Final Report

Team MAICKers

Isaiah Britto

Angela Gadon

Kiana Ziglari

Christian Devile

Michael John Bradford

December 1, 2017
CS 141 - Intro To Prog & Problem Solving
Edwin Rodríguez

Introduction

As a group of 5 we coordinated our code on gitHub and met on a Discord server intermittently throughout each week. We planned the program's architecture extensively before coding so that we had an understanding of how the classes would interact. We took a bottom-up approach to building the code, which helped us realize which classes we needed and which were vestigial. The AI was programmed in tandem with the creation of Entities, while the Graphical User Interface was implemented after the text-based game functioned properly with all required components.

Problem Description

Create a small text-based game. In this game there is a building that contains nine rooms. In one of these rooms, there is a briefcase. The player's objective is to navigate the building, check the rooms, and find the briefcase.

The building is pitch black, and there are six ninjas on patrol. In addition to the ninjas and the rooms, there are three power-ups randomly placed throughout the room: invincibility, a radar, and a bullet. The invincibility power-up makes it so that the player's are immune to any ninja stabs for five turns. The radar allows the player to know exactly which room the briefcase is. The bullet gives the player another bullet, only if they have used their first one already.

On the ninjas' turn, the first thing they do is check the squares surrounding them. If a ninja starts their turn and the player is in an adjacent square, the player loses a life and has to restart in the bottom left corner. The player has three lives, and if they lose them all it's game over. After the ninjas attempt to stab the player, they then use the rest of their turn to move. There are two types of movement the ninja can do. Ninjas can either make random moves or calculated ones, depending on what the player chooses. If the player chooses to enable the ninja AI, the ninjas will move towards a player if the player is within their line of sight, and a room is not separating them.

On the player's turn, they can look, and then perform another action. Looking will show the player what's located in the two squares ahead of them. After looking, the player can decide to either move, or shoot. If they decide to shoot, the bullet will kill the first ninja it hits, given that one is in that direction and there is no room separating the

player and the ninja. During any of their turns, the player can choose to save the game and load it at another time.

Approach To Solution

The first thing we did for this assignment, before writing any code, was to plan the classes we would need and how they would interact with each other. Once we thoroughly planned everything, we made the classes in Eclipse, filling them with empty methods based on what we thought we would need. All of these methods were documented so that we'd know exactly what they're meant to do. For some of the methods, we found it difficult to know what we would need without writing some code to get an idea of how the program would run.

When planning we realized that an essential part of the program was how we stored the game board and the location of everything inside it. Initially, we decided that everything would keep track of where it was using an integer to represent its x-location and an integer to represent its y-location. We realized, however, that this would make it difficult to check whether anything was in a particular square. With this method, rather than being able to check what's in a specific square, every single item would have to be checked to see where it is. This meant all 6 ninja locations would have to be checked, as well as all the power-up locations, room locations, and the player location. Instead, we thought it would be better to store everything into an array, so that the specific array index could be checked, making things run much more efficiently.

Once we finished creating the skeleton code, we split the classes up, giving different responsibilities to each group member. This allowed us to all work at our own pace, uploading to GitHub whenever we made significant progress. By synchronizing

our code we were able to work together, building off of what the other group members were doing.

Once major parts of the code were finished, we of course found some minor errors and bugs that needed to be fixed. Whoever found the bug would do their best to find what was wrong in the code and fix it on their own, however the more complex problems were solved through group effort. We would tell the other group members exactly what was wrong in the program and what we thought the error in the code might be. With all of us taking a look at it, at least one person would be able to pinpoint the problem and a solution for it

The User Interface needed some special attention as we planned to attempt a Graphical User Interface if project time allowed for it. We decided to create a super class called UserInterface that would have all of the abstract methods that the Game Engine could call to command the UI, and return input. This way, at runtime, the ui could be specified to be Console based, or GUI based, and the remainder of the code would function without issue.

Discussion Of Implementation

What we considered to be one of the most pivotal points of the assignment was how the board and the objects placed on it were represented. We decided we would represent the board with an array of Strings. The array index represents the location on the board, and the String represents the entities in this location. The fact that there could be two things on a square at once complicated the situation. So we decided to have each String contain three characters. The first character represents the player, the second represents an item, and the third represents a ninja. For example, if a String read “100”, this means that there is a player in this location and nothing else. If a String read “0r1”, this means the player is not in this location, but the radar power-up and a ninja is.

Our program is centered around the GameEngine class, which does the majority of the logic computing and executes the turns step by step. This generally starts with the GameEngine asking for input from the UserInterface. The UserInterface will print a message to the user, and ask them to enter input. Before returning this input, the UserInterface verifies that what was entered is a valid option. This input is then returned to the GameEngine, which uses the input to execute the player’s desired actions. Based on these actions, the GameEngine will then update the GameBoard, moving anything that has been moved, removing any power-ups that have been used, or removing any ninjas that have been shot. This updated map is sent to the UserInterface, which transforms the map from the three character data Strings to a user-friendly display and prints it out.

Testing Data

Test #	Action	Expected Result	Actual Result	Check
1	User inputs a letter when a number is expected	A message prints telling the user their input is invalid, and prompts them to enter again.	A message prints telling the user their input is invalid, and prompts them to enter again.	✔
2	User inputs a number when a letter is expected	A message prints telling the user their input is invalid, and prompts them to enter again.	A message prints telling the user their input is invalid, and prompts them to enter again.	✔
3	User inputs an option out of the given bounds	A message prints telling the user their input is invalid, and prompts them to enter again.	A message prints telling the user their input is invalid, and prompts them to enter again.	✔
4	User tries to walk into a wall/Off the board	The user is told this is an invalid move and is again asked which direction they would like to move in	The user is told this is an invalid move and is again asked which direction they would like to move in	✔
5	User saves a game twice with the same name	The second save point takes priority and that game from that point will be saved with the given file name	The second save point takes priority and that game from that point will be saved with the given file name	✔
6	User tries to shoot when they are out of ammo	The user isn't allowed to use the shoot action.	The user isn't allowed to use the shoot action.	✔
7	User tries to look after they've already looked	The user isn't allowed to use the look action a second time in one turn.	The user isn't allowed to use the look action a second time in one turn.	✔

8	User tries to do more than one action on their turn (aside from looking)	The user isn't given the option to do more actions than they are allowed	The user isn't given the option to do more actions than they are allowed	✔
9	User tries to use the look action to see inside a room	The user cannot see what is in the room unless they use their movement to go into it.	The user cannot see what is in the room unless they use their movement to go into it.	✔
10	User shoots in a direction with multiple ninjas	Only the first ninja in that direction is killed.	Only the first ninja in that direction is killed.	✔
11	User shoots a ninja	That ninja is killed and removed from the board	That ninja is killed and removed from the board	✔
12	User tries to load a game that doesn't exist	The user is told this game doesn't exist and is asked to re-enter which game they were trying to load	The user is told this game doesn't exist and is asked to re-enter which game they were trying to load	✔
13	User gets invincibility power-up	The user cannot be stabbed by ninjas for the next five turns	The user cannot be stabbed by ninjas for the next five turns	✔
14	User got invincibility power-up but five turns have passed	The user can be stabbed by ninjas as usual	The user can be stabbed by ninjas as usual	✔
15	User gets radar	The user can see where the briefcase is	The user can see where the briefcase is	✔
16	User gets ammo while out of ammo	The user gains one bullet	The user gains one bullet	✔

17	User gets ammo while they already have ammo	The user doesn't gain an extra bullet	The user doesn't gain an extra bullet	✔
18	User walks out of the power-up space	The power-up is removed from the board	The power-up is removed from the board	✔
19	User gets the briefcase	The user wins the game and is asked if they would like to play again	The user wins the game and is asked if they would like to play again	✔
20	User is stabbed by a ninja	The user loses a life. If they have at least one life, they are transported back to the bottom left corner	The user loses a life. If they have at least one life, they are transported back to the bottom left corner	✔
21	A ninja is in the start space after user is stabbed	That ninja is removed and placed on a random spot of the board	That ninja is removed and placed on a random spot of the board	✔
22	One ninja moves into another ninjas space	This cannot happen, only one ninja is allowed per square	This cannot happen, only one ninja is allowed per square	✔
23	A ninja walks over a power-up	The power-up isn't used, and remains there for the player	The power-up isn't used, and remains there for the player	✔
24	When AI is on and ninja is in line of sight of the player	The ninja moves one square towards the player	The ninja moves one square towards the player	✔
25	When AI is on and ninja isn't in line of sight of the player	The ninja moves randomly	The ninja moves randomly	✔
26	When AI is on	The ninja moves	The ninja moves	✔

	and ninja is in line of sight of the player but they're separated by a room	randomly	randomly	
27	When AI is on and ninja is in line of sight of the player but they're separated by another ninja	The ninja moves one square towards the player	The ninja moves one square towards the player	✔
28	When AI is on and user moves out of line of sight without using a Room	The AI ninja then moves randomly since the player is not in the field of vision for the ninja anymore	The AI ninja then moves randomly since the player is not in the field of vision for the ninja anymore	✔
29	When AI is on and user moves out of line of sight using a Room	The AI ninja then moves randomly since the field of vision for the ninja is blocked by the room	The AI ninja then moves randomly since the field of vision for the ninja is blocked by the room	✔
30	Loading a saved game with the radar power-up active	The briefcase is still visible to the player	The briefcase is still visible to the player	✔
31	Loading a saved game with the invincibility power-up active	The player remains immune to stabbing until the five turns are over. (If a player had already used three of the turns, they only get two more when they load)	The player remains immune to stabbing until the five turns are over. (If a player had already used three of the turns, they only get two more when they load)	✔
32	Loading a saved game that has ninja AI enabled	The ninjas continue to move as they should when AI is enabled	The ninjas continue to move as they should when AI is enabled	✔
33	Loading a saved	The same amount of	The same amount of	✔

	game with less than 6 ninjas on the board	ninjas are on the board as when the game was saved	ninjas are on the board as when the game was saved	
34	Loading a saved game with the player out of ammo	The player is still out of ammo	The player is still out of ammo	✔
35	Loading a saved game with the player at less than max lives	The player's life total is whatever it was when it was saved	The player's life total is whatever it was when it was saved	✔
36	User loses a life then saves and loads game	The player should only have two remaining lives	The player should only have two remaining lives	✔
37	When User loses two lives then saves and loads	The player should only have one remaining life	The player should only have one remaining life	✔
38	When User saves continues playing exits without saving and then loads previous save.	The game starts in the same state as when they saved, ignoring any action done after that	The game starts in the same state as when they saved, ignoring any action done after that	✔
39	When User plays a new game and then exits without saving, and loads a previous save.	The game they exited is erased, the previous save should load accurately	The game they exited is erased, the previous save should load accurately	✔
40	When A User starts a new game. (Initialization)	Everything is placed on the board in random positions, with only the briefcase in a room	Everything is placed on the board in random positions, with only the briefcase in a room	✔
41	A User opens	The help menu displays	The help menu	✔

	the help menu.		displays	
42	User exits the help menu.	The user returns to the main menu	The user returns to the main menu	✔
43	User beats the game without saving.	The user is asked if they would like to play again	The user is asked if they would like to play again	✔
44	User loads a previous save then beats the game.	The user is asked if they would like to play again and the save point is still there	The user is asked if they would like to play again and the save point is still there	✔
45	User saves, continues playing and then beats the game.	The user is asked if they would like to play again and the save point is still there	The user is asked if they would like to play again and the save point is still there	✔
46	User chooses to play again	The game restarts	The game restarts	✔
47	User chooses not to play again	The program quits	The program quits	✔
48	User uses the look action and there is a room directly in front of the direction they choose to look	The user can't see anything because the room blocks their line of sight	The user can't see anything because the room blocks their line of sight	✔
49	User enters -1 to see debug mode	Everything on the board should be revealed, and the asterisks should be gone	Everything on the board should be revealed, and the asterisks should be gone	✔
50	User goes into debug mode at the beginning of their turn, then executes the rest of their turn	The user should be able to do the next two actions of their turn as usual	The user should be able to do the next two actions of their turn as usual	✔

51	User goes into debug mode after looking	The user should be able to move or shoot as usual	The user should be able to move or shoot as usual	✔
52	User leaves debug mode	Everything should be covered and look exactly like it did before debug mode was entered	Everything should be covered and look exactly like it did before debug mode was entered	✔
53	User enters debug mode and exits debug mode in the same turn	The user should be able to enter/exit debug mode on their turn as much as they want and still execute the rest of their turn	The user should be able to enter/exit debug mode on their turn as much as they want and still execute the rest of their turn	✔
54	User tries to move into a room from the north side	Their location doesn't change but a message prints letting them know whether or not there is a briefcase in the room	Their location doesn't change but a message prints letting them know whether or not there is a briefcase in the room	✔
55	The user tries to move into an empty room from the north side	A message prints saying that the room is empty	A message prints saying that the room is empty	✔
56	The user tries to move into the room with the briefcase from the north side	A message prints saying that the briefcase was found and the game was won	A message prints saying that the briefcase was found and the game was won	✔
57	The user wins the game	A message prints saying they won, and they are asked whether or not they want to play again	A message prints saying they won, and they are asked whether or not they want to play again	✔
58	The user loses the game	A message prints saying they lost, and they are asked whether or not they want to play again	A message prints saying they lost, and they are asked whether or not they want to play again	✔

59	The user starts a game, passing in “-gui” to enable GUI mode	The GUI is enabled	The GUI is enabled	✔
60	Player is stabbed by a ninja.	Player is returned to start position while the remaining board remains unchanged. Ninjas, Briefcase and Power-ups remain unchanged.	Player is returned to start position while the remaining board remains unchanged. Ninjas, Briefcase and Power-ups remain unchanged.	✔
61	Player collects Radar in debug mode	Nothing happens because the briefcase is already shown in debug mode	Nothing happens because the briefcase is already shown in debug mode	✔
62	Player moves into a ninja's location	The player gets stabbed, loses a life.	The player gets stabbed, loses a life.	✔
63	Player enters a 1 after they’ve started a game	The player is asked what direction they’d like to move in	The player is asked what direction they’d like to move in	✔
64	Player enters “n” to move North	The player moves up on the game board	The player moves up on the game board	✔
65	Player enters “s” to move South	The player moves down on the game board	The player moves down on the game board	✔
66	Player enters “e” to move East	The player moves right on the game board	The player moves right on the game board	✔
67	Player enters “w” to move West	The player moves left on the game board	The player moves left on the game board	✔
68	User looks in debug mode	Nothing should change since all the squares are shown anywars	Nothing should change since all the squares are shown anywars	✔

69	User shoots into a direction with no ninjas	No ninjas are removed from the board	No ninjas are removed from the board	✔
70	User shoots into a direction with a powerup	The powerup remains there and is unaffected	The powerup remains there and is unaffected	✔
71	User tries to move in the direction of a wall when they are already on the edge of the board	Their location doesn't change and the user is told the move is not valid and do please choose another direction.	Their location doesn't change and the user is told the move is not valid and do please choose another direction.	✔
72	User starts a game in the text based console, saves it, and tries to load it from the GUI	The user is able to save the game, exit the text based console, open the GUI and load it there. The game will start on the GUI from the save point	The user is able to save the game, exit the text based console, open the GUI and load it there. The game will start on the GUI from the save point	✔
73	User initiates a new GUI game	When a new GUI game is started, the rooms, power-ups and ninjas are properly pictured respectively	When a new GUI game is started, the rooms, power-ups and ninjas are properly pictured respectively	✔
74	Player enters a 2 after they've started a game	The user is asked what direction they'd like to look in	The user is asked what direction they'd like to look in	✔
75	Player enters "n" to look North	The player can see two squares above the Spy's location if not blocked by a room or wall	The player can see two squares above the Spy's location if not blocked by a room or wall	✔
76	Player enters "s" to look South	The player can see two squares below the Spy's location if not blocked by a room or wall	The player can see two squares below the Spy's location if not blocked by a room or	✔

			wall	
77	Player enters “e” to look East	The player can see two squares right the Spy’s location if not blocked by a room or wall	The player can see two squares right the Spy’s location if not blocked by a room or wall	✔
78	Player enters “w” to look West	The player can see two squares left the Spy’s location if not blocked by a room or wall	The player can see two squares left the Spy’s location if not blocked by a room or wall	✔
79	User uses the look action and there is a room 2 squares in front of the direction they choose to look	The user is only able to see inside the square directly in front of them, the second square is blocked by the room	The user is only able to see inside the square directly in front of them, the second square is blocked by the room	✔
80	User chooses to play the game without ninja AI	The ninja move randomly on the board	The ninja move randomly on the board	✔
81	Player Shoots in a direction that contains no ninjas	The game board remains unchanged and the player loses the bullet	The game board remains unchanged and the player loses the bullet	✔
82	Player tries to kill a ninja when there is a room between them	The bullet is stopped by the room, the ninja remains on the board and the player loses the bullet	The bullet is stopped by the room, the ninja remains on the board and the player loses the bullet	✔
83	User quits a game at the main menu	The program closes	The program closes	✔
84	User tries to enter a room from the East	The user is told this is an invalid move and is again asked which direction they would like to move in	The user is told this is an invalid move and is again asked which direction they would like to move in	✔

85	User tries to enter a room from the West	The user is told this is an invalid move and is again asked which direction they would like to move in	The user is told this is an invalid move and is again asked which direction they would like to move in	✔
86	User tries to enter a room from the South	The user is told this is an invalid move and is again asked which direction they would like to move in	The user is told this is an invalid move and is again asked which direction they would like to move in	✔
87	Player quits a game after they've started a new game	The program closes and any moves that were made are lost	The program closes and any moves that were made are lost	✔
88	Player quits game after loading a previous game	The program closes and any moves that were made are lost	The program closes and any moves that were made are lost	✔
89	Player shoots in the direction of a power-up	Power-up remains on the board. Nothing on the board changes, just the player wasted their bullet	Power-up remains on the board. Nothing on the board changes, just the player wasted their bullet	✔
90	User gets trapped by ninjas on all sides with no bullets left	Player can only move into ninja and die, losing a life	Player can only move into ninja and die, losing a life	✔
91	User gets trapped by a combination of ninjas and walls with no bullets left	Player can only move into ninja and die, losing a life	Player can only move into ninja and die, losing a life	✔
92	User gets trapped by a combination of ninjas, walls,	Player can only move into ninja and die, losing a life	Player can only move into ninja and die, losing a life	✔

	and a room with no bullets left			
93	User looks at a square where there are both a ninja and a power-up present	The visibility of the ninja takes priority, therefore the ninja is shown and not the power up	The visibility of the ninja takes priority, therefore the ninja is shown and not the power up	✔
94	Two ninjas are near each other and move into each others spots	The two ninjas physically swap locations but it appears as though they don't move one the board	The two ninjas physically swap locations but it appears as though they don't move one the board	✔
95	Player wants to walk over a power-up but not collect it	There is no way to not collect the power-up. The power-up is activated as usual when the user moves into that square	There is no way to not collect the power-up. The power-up is activated as usual when the user moves into that square	✔
96	User loads a game in the GUI that the player had already collected the radar power-up	The location of the radar is still visible	The location of the radar is still visible	✔
97	User loads a game in the GUI with no ammo left	The player should still have no ammo when loading the game into the GUI	The player still has no ammo when loading the game into the GUI	✔
98	User loads a game in the GUI with 2 lives left	The player should still have only two lives left when loading it into the GUI	The player still has only two lives left when loading it into the GUI	✔
99	User loads a game into the GUI with 1 life left	The player should still have only one life left when loading it into the GUI	The player still has only one life left when loading it into the GUI	✔

100	User loads a game into the GUI which already has the the extra bullet power-up collected	The player should still have a bullet to use but the extra bullet power-up should be off the GUI gameboard.	The player still has a bullet to use but the extra bullet power-up should be off the GUI gameboard.	✔

Conclusions

Overall we learned that planning and communication are the two most important parts of programming a group project. After changing our approach to the program, we realized that there are advantages and disadvantages to any solution. One of the most important things we learned was that by spending the extra time to thoroughly plan our approach and think of what methods we would need, we ultimately saved time in the long run. While we did need to add more methods even after planning, having the framework set made it much easier to see what was still necessary.

Suggestions For Improvements

The game has lots of room to improve and expand from the original concept without sacrificing the principles of the game.

To increase the complexity of the gameplay, we could add more difficulties for the player to choose from. For each of these difficulties the AI of the enemies would account for a larger range of sight, or have more abilities like moving two spaces in one turn. Additionally, more objects could be added to the map. For example hiding spots could be placed randomly, acting as safe spots for the player to avoid a ninja stab, and also evade a ninja's line of sight with AI enabled.