Defence Science and Technology Group (DSTG) and Swordfish Computing Project

Distributed Decision-Making

THE UNIVERSITY
of ADELAIDE

Sprint Retrospective 3 – a1734056

Group COMPLEX 8

a1734056	Hayden Lee
a1734069	Vinh Nguyen
a1743599	Nathan Van der Hoek
a1744852	Harrison Bagley
a1746088	Daniel O'Connor
a1746146	Patrick Capaldo
a1748751	Sarah Damin
a1749935	Sam Davies
a1773841	Hayley Richardson

I attended the sprint review/planning meeting on 05/10 with the tutor

What went well in the sprint?

The team worked well in the third sprint and a lot more progress could be visibly seen with improved use of the Github code repo and project board. Meetings have maintained efficient pacing, group members were able to share ideas and knowledge, and tasks were organized and distributed more effectively. Knowledge sharing went especially well. At the start of the sprint, we still were not clear on whether the MASON framework suited our purposes, but by the end we had a much clearer understanding with tangible functionality that can be demonstrated. More proficient team members wrote documentation and tutorials to aid others that were having difficulty with the setup, which was extremely helpful in getting everyone up to speed. This helps with software development as starting fresh with new software and a new environment can be very challenging at the start. By scheduling meetings and creating documentation within the subteam, it ensures that each member has a baseline level of understanding so that they can work separately on additional implementation/features sooner and with more confidence. Discussion and collaboration with the product owner has also improved significantly, particularly with completing project board tasks since our original visions of what constitutes a completed task often extended beyond what was required at a minimum. This helped with ensuring a streamlined workflow/momentum and assisted with removing blockers.

What could be improved?

For some group members, including myself, there was a lack of diligence with completing tasks by the assigned 'due date'. In my case, this was due to not being able to meet with my other sub-team members to try and resolve code compilation issues, a consequence of not merging early which caused integration issues. This was a significant blocker with software development, as it became unclear with which program should be used going forward which halted progress over the uni break. Another area that could be improved was the use of slack channels, especially for MASON help, as team members experiencing difficulties would go onto the slack channel to check for help, find nothing, and have to scour through the messenger chat for files instead. This was a very inefficient way of working. More disciplined use of the slack standup bot and weekly review could also be improved, as doing these activities later in the week often lead to a rushed workflow/atmosphere.

What will the group commit to improve in the next sprint?

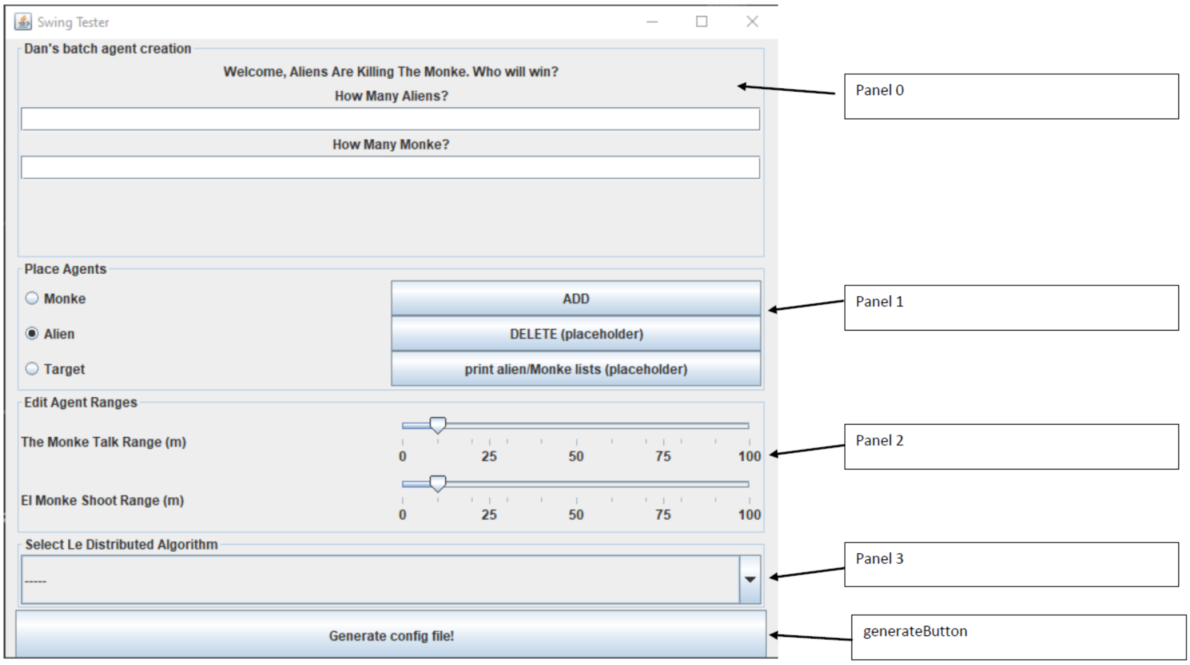
The group will maintain keeping track of tasks by creating newly assigned tasks into cards onto the GitHub project board. The group will also try and break these into singular sub-tasks where required to make them more traceable, easier to complete, and aid with distributing amongst team members. This will aid software development by reducing the number of periods where there is no perceived progress. The team will also aim to shift communications from Facebook messenger to Slack and GitHub. Instead of using messenger to inform the group directly about software development activities, these will instead be communicated in Slack with a quick "hey check Slack" message on messenger to raise attention. The team will also aim to reduce integration nightmares by merging more frequently and by starting work earlier from the same file.

Comment on your progress this sprint

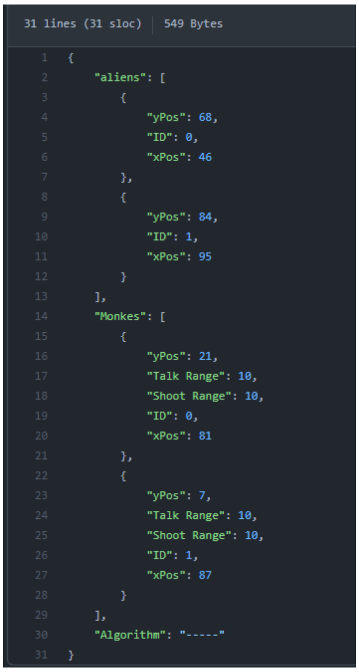
My tasks:

- Configuration Profiles #3
- Frontend GUI

As of 18/09, this is what the front end UI looks like:



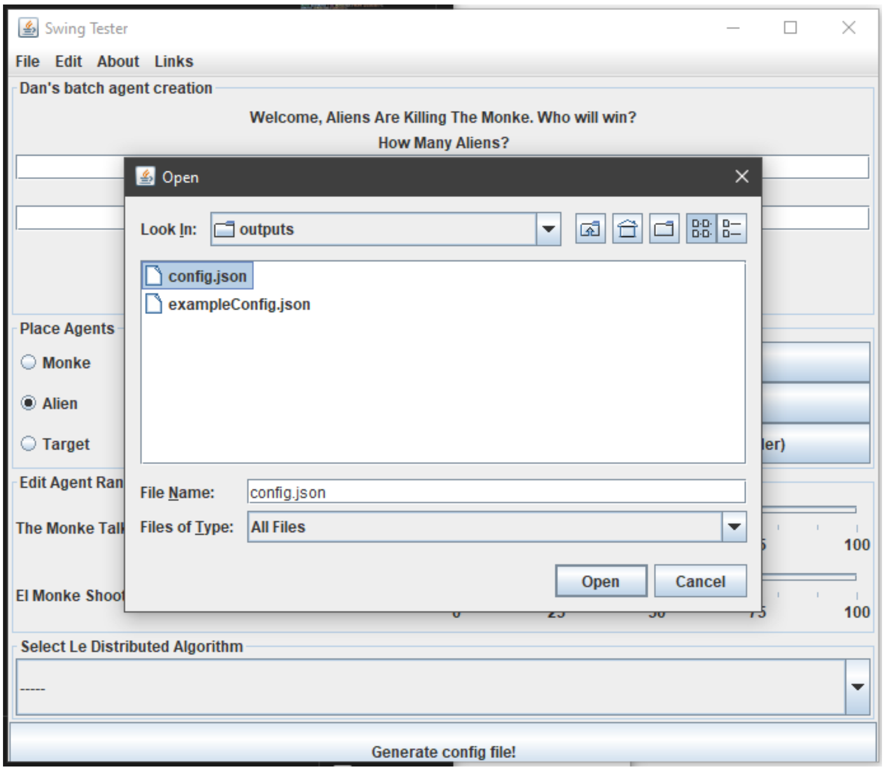
It features multiple panels with agent creation functionality working, which ultimately produces a config file. Since the last retrospective, it also captures additional details such as the algorithm selection, and talk/shoot ranges only for the monke agents. The X and Y positions have also been randomised as well.



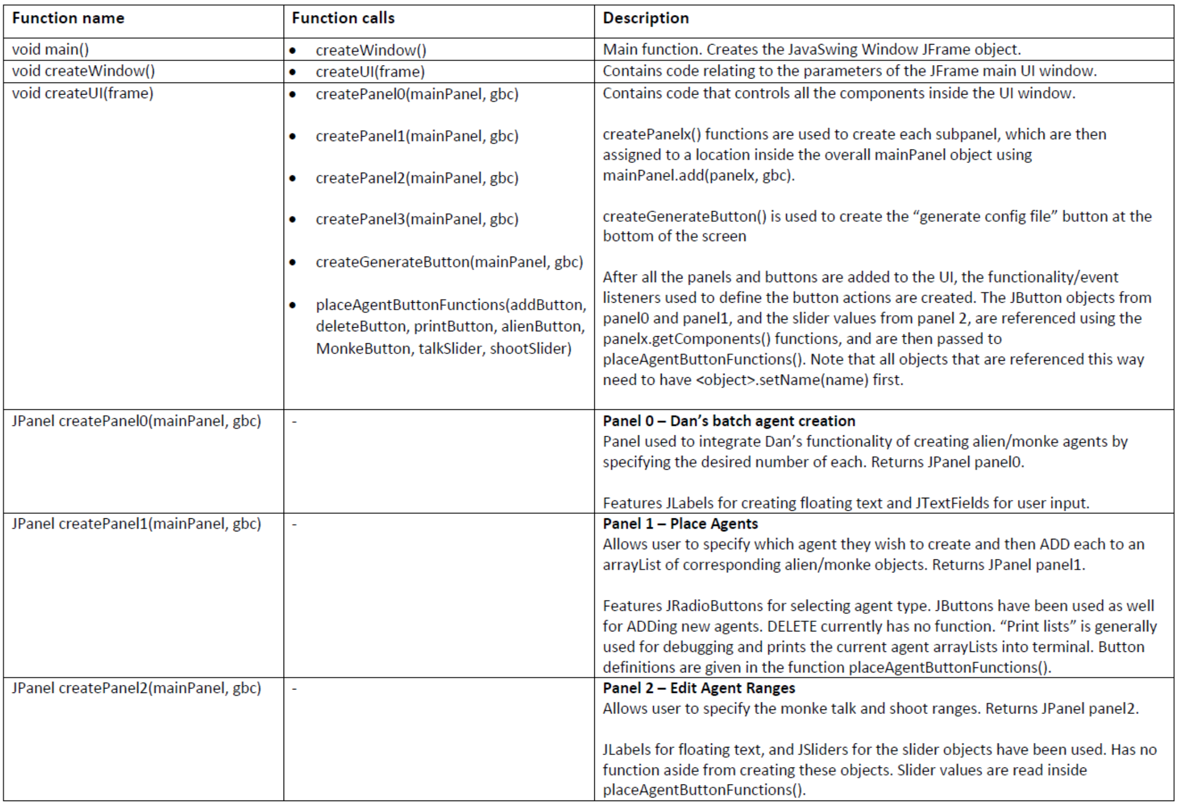
31 lines (31 sloc) | 549 Bytes

```
{
    "aliens": [
        {
            "yPos": 68,
            "ID": 0,
            "xPos": 46
        },
        {
            "yPos": 84,
            "ID": 1,
            "xPos": 95
        }
    ],
    "Monkes": [
        {
            "yPos": 21,
            "Talk Range": 10,
            "Shoot Range": 10,
            "ID": 0,
            "xPos": 81
        },
        {
            "yPos": 7,
            "Talk Range": 10,
            "Shoot Range": 10,
            "ID": 1,
            "xPos": 87
        }
    ],
    "Algorithm": "-----"
}
```

An additional menuBar feature has also been added. This allows for import functionality which allows for a user to select a file to "open", which selects that config file and creates the list of aliens and monkes with the parameters specified in the file.

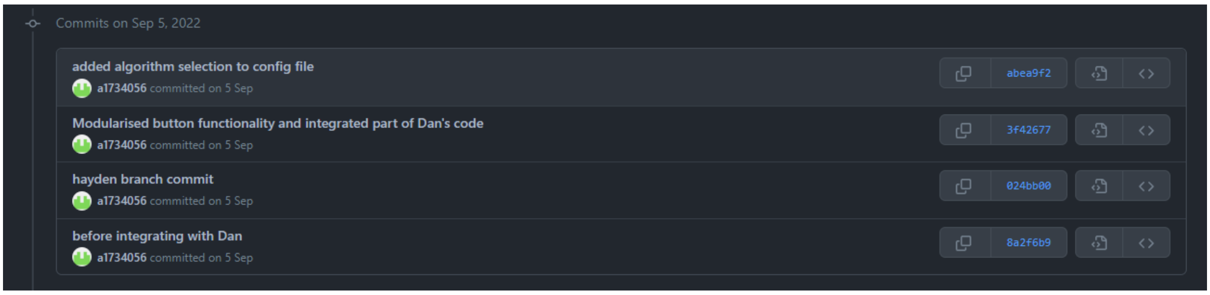


Whilst merging with Dan’s code, I also tried to refactor it so that each panel’s creation was abstracted away into its own function. To aid with this, I wrote documentation about the purpose of each. Here is a screenshot of the first page:



Function name	Function calls	Description
void main()	• createWindow()	Main function. Creates the JavaSwing Window JFrame object.
void createWindow()	• createUI(frame)	Contains code relating to the parameters of the JFrame main UI window.
void createUI(frame)	• createPanel0(mainPanel, gbc) • createPanel1(mainPanel, gbc) • createPanel2(mainPanel, gbc) • createPanel3(mainPanel, gbc) • createGenerateButton(mainPanel, gbc) • placeAgentButtonFunctions(addButton, deleteButton, printButton, alienButton, MonkeButton, talkSlider, shootSlider)	Contains code that controls all the components inside the UI window. createPanelx() functions are used to create each subpanel, which are then assigned to a location inside the overall mainPanel object using mainPanel.add(panelx, gbc). createGenerateButton() is used to create the “generate config file” button at the bottom of the screen After all the panels and buttons are added to the UI, the functionality/event listeners used to define the button actions are created. The JButton objects from panel0 and panel1, and the slider values from panel 2, are referenced using the panelx.getComponents() functions, and are then passed to placeAgentButtonFunctions(). Note that all objects that are referenced this way need to have <object>.setName(name) first.
JPanel createPanel0(mainPanel, gbc)	-	**Panel 0 – Dan’s batch agent creation** Panel used to integrate Dan’s functionality of creating alien/monke agents by specifying the desired number of each. Returns JPanel panel0. Features JLabels for creating floating text and JTextFields for user input.
JPanel createPanel1(mainPanel, gbc)	-	**Panel 1 – Place Agents** Allows user to specify which agent they wish to create and then ADD each to an arrayList of corresponding alien/monke objects. Returns JPanel panel1. Features JRadioButtons for selecting agent type. JButtons have been used as well for ADDing new agents. DELETE currently has no function. “Print lists” is generally used for debugging and prints the current agent arrayLists into terminal. Button definitions are given in the function placeAgentButtonFunctions().
JPanel createPanel2(mainPanel, gbc)	-	**Panel 2 – Edit Agent Ranges** Allows user to specify the monke talk and shoot ranges. Returns JPanel panel2. JLabels for floating text, and JSliders for the slider objects have been used. Has no function aside from creating these objects. Slider values are read inside placeAgentButtonFunctions().

As I tasked myself with making the merges, the complexity was relatively high which kept me preoccupied throughout the sprint. A screenshot of my commits for the UI is shown below.



Commits on Sep 5, 2022

added algorithm selection to config file
a1734056 committed on 5 Sep

Modularised button functionality and integrated part of Dan's code
a1734056 committed on 5 Sep

hayden branch commit
a1734056 committed on 5 Sep

before integrating with Dan
a1734056 committed on 5 Sep