Team Task Management with Space Invaders

With the previous lab exercise we started to work together as a team: identifying tasks that needed to be done, distributing those tasks amongst the team, and collaborating using source control to combine our efforts.

This week we are going to add a layer of management on top to better allow us to keep track of tasks, including who they are assigned to and how they are progressing.

Time constraints

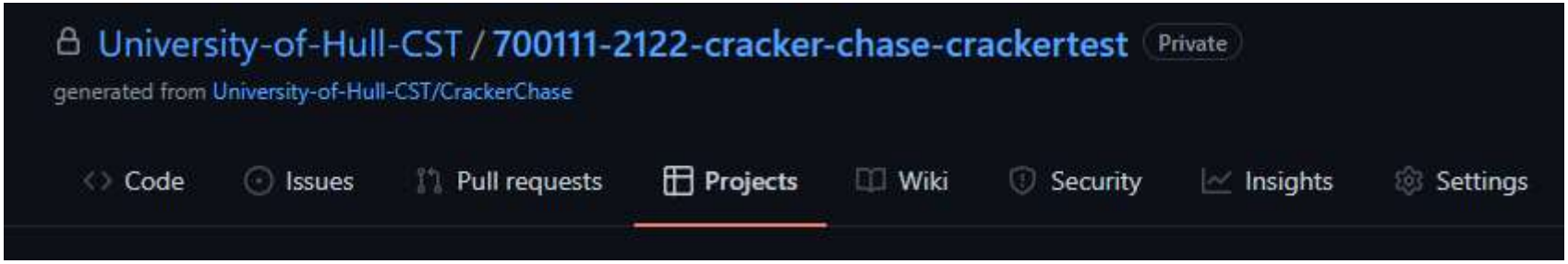
This exercise is deliberately time constrained. You should not be working on issues outside of the lab sessions allocated to it. Feel free to read this ahead of the session, but please don't start the activity.

Kanban using Projects in GitHub

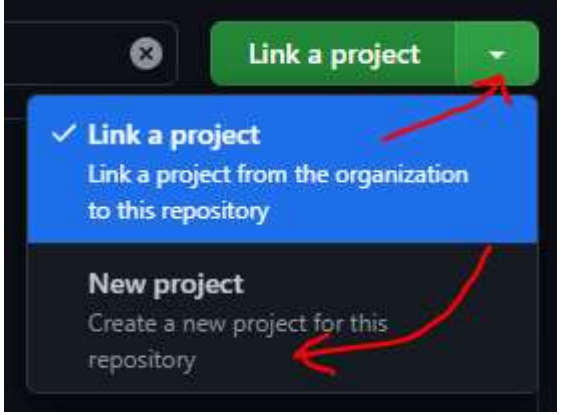
Kanban (https://en.wikipedia.org/wiki/Kanban_(development)) has found popularity as a lightweight project management tool in Agile software development. In its early use (which still has a place) it manifested as sticky notes on a board, but nowadays many online implementations exist. Some advantages of it being online is its persistence, particularly in a remote working environment, and traceability, so you can track the lifecycle of tasks.

GitHub has its own implementation of the Kanban concept called "Projects".

Your team's first task (note this only gets done once per team) is to navigate to your team's Cracker Chase repository (the one in the screen shots is my test repo). In the menu bar at the top, click on the "Projects" tab.

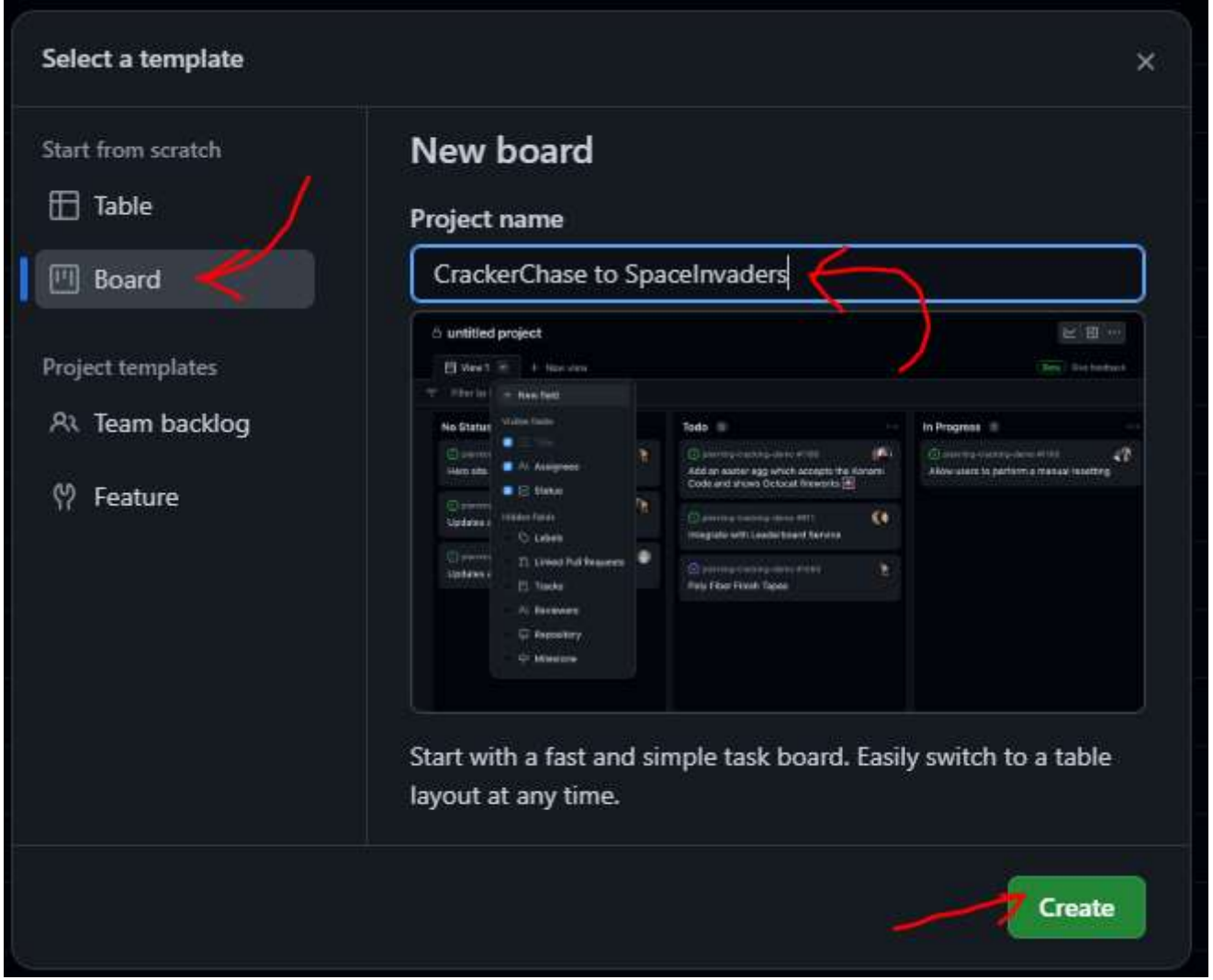


This will open a new page where there are currently no projects listed. You should click on little drop down arrow next to the "Link a project" button in the top right of the page. Then select the "New project".

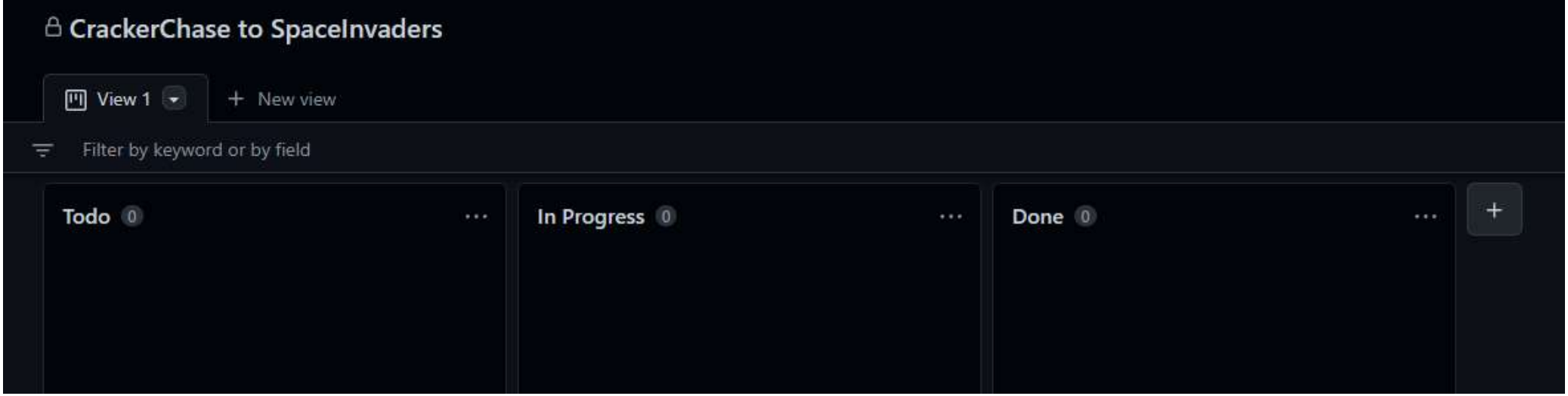


Now you can click on the green "New project" button that is revealed in the same spot.

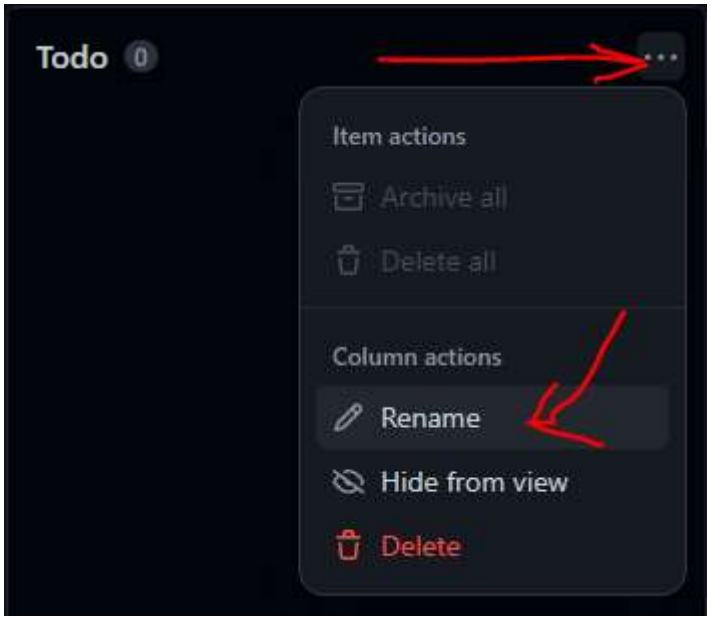
This will reveal a new form. You need to give your project a name. Spoiler alert, I recommend something like "Space Invaders Cracker Conversion (*yourteamname*)". A concrete example of this would be "Space Invaders Cracker Conversion (antelope 2223)". Make sure that you select the "Board" option on the left. Then click the "Create" button.



This will create a default empty board.



As your first column, you will need a "To Do" list which is also often known as the Backlog. So let us rename it "To Do (Backlog)".

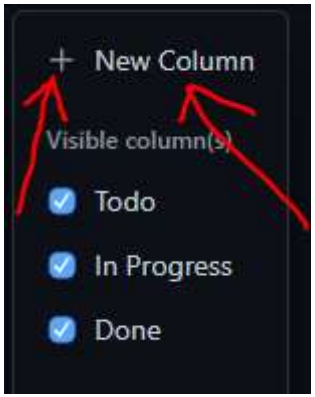


The "In Progress" column we will keep the same. We will use this column to move tasks that are currently being worked on.

But we are not done with our columns yet. We need a place to put tasks that are complete, but rather than have a single "Done" column we are going to make several different columns so that we can keep track of *when* they were completed. So start by renaming our "Done" column to "Sprint 1".

One of the popular flavours of Agile software development methodologies is **SCRUM** (https://en.wikipedia.org/wiki/Scrum_(software_development)) . A defining characteristic of this methodology is the dividing of development time in to "Sprints". The most common sprint duration is around 2 weeks, but can vary from project to project.

For this week's exercise we are going to have six 40 minute sprints, spread over two sessions. For our columns this means that we need six extra columns named "Sprint 1", "Sprint 2" etc.



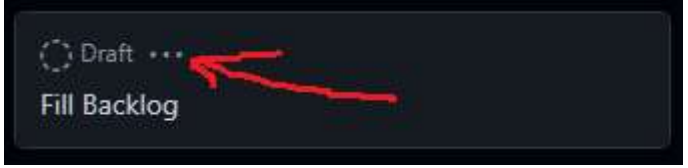
You can add a column by clicking on the "+" to the right of the columns, then you can click on the "New Column" button. This will create a new column that you can enter a name for.

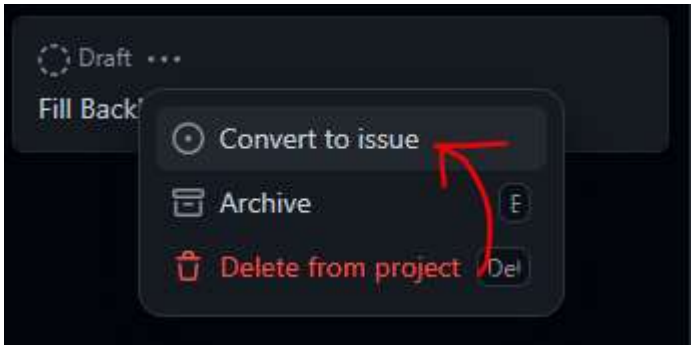
Adding Items

We can add items to our backlog. At the bottom of each column there is an "Add item" button. Click this button in the Backlog column.

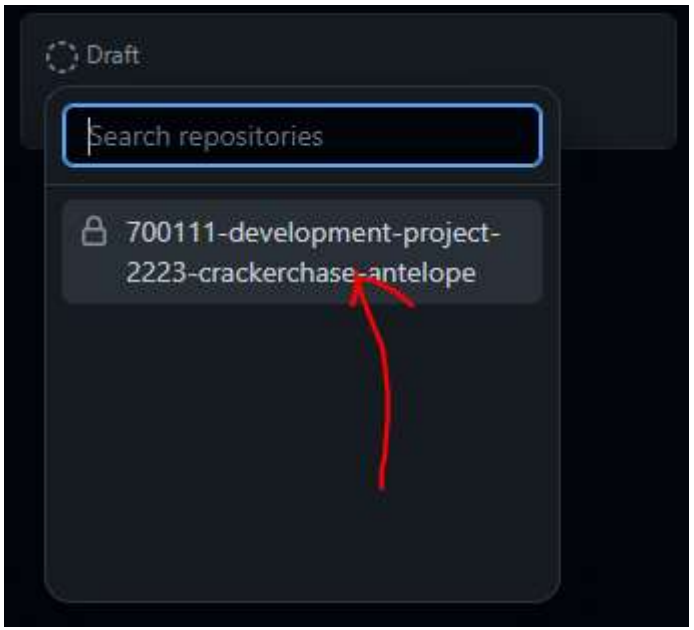
This will reveal a text box where we can add a meaningful title for the task. Let's call the first team task "Fill backlog".

The draft that is created is not terribly useful to us because it is just a bit of text. We can move it around, but if we convert it to an "Issue", then it is much more powerful. In the top right of our "Fill backlog" draft, click the ellipsis "...", then click the "Convert to issue" option that is revealed.



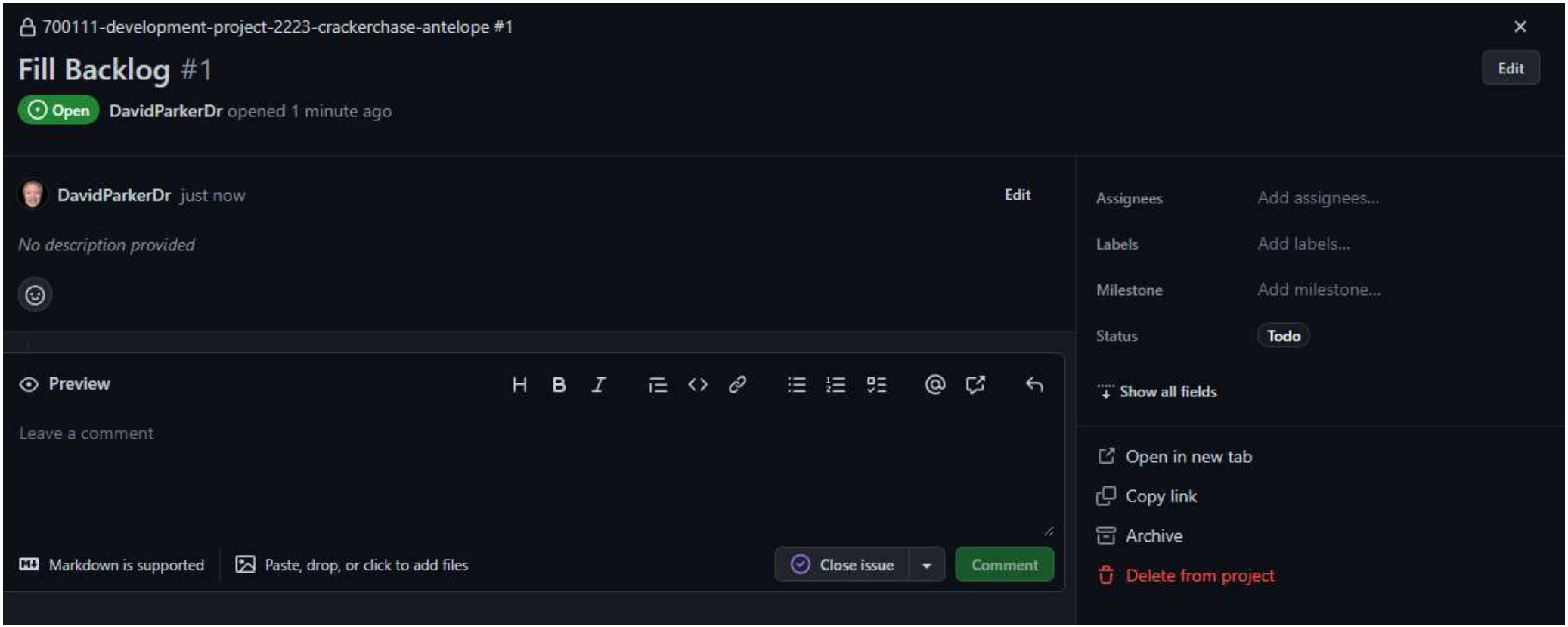


This will prompt you to select the repository to add the issue to. If there is more than one option, choose your group's repository.



You will notice that the note has now changed in appearance in our backlog.

If you click on the name of the issue, you will reveal a side panel on the right of the page. This gives us an overview of the issue and allows us to carry out most of the tasks required. For the full functionality of the issue view, you can click on the name of the issue in the top left, in this case "Fill Backlog".



Assigning Tasks to Team Members

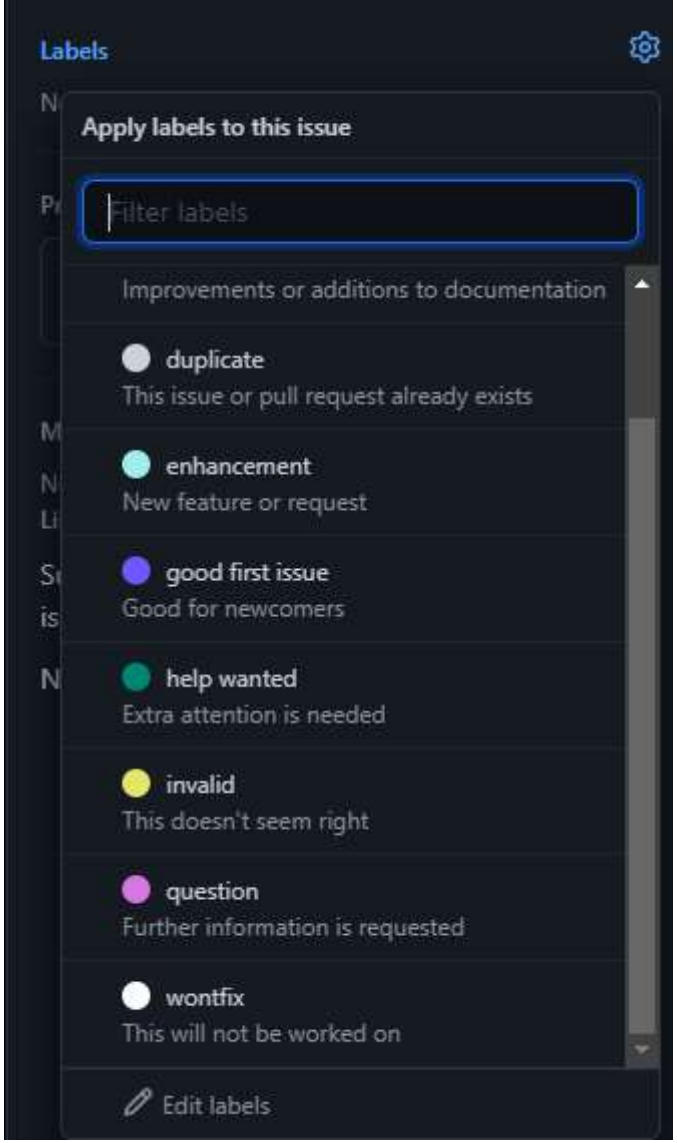
The first think we are going to do with our issue is "Assign" it to team members. For this task, we want everyone involved.

Click on the little cog icon link to the right of the word "Assignees" in the issue side panel. Make sure that everyone in your team is added to this issue. You can do that by clicking on their names in the list.

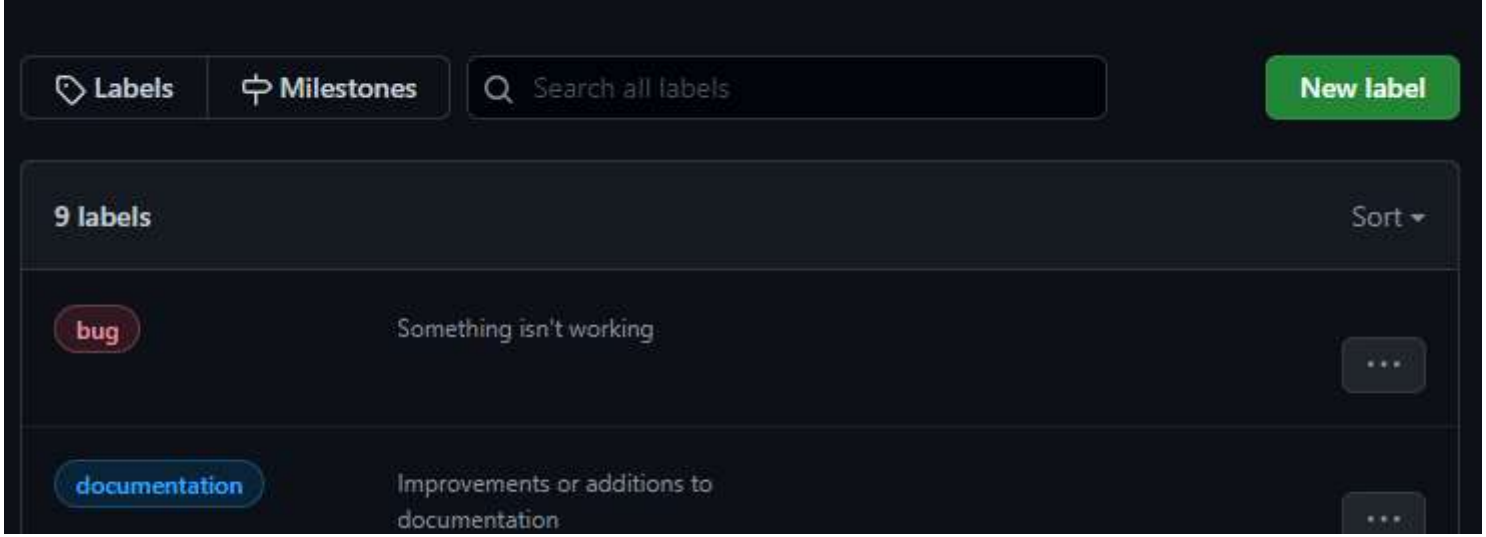
Adding labels

Once we have a lot of tasks to manage it will be useful to be able to add labels to them to help with the overview.

Clicking on the little cog icon to the right of "Labels" will reveal another panel that allows you to add one or more colour coded labels to an issue. There are some built in ones, but we can add our own using the "Edit labels" button. Click on this button.



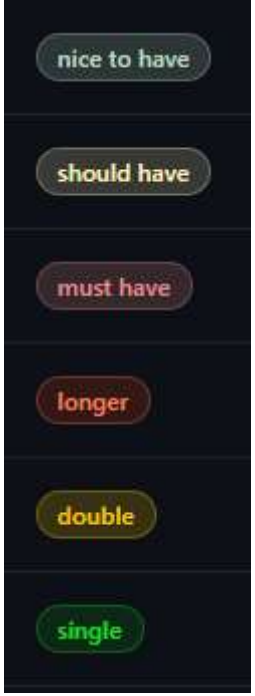
This takes us to a new page that shows all of the existing labels. We can also click on the "New label" button in the top right to add one. We are going to do this.



We are going to create three labels to indicate our estimate of how many sprints our future tasks will take. People are not very good at estimating time units precisely, especially when the time periods get longer, so we are going to talk in terms of Sprints. Will it be a "single" sprint, a "double" sprint, or "longer". I assigned the colours green, yellow, and red respectively for my labels.

It would also be useful to indicate how important each task is. You can maybe use "must have", "should have", and "nice to have", with each one being less important.

We now have 6 new labels that we can use for our issues planning.



Navigate back to our project board and click on the "Fill backlog" issue again to reveal the side panel. Open the labels list again and assign the "single" label and the "must have" label to this issue.

You should now see our issue in the Backlog updated.

If you click and drag on it in the project board, then you can move it between columns. Do this to move it in to the "In Progress" column.

The brief

Ok, so now you have a set of tools at your disposal for managing the exercise. But what is the exercise? In the last session you we tasked with making some updates to the fabulous Cracker Chase game. This time we are going to take this a bit further, by converting it to **Space Invaders (https://en.wikipedia.org/wiki/Space_Invaders)** !

The Cracker Chase game provides a basic framework: you can make things appear on screen, you can move things around, you can tell when things have collided with other things. This is essentially all you need to create the venerable classic of Space Invaders.

Assets

To make it look like space invaders, you will need some art assets. You don't have to make them as they are provided here:

- **Alien 1** (https://commons.wikimedia.org/wiki/File:Space_invaders_character_4.jpeg)
- **Alien 2** (https://commons.wikimedia.org/wiki/File:Space_invaders_character_6.jpeg)
- **Alien 3** (https://commons.wikimedia.org/wiki/File:Space_invaders_character_5.jpeg)
- **Player ship** (https://commons.wikimedia.org/wiki/File:Space_invaders.character.jpeg)
- **Flying Saucer** (https://commons.wikimedia.org/wiki/File:Space_invaders_character_3.jpeg)
- **Bunkers** (https://commons.wikimedia.org/wiki/File:Space_invaders_character_2.jpeg)

Fill your backlog

So at this stage, all members of your team have been assigned the task "Fill backlog". To complete this task you will have to discuss together all of the tasks that you think will be necessary to complete the brief of converting your Cracker Chase game to Space Invaders.

As you identify tasks, add them to your backlog. As a first pass, you may consider just using notes as placeholders for your discussion. Then as you firm up the tasks, convert them to issues and add labels to indicate their level of priority and your estimate of how many sprints each task will take. When you estimate the timings, consider the person hours involved (or perhaps the person sprints). You will have up to four developers working each sprint. Your estimate should be based on 1 developer assigned to the task.

Keep in mind, that you want to try and break down large tasks in to smaller tasks and that you also want to try and avoid having multiple branches working on the same files (to avoid merge conflicts).

Working with issues

Once your backlog is established, you will be able to start work on the issues. The general pattern that you want is as follows:

- Allocate the issue to a developer
- Move the issue in to "In Progress"
- The developer will create a branch to work on the issue
- When happy that the issue has been solved, commit and push the branch
- Generate a Pull Request
- Assign a reviewer
- The reviewer should approve and merge the Pull Request (if satisfactory)
- Maintain communication with the team

Issue Branches

It is a good idea to have a naming convention for your branches. If you include a reference to the issue that you are working on, that helps with traceability. Each issue that you create will have an id number. For example, your "Fill backlog" issue probably has the id #1.

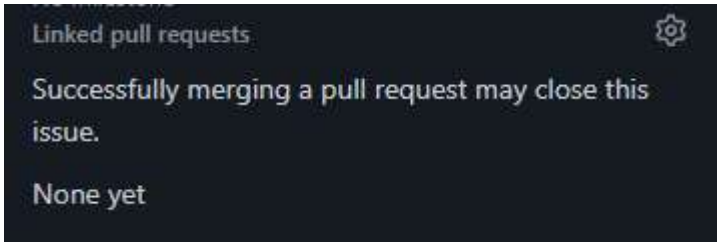
Now you wouldn't create a branch for the "Fill backlog" issue as you aren't interacting with the source. But a hypothetical future issue might be to acquire all of the image assets and add them to the project. Perhaps this is issue #5.

So when creating a branch to work on this issue, a sensible branch name would be "issue5-add-image-assets-to-project". It includes an specific reference to the original issue, and it has a quick title that also relates to the issue for easy reading when viewing amongst multiple branches.

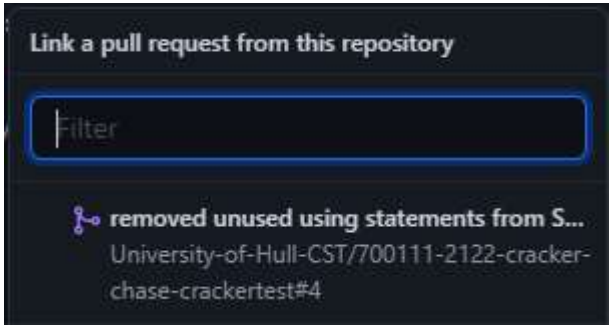
It is important to maintain a convention such as this for consistency.

Closing Issues

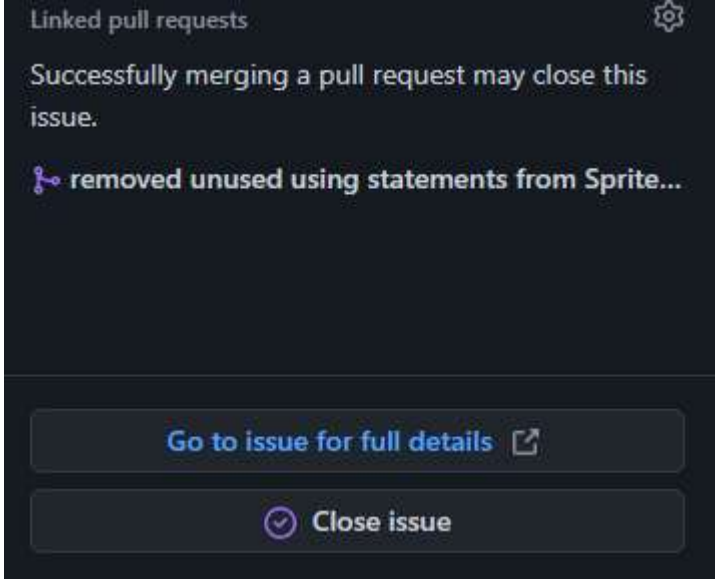
A neat way of closing issues is to link them to the Pull Request that fixes the issue. You can do this from the specific issue page, or from the side panel view of the project board.



Click on the little cog icon to the right of "Linked pull requests". This should reveal a searchable list of pull requests that you can link to this issue. If you click on one it will link the pull request and the issue.



One you have done this, you can close the issue. By clicking on the "Close issue" button.



Now you can move the issue card in to the column that corresponds to your current sprint.

End of Sprint (Retrospective)

At the end of each sprint, you should get together as a team to discuss how the sprint went. You can reevaluate tasks and issues, you may add things to your backlog. As issues get completed, new allocations of work should happen and moved from the backlog to the in progress column.

It may be that as part of this reflective process it becomes apparent that you won't be able to achieve everything you set out to do. So further rounds of prioritisation may determine which features get focused on and which get abandoned.

Communication

Communication is key. It is important to ensure that everyone knows what they are doing, what is being done and by whom, and the avoidance of merge conflicts. Make an extra effort to engage all of your team members in any discussions. You may feel that it would be easier to just do it all by yourself; this is bad team work.

Crackers?

Linked below is some information about the base Cracker Chase game and how it is structured. This may aid in the Space Invaders conversion.

Cracker Chase Guide.pptx (https://canvas.hull.ac.uk/courses/65765/files/4250740?wrap=1) ↓ (https://canvas.hull.ac.uk/courses/65765/files/4250740/download?download_frd=1)