:

What should be changed:

F.4 Programming Logs

F.4.1 26/11

Programming Log	
Participants	Alex
Date	26-11-2018
Time Started (nearest 15 mins)	15:00
Time Finished (nearest 15 mins)	18:30

Programming Objectives

- Update the 1st window of GUI to allow users to define the maximal number of steps in one turn.
- Update the 2nd window of GUI to offer a mini-map for players to show their current location in the whole map. The size of this mini-map should always be 100*100(pixel).
- Meantime, to offer a 25*25 map for the current player in which the current players are always in the centre. So the second map changes while the bot is moving and always only shows part of the whole map. This is to allow players see their surroundings clearly ,because if we print the whole map in just one frame ,everything becomes very small.

Outcomes

- All complete

F.4.2 27/11

Programming Log	
Participants	Michael, Andy, Alex
Date	27-11-2018
Time Started (nearest 15 mins)	15:42
Time Finished (nearest 15 mins)	23:30

Programming Objectives

- Refactor the code so that the separation between the bots, map obstacles, and collectables is more logical.
- Restructure the program in a more logical manner.

Outcomes

- The refactor was completed in full, with the code almost completely re-written from the top up.
- The MapInt object was changed into a MainMap object, this is an array of mapTiles, much like the previous version. However critically mapTiles now has three arrays, of an obstacle object, a collectable object and a bot object. This allows for multiple objects in the same tile, and brings everything into oo.
- Three new objects were created as containers for the obstacle, map tile and bot.
- A utilities class was created as a container for several functions useful to the program, specifically the random number function.
- The botlist class was created so that an easy to access record of current bots on the map is kept.
- The BotProgram class remains unchanged.
- Repackaged up most of the code to run the game in functions within MainMap, simplifying things greatly.
- The bots are now kept track of in tow places, both the botList and mainMap, with copies of each bot existing in both. This is to allow rapid indexing.
- The map has been made larger to allow for wall objects on the edge tiles, this simplifies the detection algorithms within BotProgram.

F.5 Customer Interview Documents

F.5.1 Interview Transcript

M: We've pretty much managed to complete the version 3 as per what we said we were going to do last week. So give you a quick lookback. So this is now a package of executable java game, so it's all been wrapped up, so it interprets the code here as you type in and then it will run the game, so you can see your bots move.

J: If I click next, it does what?

M: A single move for each bot and equally we've now implemented auto running so we can do the next 20 turns for instance, and it will complete the next 20. And simultaneously we've also implemented it so that you can change the code middle way through the game, so it is actually reading the program every time it makes moves, so you can change the program as we can move, but we do need to formalize the rules a bit better for this, and the moment it's just been implemented, but it needs to be.. it's kind of debugs a little bit per say.

J: How do I know how well I am?

M: So we have a score for player 1 here and a score for player 2 here.

J: It was a bit small.

M: So this is the game it's currently in, pretty much.

J: Ok cool

M: So moving ahead was version 4, we would like to implement multiple bots per turn, we would like to have finalized that sight, we would like to keep with this fixed map size with time being just to prioritise the other facets first, finalize that start, formalize the changing code middle way to, which was what I talked about earlier, and allow you to change the auto run makes, we can run it for more or less, and finally we have to make sure how the game ends that it has a sort of endings screen as such which says who's won quite ambiguously.

J: The auto run is that.. that's one player's choice, isn't, it?

M: Currently this is one we need to change how we will, formalisely how we will change the code, so that you can either run it before 20 turns automatically, then change the code, or you can do a manual next one, next one, next one for 20 turns and then change the code, that certainly looking at who has an agreement on whether you can do it.

J: Coz it could be beneficial for one player, for it to be a number of turns.

M: As always in deciding how many?

J: Yes.

M: so in the changing the auto run in.

J: Yes.

M: I see, Would something perhaps a bit like.. Then moving forward to version 5, so this will be our final sprint is to implement a syntax chapter the game and then to the map scrolling, so we have an arbitrary map size.

J: So how's the documentation going?

O: Pretty well, it's mostly altogether, we have a few holes but it's all linked up, Guess just filling gaps and keep on doing what we are in the current print.

J: Ok good, the question I'm going to ask this week is what you are going to do with play testing.

O: Play testing, what do you mean by play testing, I've never heard of that before.

M: As in getting people to play the game and see how it works for them?

J: That's one spectrum for play testing, yes, which is getting other people to play and write the view. With developing games there has to be a formal testing which assesses the playability of the game, such as does it provide you with a degree of engagement.

K: This is what we do for this scale, we were going to ask if you can fill it so we can have feedback.

J: Then I have to play the game, I can't do it without..

O: With this mention, I don't really think about in advance, should we research for this?

J: It's kind of like an acceptance testing but for games because if you've got a level which is impossible to beat, then nobody is going to be very happy with that, if you've got a level that is too easy, again, people would just give up. So it's a kind of meta requirement of any game. You should have playability , which moves the things what you are picking up at the moment in the flow in the engagement, so it's how much you in presenting and what contributes for that other things like, is what I'm doing achievable or am I just picking with random factors, that can I understand what.. have I got the reasonable change of looking at what the game is going to next, so I can come up with appropriate response to that.

O: Okay, I think we have sort of spoken about something like that. Not in so many words and not with the same name, but we'll address it.

J: So it's the present view of testing, anything else?

O: I don't think there's anything else.

J: Do we, all happy with in terms of what you've gotten and things that you are going to deliver, there are 17 days left.

O: We are working to Hannison on the 11th of Dec, because two of us are away for the last 3 days, so to get it out weighted, we will get it done by then, unless something exceptional happens, but we'll basically place the documentation and there's a case of filling the gaps and adding as we finish off this sprint.

Appendix G

Sprint Seven: 29/11-5/11

Return to chapter 7.

G.1 Meeting Minutes

G.1.1 04/12

Meeting Minutes - 04/12

Attendees: Kat, Micheal, Alex, Andy

.

Discussing the game progress at its current state:
Modifying code,
Formalising skips,
Updating the functionality

Testing:
Testing the game
Code overview,
Test cases

G.1.2 05/12

Meeting Minutes - 05/12

Attendees: Kat, Micheal, Alex, Andy

CHecking the documentation:
Discussing what needs to be done during this sprint cycle and task allocation.

.
Discussing the game progress at its current state:
Fromlised skipping and next turns,
Scrolling text area.
Multiple bots per team.
Planning the production of the maintenance guide and the syntax guide and video tutorial.

Modifying code,
Getting a start and end screen.

Customer interview:
Showing the progress the existing product.
Getting the feedback about the game.

G.1.3 10/12

Meeting Minutes - 10/12

Attendees: Kat, Micheal, Alex, Andy, Olivia

Finishing the documentation: Final report, User manual, installation guide, maintenance manual.

Final code review.
Discussing the contribution percentage and the project submission.
Doing the group questionnaire for the project.

.

G.2 User Stories

Backlog - Sprint 7	In Progress	Completed
6.1 - As a player, I want the turn to be fixed externally	/	6.1 - As a player, I want the turn to be fixed externally
6.2 - As a player, I want to have a narrative, but I don't care what kind of narrative it could be	/	6.2 As a player, I want to have a narrative, but I don't care what kind of narrative it could be
6.3 - As a player, I want to have a video as the user manual	/	/
6.4 - As a player, I want to have a syntax documentation so that I can know how to write the program for the bot	/	6.4 - As a player, I want to have a syntax documentation so that I can know how to write the program for the bot
/	/	/
/	/	/
/	/	/
/	/	/
/	/	/

G.3 Test Cases

G.3.1 29/11 05/12

SPRINT 7

6.1 - As a player, I want the turn to be fixed externally	pass
6.2 - As a player, I want to have a narrative, but I don't care what kind of narrative it could be	pass
6.3 - As a player, I want to have a video as the user manual	fail
6.4 - As a player, I want to have a syntax documentation so that I can know how to write the program for the bot	pass

G.4 Programming Logs

G.4.1 04/12

Programming Log	
Participants	Michael
Date	04-12-18
Time Started (nearest 15 mins)	15:00
Time Finished (nearest 15 mins)	17:00

Programming Objectives

- Formalise the way in which the skip button works.
- Display the current line location.

Outcomes

- Code now only editable editable every 10 turns.
- When not editable the text boxes grey out.
- The skip button now take the game to the next point a which the code is editable.
- The line location is also editable.

G.5 Programming Logs

G.5.1 05/12

Programming Log	
Participants	Michael
Date	05-12-18
Time Started (nearest 15 mins)	09:00
Time Finished (nearest 15 mins)	15:00

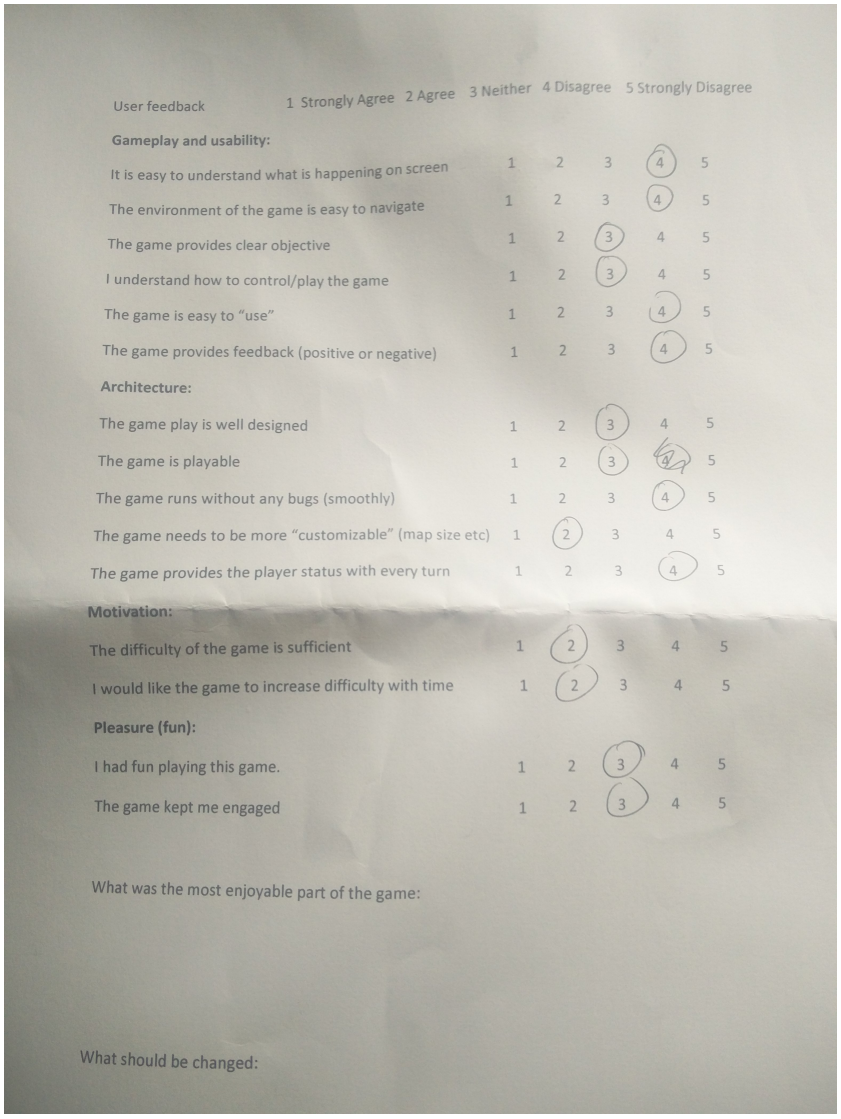
Programming Objectives

- Create a title screen.
- Add the capacity for multiple bots to the game.
- Allow for users to change the number of bots and number of coins before starting a game.
- Create scrolling tabbed text areas for the bot code input.

Outcomes

- Title screen implemented which allows users to select the number of turns between modifying the code, the number of coins, and the number of bots per game.
- Set the text areas to scrolling text areas to allow for code larger than the area provided.
- As multiple bots are now implemented, each textarea has a tab for each bot.
- The turn counter now only goes up by one once all bot's have had a turn - much like how turns operate in the game civ.
- Text areas default boot up with end program, this means that pressing next will work from the get go, making it easier to user.
- The game can be shut down and restarted anew from the title screen.
- The UI creating elements have largely been put in functions to simplify the code.
- The tabs are now created by a using lists of java swing objects.

G.5.2 Likert Scale with Customer Feedback



User feedback	1 Strongly Agree	2 Agree	3 Neither	4 Disagree	5 Strongly Disagree
Gameplay and usability:					
It is easy to understand what is happening on screen	1	2	3	(4)	5
The environment of the game is easy to navigate	1	2	3	(4)	5
The game provides clear objective	1	2	(3)	4	5
I understand how to control/play the game	1	2	(3)	4	5
The game is easy to "use"	1	2	3	(4)	5
The game provides feedback (positive or negative)	1	2	3	(4)	5
Architecture:					
The game play is well designed	1	2	(3)	4	5
The game is playable	1	2	(3)	(4)	5
The game runs without any bugs (smoothly)	1	2	3	(4)	5
The game needs to be more "customizable" (map size etc)	1	(2)	3	4	5
The game provides the player status with every turn	1	2	3	(4)	5
Motivation:					
The difficulty of the game is sufficient	1	(2)	3	4	5
I would like the game to increase difficulty with time	1	(2)	3	4	5
Pleasure (fun):					
I had fun playing this game.	1	2	(3)	4	5
The game kept me engaged	1	2	(3)	4	5

What was the most enjoyable part of the game:

What should be changed:

G.6 Customer Interview Documents

G.6.1 Interview transcript

M: So we have pretty much got our final game, we don't think we will be able to push out another iteration, because we got documentation to finish off,

J: That's very sensible, you don't need to be doing more functions up to the last minute, and you could say the game is what it is or the product is what it is come next Friday, you could also equally say, the game is what it is come this Friday, and stop adding any function at that point, you will have a function backlog, you have that function backlog recorded in your documentation, and that's where its face, and you can make sure the rest of the documentation is up to scratch, cos that's what's being logged.

M: For you to have a quick go and play the game, pretty much in the final state, in all of that, you can fill the format or doing a little bit of filling out the format.

J: How long is that gonna take?

M: Oh five minutes or so, hopefully. So you see much of programming functionality before, so what we can do now is we can change the number of bots per team, so two is a good one to start but you can go after 10. The map size, this is nine all limited to nine..

J: number 10 and the location is five, ok. The graphics all looking better than it was.

M: What we've got is 2, if you'd like to be player2, I could be player1 coz I've already got the program there, so you can switch over tab like that and then you can start writing a little bit program, so you've always gotta have in program, and you've got lots of different commands, you can see some of them on there as more sophisticated examples, and if you want to work out which bot is yours , you can put your mouse over the bot and it will tell you who exactly it belongs and which bot it is, so at the moment, you will program, so get it…

J: So is it one command per line, so if it went left, left, left…

M: Then it will pick up the coin, so we can quick next turn, you see they are all grid out so you can no longer edit it and so you ohh, I think I've had an error there, probably a syntax error which is caused to clash somewhere along the line, so that's one thing we would like to get implemented but never mind. Would you like me to reboot it, apologies for this problem, you got anything to discuss in the meantime, then go for it.

M: you got any questions about any aspects of the project as a whole documentation or anything.

K:Me and Olivia are mainly working out the documentation and now she's gone, most of it is done, well I'll have to go to conference at 12, so it cuts us even more to 3 people to do the next sprint cycle, this is pretty difficult to manage, not only we have but the fact that we have one person short at the beginning, so now Olivia be gone and I'll be gone soon so basically that leaves the final polishing to 3 people, we wrote this in this report.

J: So what stage is your documentation in at the moment, so if you and Olivia are working together, I would expect that it shouldn't make much difference as Olivia is not here, because you know where you are in the documentation as well.

K: It mostly set back with overleaf, because I've nerve used it and Olivia set up the documentation for the way it works, and I have to sort of learn about this but it was not that difficult, but it is quite time consuming, but I think we are mostly fine, we should be able to do within the time.

J: Ok good, so you've already made decision of not adding more functions, and that's fine, so essentially in order to be possible that stopping development now are probably from fixing things, behavior like this happening, and so I would imagine it ought to be possible to do what you need to do by Friday without too much difficulty, because as last week, you had something that worked in the way, this week you've got something that looks better, have you change the UI?

M: Yeah, quite a bit. So you can have multiple bots and a start screen, syntax error on the programming language, so the thing is that if you have a syntax error in your programming language, even on the previous version it would have terminated in an infinite loop and it doesn't proceed, the syntax check is quite a lot extra work to do, we don't think we'll be able to pull off and coz that's not easy, so anyway, it should work properly now again, I've got some syntax correct code there, If you click, you still got your basic programming, and keep on clicking next, so you'll do one 4 turn that is considered move for every bot, and once you reach five turns, and you can skip to the next point you can modify it too.

J: so how do I get it to restart?

M: It no long moves, and you can now set the line location to 0, which brings the computer program back to the very top, and you can now modify your code again,

J: So that's, so if you have those lines of numbers, all should be cleared off, that has to be done automatically.

M: It does command with current line as you need it.

K: If you have time and just quickly… Julian was filling out the questionnaire.

K: Are you happy with the product we provided, we didn't have many requirements besides this multiplayer set with bots, so we sort of felt like not knowing what we need to do. This is why we have this interview with you.

J: On the other hand, what you've got there, it is a multiplayer game, coz you've got multiple bots there, so those are the that are playing the game , so that seems to be doing well what was asked for, sure it's not online, but well that's something on the function backlog, rather than it's not something that can't be done, it could be very clear that one could extend to be another player somewhere else, also have a similar screen to that one, you can see their code as well, so are you concerned about with respect to the original requirement.

K: You could be meant to the create right product not just the working product, would you call this a right product for this project?

J: I think it's a right kind of product, it takes different with respect to the problem, that's nice, I think it needs more development as well, but as we said, it's in what state it is at the point of which the project Is running out of time, because you've followed an agile process, you have got a working product which meets some of the requirements better than some of the others, but it nevertheless means the requirement is based on what we've discussed, in other sense, if one flows the process, and the product will be the right product, and this is what we have done together.

J: Are you all happy with where the product is?

K: Despite the fact our team was not very strong in terms of programming, because Michael is the strongest one and many of us were just tipping our toes, I think we did quite impressive work.