Method selection and planning

a) Give an outline and justification of the team's software engineering methods, and identify any development or collaboration tools that the team has used to support the project or the team working. Justify the fitness of the selected tools with the team's software engineering methods and discuss alternatives considered. (3 marks, ≤ 2 pages).

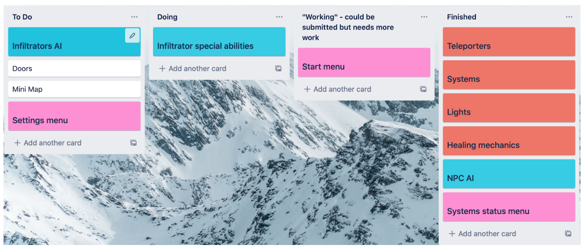
b) Outline the team's approach to team organisation, and explain why the chosen approach is appropriate for both the team and the project (2 marks, ≤ 1 page).

c) Give a systematic plan for the project. Your plan should lay out the key tasks, their starting and finishing dates, as well as task priorities. The plan should also identify a critical path and task dependencies. Provide weekly snapshots of the plan on your team's website and discuss how the plan evolved throughout the duration of the project (5 marks, ≤ 2 pages).

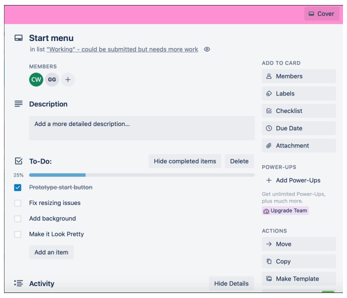
a) We used the Scrum method for software development as it allowed a more dynamic approach to tasks. Scrum is a method focused on planning and identifying requirements in the initial stage, then moving to a short sprint of development (2-4 weeks) wherein the development team meet daily to ensure the project is on task and to check there is no overlap in work. It allowed for greater transparency for every team member to access project information and it allowed for regular inspection of progress so adjustment can be made where necessary as issues arise. As we have a very short time frame in which to complete the project, we chose an agile method to be able to plan and organise our team's time effectively.

We used the following collaborative tools:

- Discord was used for communication and conducting team meetings. We have a team server with separate channels for general conversations, useful links/information and implemented a bot to notify of new git commits. This tool enabled us to
- Google doc was used for documentation, as it allows for synchronous collaboration from the whole team remotely with access to edit and comment history. The data is stored externally on a cloud server so it is more protected against data loss and is saved every few seconds.
- Trello is a kanban board we used to track ongoing tasks, their progress and completed work. Each pair of programmers was assigned a colour to easily identify their set tasks, and are able to move their tasks to the various columns and add a checklist of items to be completed to finish their task, breaking it down further (see screenshots below). Any unassigned tasks are left white for a team to pick up as their own when they finish tasks.
  This went well with our Scrum approach to development as it meant we could easily identify what was needed to be done and any changes to our plan could be effectively communicated. We had a weekly board for all ongoing tasks (not pictured) and a board specifically for implementation tasks.



To Do	Doing	"Working" - could be submitted but needs more work	Finished
Infiltrators AI	Infiltrator special abilities	Start menu	Teleporters
Doors	+ Add another card	+ Add another card	Systems
Mini Map			Lights
Settings menu			Healing mechanics
+ Add another card			NPC AI
			Systems status menu
			+ Add another card



Cover

Start menu
in list "Working" - could be submitted but needs more work

MEMBERS
CW GG +

Description
Add a more detailed description…

To-Do: Hide completed items Delete
25%
- [x] ~~Prototype start button~~
- [ ] Fix resizing issues
- [ ] Add background
- [ ] Make it Look Pretty

Add an item

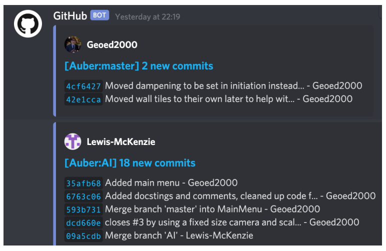
Activity Hide Details

ADD TO CARD
Members
Labels
Checklist
Due Date
Attachment

POWER-UPS
+ Add Power-Ups
Get unlimited Power-Ups, plus much more.
Upgrade Team

ACTIONS
Move
Copy
Make Template

- Github was used as a remote repository for the project and storing code. We also had a bot on our team discord server to send a message when new commits, branches and merges occurred so the whole team could see what work was ongoing. This stream of information supplemented our daily scrum meeting.



GitHub BOT Yesterday at 22:19

Geoed2000
[Auber:master] 2 new commits
```
4cf6427 Moved dampening to be set in initiation instead… - Geoed2000
42e1cca Moved wall tiles to their own later to help wit… - Geoed2000
```

Lewis-McKenzie
[Auber:AI] 18 new commits
```
35afb68 Added main menu - Geoed2000
6763c06 Added docstings and comments, cleaned up code f… - Geoed2000
593b731 Merge branch 'master' into MainMenu - Geoed2000
dcd660e closes #3 by using a fixed size camera and scal… - Geoed2000
09a5cdb Merge branch 'AI' - Lewis-McKenzie
```

- Git was used as a version control system to track all commits from all team members.

The alternative collaborative tools we considered were Monday and Asana. We did not choose to use Monday as we found the website to be incompatible with members of the team using Mac OS and the members of the team who could use it found the tool to be hard to understand and use. We did not choose Asana because the features we liked in Trello were available there but behind a paywall and, again, the tool was unwieldy.

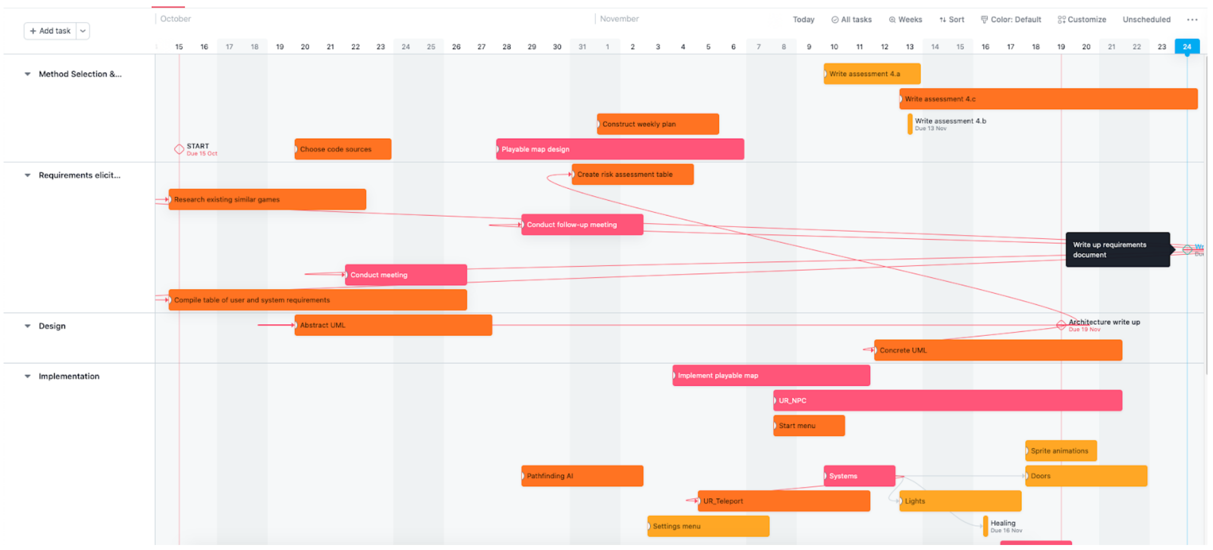
b) Our approach to team organisation was to track all ongoing tasks on the Trello board supplemented by the checklist system. We kept all non development work, such as contributions to our list of references, documentation, basic sketches of the game map and other things organised within folders on our team google drive. This made any resources needed for implementation easy to access and find for the whole team. The scrum master, Chloe, went through all of the tasks on a weekly basis to make sure they were well organised and to see if anything needed to be modified. This meant that the team was always on top of things and well organised. For the team, this meant that we were able to work more effectively as everything we needed was readily available in a clear format.

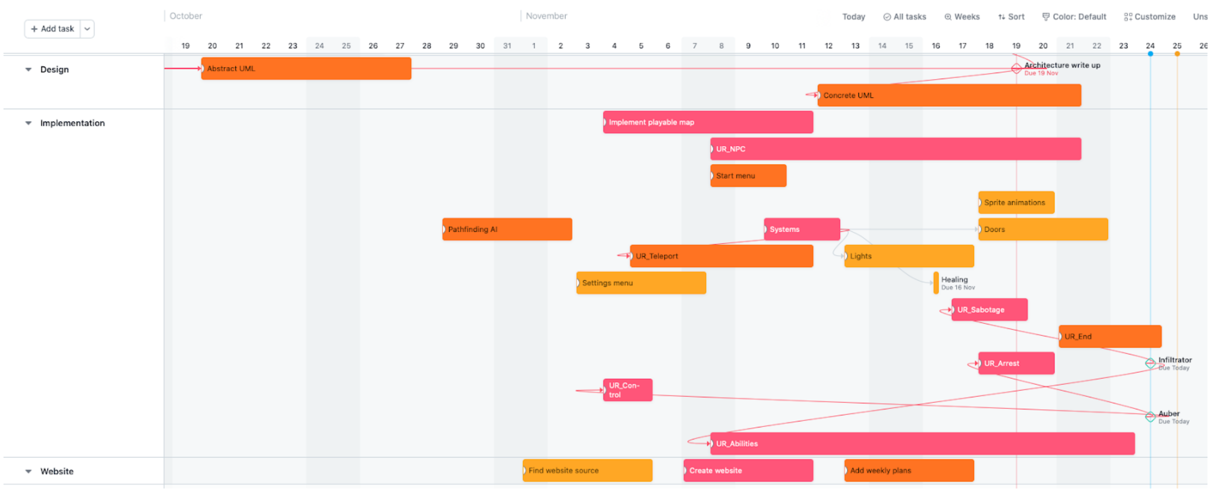
c) When identifying our key tasks, we used a table format alongside a traffic light system of colour coding for priority, which we transferred over to our Trello board for the weekly plan. Red is high priority, orange is medium and green is low. Previous weeks' tasks moved up a priority level if they went uncompleted the week before.

Task	Dependant(s)	Start date	Finish date	Priority
Requirements	Ben	12/10		High
Abstract UML	Lewis, Matthew	12/10		Medium
Risk assessment	Matthew	15/10		High
Method selection and planning	Chloe, Ben	15/11	24/11	
Architecture write up	Chloe, Lewis			
Team/customer meeting(prep, meeting, write up)	Chloe, Ben	26/10, 02/11		Medium
Code planning	Chloe, Haopeng	02/11	Ongoing	Medium
Playable map design	Chloe	23/10	03/11	High
Playable map implementation	Haopeng, George	04/11	10/11	High
UR_Control	Haopeng	04/11	05/11	High
Start menu	Chloe, George	08/11		Medium
Systems menu	Haopeng	16/11	17/11	Low
Pause menu				Low
UR_Teleport	Haopeng	05/11	8/11	High
Infiltrator AI	Haopeng, Lewis	15/11	20/11	High
UR_Abilities	Haopeng	08/11	23/11	High
UR_Sabotage	Haopeng	17/11	19/11	High
UR_NPC	Haopeng, Lewis	18/11	21/11	High
Settings menu				Low
Systems	Haopeng	12/11	16/11	High
Lights	Matthew	15/11		Low
Healing mechanics	Haopeng	16/11	16/11	Low
Doors	Lewis			Low
Sprite animations	George	18/11		Medium
Mini map	Matthew	17/11		Medium
Demo mode				Medium

UR_Arrest	Haopeng	18/11	20/11	High

We also created a Gantt chart to plan our tasks which can be seen weekly on the website





As the project evolved, we added more tasks as we worked due to thinking of new ideas and features that needed to be added. We started off heavy with planning and documenting, and then had a month long sprint of implementation. During this stage, people from separate parts of the product would meet bi-weekly to discuss progress, and offer outsider insight onto those particular pieces of work, which proved invaluable as it meant we could plan to develop additional features which were not thought of in the first stage of requirements creation.