Windows/Linux

SSH-Extension

From the extensions-pane you should find and install the **'Remote - SSH'** extension from Microsoft.

Setting up your Virtual machine

To make sure that all handins conform to the same standard and avoid compatibility-issues we will supply you with a virtual machine the has the same specifications that we as TA's use for testing. This will ensure that if everything looks good on your end, it probably looks good on our end too.

In the following we will refer to your virtual computer as Guest, and your actual computer as Host. Only step 4 requires you to do something within the Guest, in all the other steps you will either interact with vscode on your Host or UTM/VirtualBox.

We will use git and GitHub to synchronise files between the Guest and Host. You must set up your GitHub repository before following this guide. We also reccommend that you familiarise yourself with git and GitHub before setting up your VM.

Prerequisites:

- GitHub repository containing project
- Virtual machine host: VirtualBox 6.1[1]
- Appropriate virtual machine image: SU23-v0.ova[2]
- VS Code
- At least 5gb of free space

Setup:

1. Create a folder for your SU projects: `SU23`.

2. Create a file "settings.config" containing the following code snippet (keep the indentations). Put the file in the `SU23` folder:

```
Host SU23
  HostName 127.0.0.1
  User student
  Port 2222
```

3. In VS Code press `f1`, search for "ssh settings" and click `Remote-SSH: Settings`. Under the Remote.SSH: Config File input the path to the config file you just created.

4. Copy the virtual machine image into VirtualBox and boot up the virtual machine.
Leave your VM running in the background, and return to VS Code (ie. return to your `Host` computer)

[1] https://www.virtualbox.org/wiki/Downloads
[2] https://absalon.ku.dk/courses/64721/assignments/178351

5. In VS Code, Press `f1`, search for "ssh connect" and click `Remote-SSH: Connect to Host...`. It will suggest the name of the folder containing the .config file: `SU23`. It will ask you to choose between either continue/cancel or linux/windows/macOs, choose continue or linux and press `enter` to connect to the virtual machine via SSH.

6. In VS Code navigate to the menu and click `Terminal -> New Terminal` to open a terminal that can execute code on the `Guest`. Create a new folder for your local repository. Write `MKDIR SU23Guest` to the termial to create a folder named `SU23Guest`. CD into the folder.

7. To clone your repository into the guest write:
`git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY`. You will be prompted to verify the Guest on GitHub. Verify the `Guest` and allow the contents of the rpository to be downloaded.

8. The VM comes preloaded with DIKUArcade, but we need to move it into your local repository:

```
$ cp -R DIKUArcade/ SU23Guest/DIKUGames/
$ rm -rf DIKUArcade/
```

You can now interact with your code through the virtual machine. The `Guest` must be turned on and you must be connected via SSH (step 5).
You can update the code to the latest version available on GitHub by typing `git pull` in the console.
Apart from ebooting the VM you shouldn't need to interact with the `Guest` directly, instead always use the VS Code terminal on the `Host`.