

### Stap 1: Anaconda herinstalleren:

Omdat we op het werk Anaconda gebruiken en het dus al geïnstalleerd staat op mijn laptop zal ik toch met Anaconda werken. Ik heb wel mijn versie, die nog met Python 3.6 werkte, eraf gesmeten (uninstall via windows configuratie scherm > apps, dan Anaconda in de lijst zoeken en op verwijderen klikken). Daarna heb ik de nieuwe versie van Anaconda voor Windows met Python 3.7 geïnstalleerd.

### Stap 2: Conda environment creëren:

Anaconda Prompt opstarten en een nieuwe conda omgeving “ai1project” creëren:

```
(base) C:\Users\Gebruiker>conda create --name ai1project python=3.7
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Anaconda3\envs\ai1project

added / updated specs:
  - python=3.7
```

Dan volgt een overzicht van de packages die gaan geïnstalleerd worden tot er gevraagd wordt of je wil verdergaan, waarop we “y” antwoorden:

```
Proceed ([y]/n)? y

Downloading and Extracting Packages
ca-certificates-2020 | 125 KB | ##### | 100%
openssl-1.1.1d | 4.8 MB | ##### | 100%
sqlite-3.31.1 | 637 KB | ##### | 100%
python-3.7.6 | 14.8 MB | ##### | 100%
vs2015_runtime-14.16 | 1.1 MB | ##### | 100%
wheel-0.34.2 | 66 KB | ##### | 100%
pip-20.0.2 | 1.7 MB | ##### | 100%
certifi-2019.11.28 | 154 KB | ##### | 100%
setuptools-45.1.0 | 527 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate ai1project
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

Dan gaan we naar de folder waarin ons PyCharm project zal gecreëerd worden en we activeren de nieuwe “ai1project” environment:

```
(base) C:\Users\Gebruiker>d:
(base) D:\>cd tibl\artificial_intelligence_kdg\artificial_intelligence_1\project
(base) D:\TIBL\artificial_intelligence_kdg\artificial_intelligence_1\project>conda activate ai1project
```

En we checken de Python versie:

```
(ai1project) D:\TIBL\artificial_intelligence_kdg\artificial_intelligence_1\project>python --version
Python 3.7.6
```

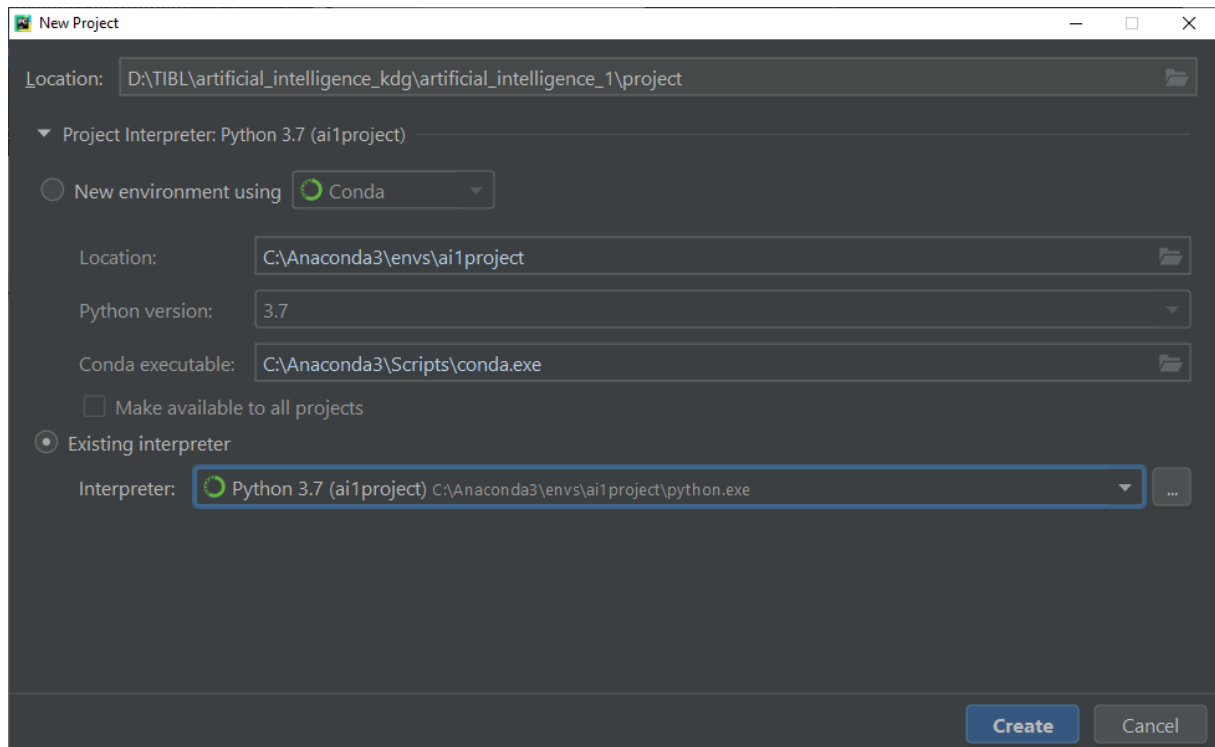
### Stap 3: PyCharm project aanmaken:

Ook PyCharm staat al geïnstalleerd op mijn laptop en dat heb ik zo gelaten. Na installatie van Anaconda kreeg ik een melding dat er een PyCharm versie is die speciaal voor Anaconda werd ontwikkeld, maar die ga ik voorlopig niet gebruiken. Ik heb wel mijn geïnstalleerde versie ge-updatet.

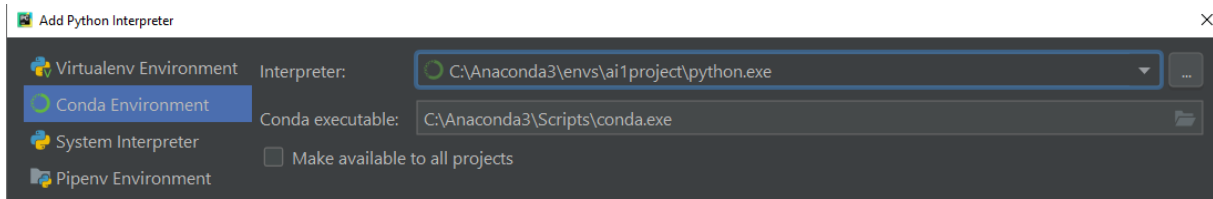
Bij het opstarten van PyCharm kies ik de optie "Create New Project":



In het volgende venster geven we bij "Location" de folder op waarin we ons project willen aanmaken. We kiezen voor "Existing interpreter", klikken op de 3 puntjes om de python.exe te selecteren van de Conda environment die we in stap 2 hebben aangemaakt.



Wanneer we op de 3 puntjes naast het invulvakje van “Interpreter” klikken, krijgen we het volgende scherm waarin we links voor “Conda Environment” kiezen en dan de juiste python.exe en conda.exe selecteren.



We klikken op “ok” en zien dat in het hoofdscherm de juiste python.exe staat aangegeven onder “Existing interpreter”. We klikken nu op “Create”. De PyCharm IDE wordt nu opgestart en met ons aangemaakte project ingeladen.

Wanneer we een terminal openen (links onderaan), dan zitten we in de juiste directory en de juiste Conda environment. Met “dir” checken we de inhoud van de directory:

```
Terminal: Local x +

(ailproject) D:\TIBL\artificial_intelligence_kdg\artificial_intelligence_1\project>dir
Volume in drive D is DATA
Volume Serial Number is 0465-F39A

Directory of D:\TIBL\artificial_intelligence_kdg\artificial_intelligence_1\project

09/02/2020  19:22    <DIR>        .
09/02/2020  19:22    <DIR>        ..
09/02/2020  19:22    <DIR>        .idea
```

#### Stap 4: OpenAI Gym installeren

In de terminal installeren we OpenAI Gym met “pip install” (zie ook <http://gym.openai.com/docs/>):

```
(ailproject) D:\TIBL\artificial_intelligence_kdg\artificial_intelligence_1\project>pip install gym
Collecting gym
```

Er volgt dan een overzicht van alle packages die worden geïnstalleerd, waaronder numpy en scipy.

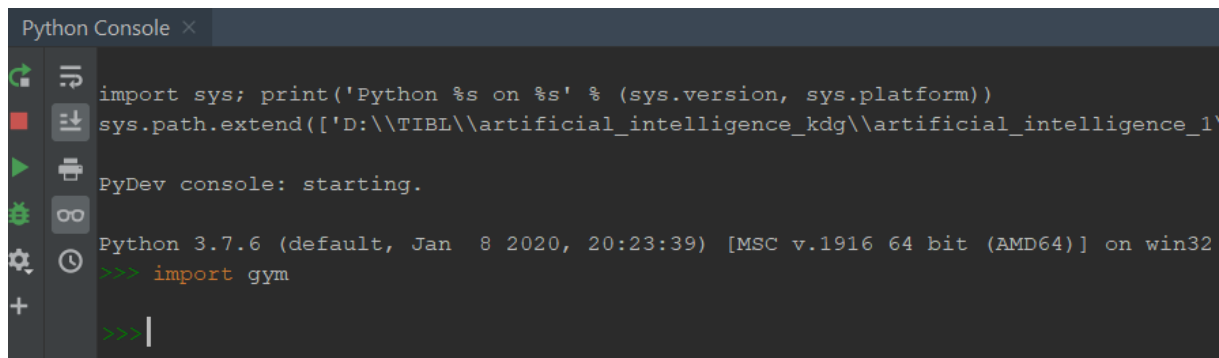
Op het einde krijgen we dan de boodschap dat de installatie gelukt is:

```
Successfully built gym future
Installing collected packages: numpy, scipy, six, future, pygame, cloudpickle, gym
Successfully installed cloudpickle-1.2.2 future-0.18.2 gym-0.15.6 numpy-1.18.1 pygame-1.4.10 scipy-1.4.1 six-1.14.0

(ailproject) D:\TIBL\artificial_intelligence_kdg\artificial_intelligence_1\project>
```

Merk op dat installatie via conda niet lukte.

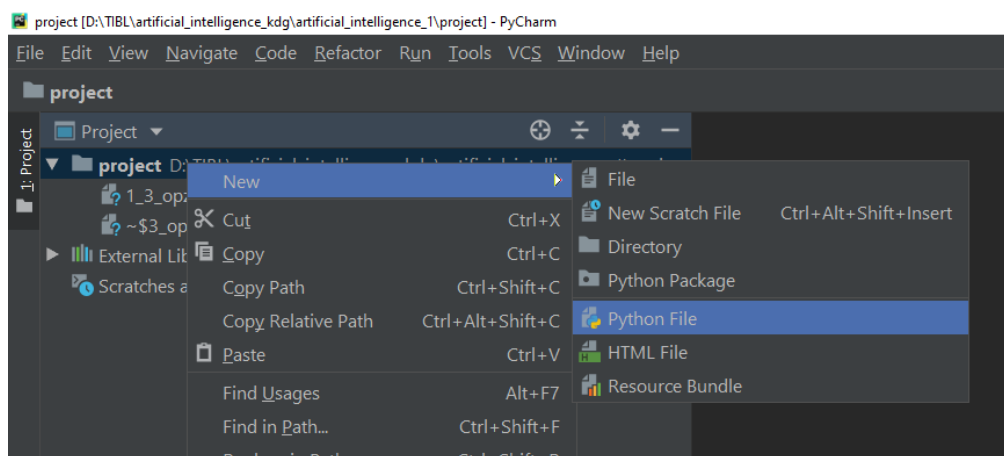
We checken of de installatie is gelukt door links onderaan een “Python Console” te openen en “gym” te importeren:



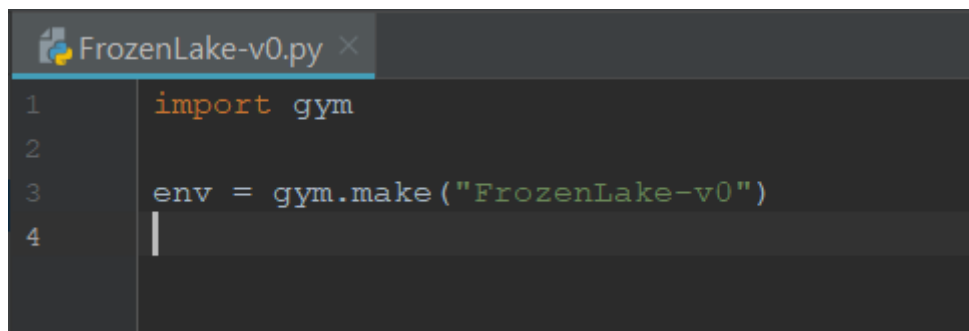
```
Python Console x
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\TIBL\\artificial_intelligence_kdg\\artificial_intelligence_1\\'])
PyDev console: starting.
Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)] on win32
>>> import gym
>>> |
```

### Stap 5: FrozenLake-v0 Environment aanmaken:

We maken een nieuwe Python file aan door rechts te klikken op de projectfolder:

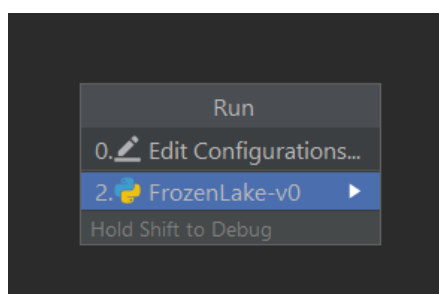


We geven de file de naam “FrozenLake-v0.py”. We importeren “gym” en maken de environment aan met “gym.make()”:

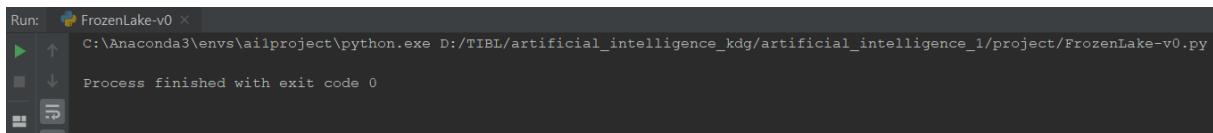


```
FrozenLake-v0.py x
1 import gym
2
3 env = gym.make("FrozenLake-v0")
4 |
```

We runnen de file via het menu “Run > Run...” en kiezen voor de file die we net hebben aangemaakt:



In het “run” venster onderaan krijgen we de boodschap “Process finished with exit code 0” als dit gelukt is:

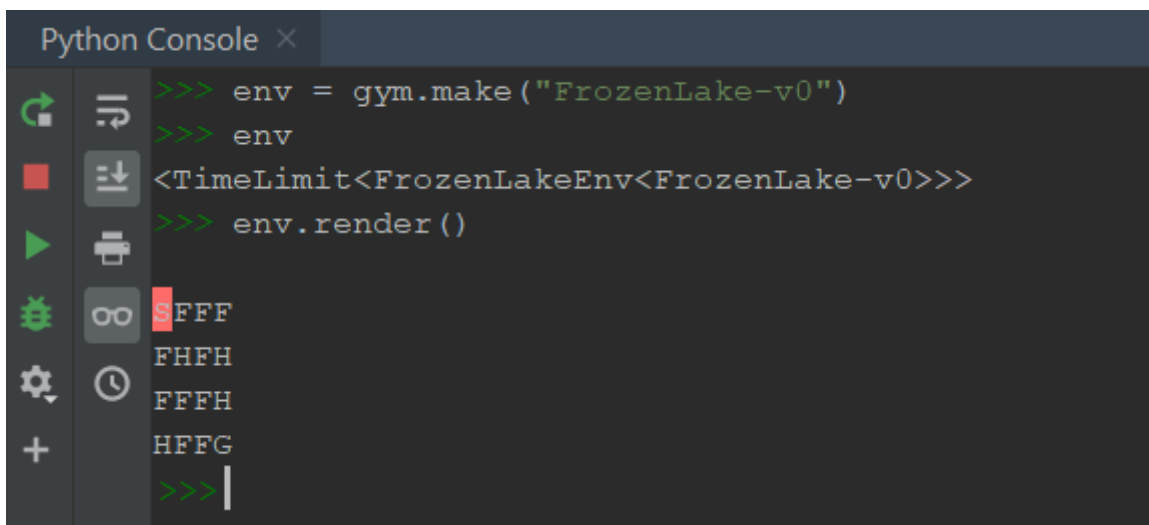


```
Run: FrozenLake-v0 x
C:\Anaconda3\envs\ailproject\python.exe D:/TIBL/artificial_intelligence_kdg/artificial_intelligence_1/project/FrozenLake-v0.py
Process finished with exit code 0
```

OPGELET! FrozenLake is een OpenAI Gym environment waarvoor we een agent zullen ontwikkelen en mag dus niet verward worden met de Conda environment!

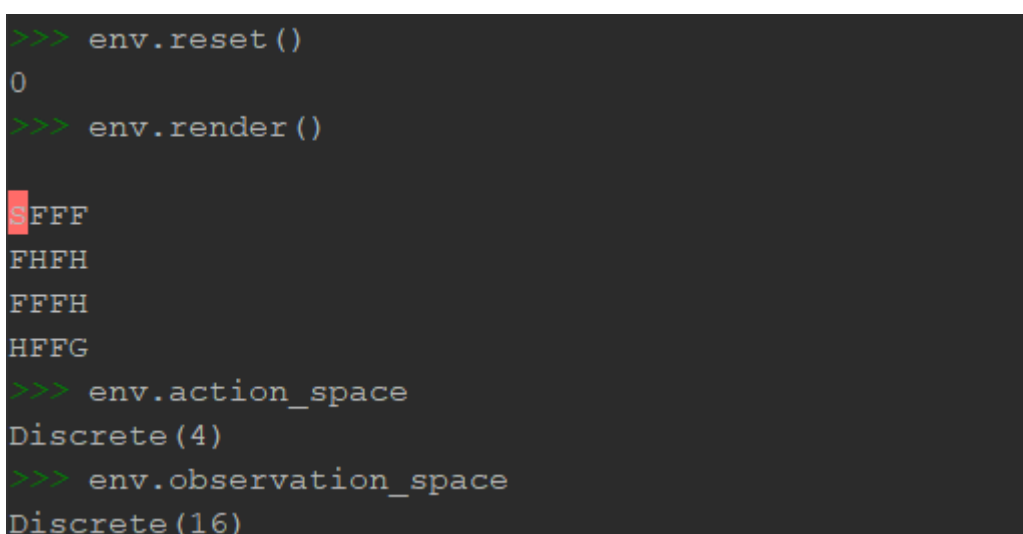
### Stap 6: FrozenLake Environment verkennen:

We maken de “FrozenLake-v0” environment aan in de console en roepen de methode “render” aan om de omgeving te zien:



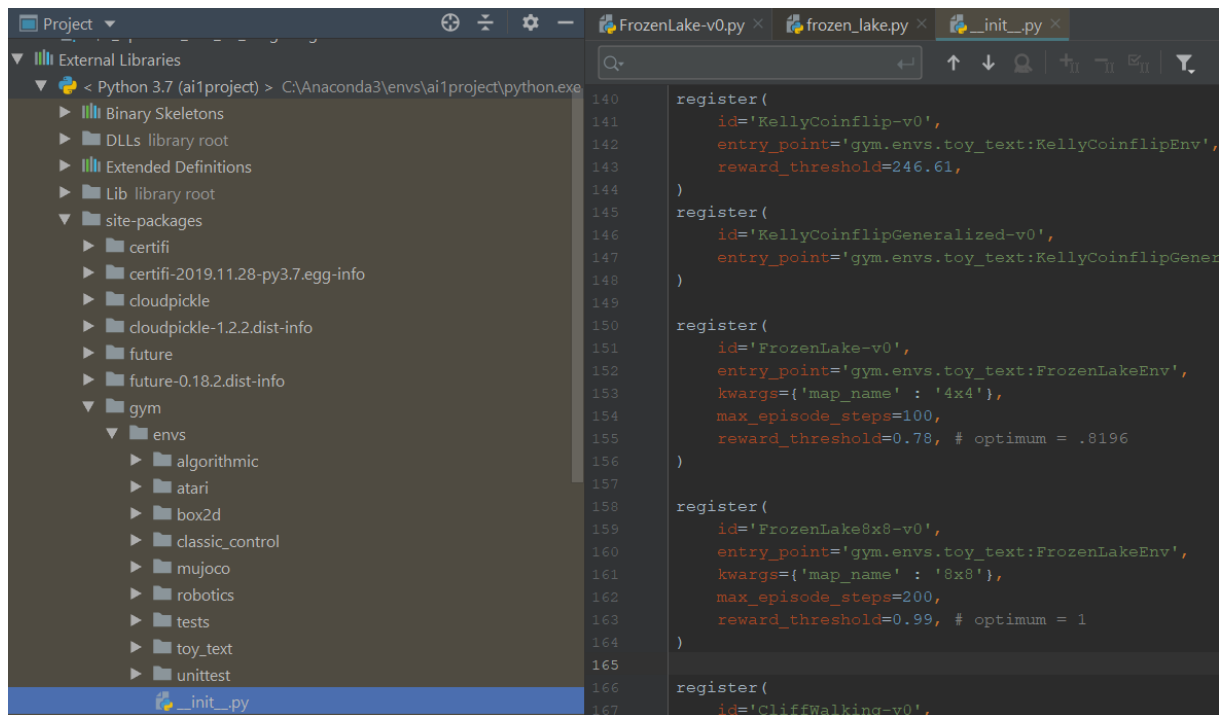
```
Python Console x
>>> env = gym.make("FrozenLake-v0")
>>> env
<TimeLimit<FrozenLakeEnv<FrozenLake-v0>>>
>>> env.render()
SFFF
FHFH
FFFH
HFEG
>>> |
```

De omgeving resetten doen we met de methode “reset”. De property “action\_space” geeft het aantal acties weer, 4 in dit geval, en de property “observation\_space” het aantal states, en dat zijn er 16:



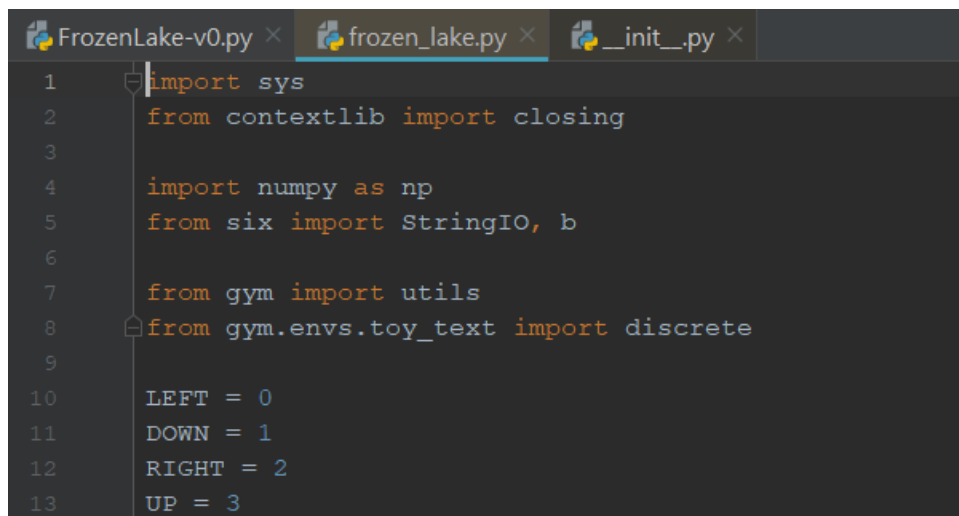
```
>>> env.reset()
0
>>> env.render()
SFFF
FHFH
FFFH
HFEG
>>> env.action_space
Discrete(4)
>>> env.observation_space
Discrete(16)
```

In de file “\_\_init\_\_.py” van de gym package, die we onder “External libraries vinden”, zoeken we op “FrozenLake” om de source terug te vinden:



De sourcefile vinden we dus in de folder “gym/envs/toy\_text”. Merk op dat het ook mogelijk is om ook een FrozenLake omgeving van 8 op 8 te nemen.

In de sourcefile “FronzenLake.py” zien we dan de code om naar boven, onder, links en rechts te bewegen:



Laten we nu eens naar onder (code 1) bewegen met de methode “step” en de omgeving opnieuw renderen. Methode “step” retourneert een tuple met 4 elementen: de nieuwe positie, de reward, een boolean die aangeeft of het spel al dan niet is beëindigd, en extra info die we voorlopig niet nodig hebben. Blijkbaar zijn we uitgedegen en daardoor staan we nu op positie 1, rechts van de startpositie 0, ipv op positie 4, onder de startpositie. De reward is 0.0, dus we hebben geen beloning gekregen. En de “False” geeft aan dat het spel nog niet is gestopt, dus we zijn niet in een gat H gevallen, en we staan uiteraard ook nog niet op de eindpositie G.

```

>>> env.step(1)
(1, 0.0, False, {'prob': 0.3333333333333333})
>>> env.render()
      (Down)
S FFF
FHFH
FFFH
HFFG

```

In de sourcefile vinden we ook nog de uitleg terug van hoe het spel moet gespeeld worden:

```

71 class FrozenLakeEnv(discrete.DiscreteEnv):
72     """
73     Winter is here. You and your friends were tossing around a frisbee at the park
74     when you made a wild throw that left the frisbee out in the middle of the lake.
75     The water is mostly frozen, but there are a few holes where the ice has melted.
76     If you step into one of those holes, you'll fall into the freezing water.
77     At this time, there's an international frisbee shortage, so it's absolutely imperative that
78     you navigate across the lake and retrieve the disc.
79     However, the ice is slippery, so you won't always move in the direction you intend.
80     The surface is described using a grid like the following
81
82         SFFF
83         FHFH
84         FFFH
85         HFFG
86
87     S : starting point, safe
88     F : frozen surface, safe
89     H : hole, fall to your doom
90     G : goal, where the frisbee is located
91
92     The episode ends when you reach the goal or fall in a hole.
93     You receive a reward of 1 if you reach the goal, and zero otherwise.
94
95     """

```

Nu vullen we ons script aan met code om het spel te spelen:

```

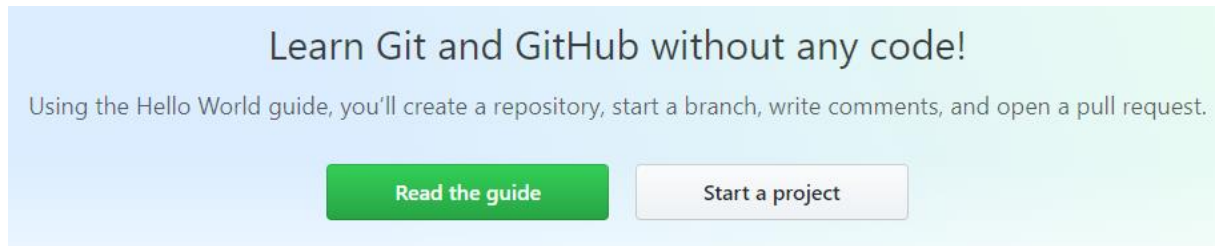
FrozenLake-v0.py x
1  # OpenAI Gym importeren
2  import gym
3
4  # FrozenLake environment aanmaken
5  env = gym.make("FrozenLake-v0")
6
7  # aantal actions en states
8  print("Action space: ", env.action_space)
9  print("Observation space: ", env.observation_space)
10
11 # environment resetten
12 env.reset()
13 env.render()
14
15 # speel het spel...
16 maxiter = 10
17 for i in range(maxiter):
18     random_action = env.action_space.sample()
19     new_state, reward, done, info = env.step(random_action)
20     env.render()
21     if done:
22         break

```

## Stap 7: Git repository opzetten:

Ik heb al GIT geïnstalleerd op mijn laptop en ik heb ook een GitHub account, dus ik ga een nieuwe repository op GitHub aanmaken:

Na inloggen op github.com, klikken we op “Start a project”:




En dan maken we de nieuwe repo aan, die we “AI1project” noemen:

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner

 alouwyck ▾

Repository name \*

/ AI1project ✓

Great repository names are short and memorable. Need inspiration? How about **potential-octo-broccoli**?

Description (optional)

Project - course Artificial Intelligence 1 - TIBL KdG

☐



Public

Anyone can see this repository. You choose who can commit.

☒



Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾

Add a license: None ▾

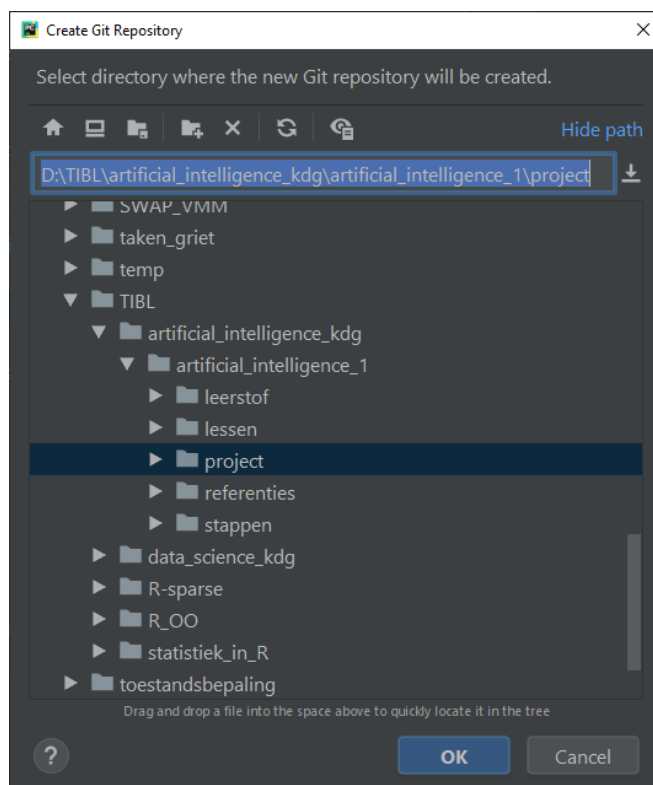
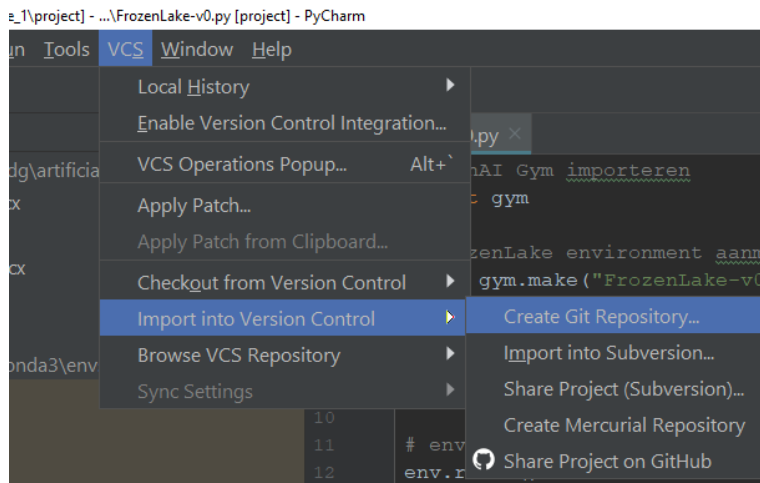


Create repository

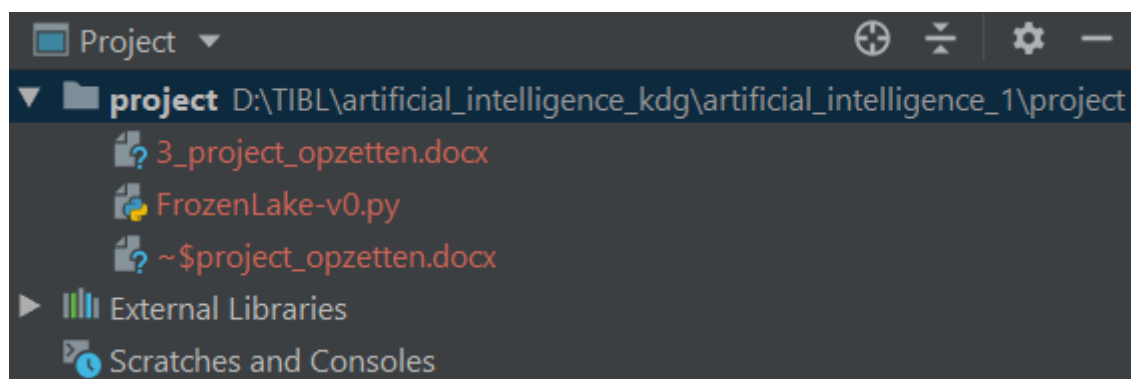
In Pycharm gaan we nu eerst een lokale repo opzetten. Hiervoor gaan we naar menu “VCS > Import into Version Control > Create Git Repository...”

In het venster checken we of onze projectfolder is opgegeven en dan klikken we op “Ok”.





In het Project venster zien we nu dat de files in de projectfolder rood zijn geworden.

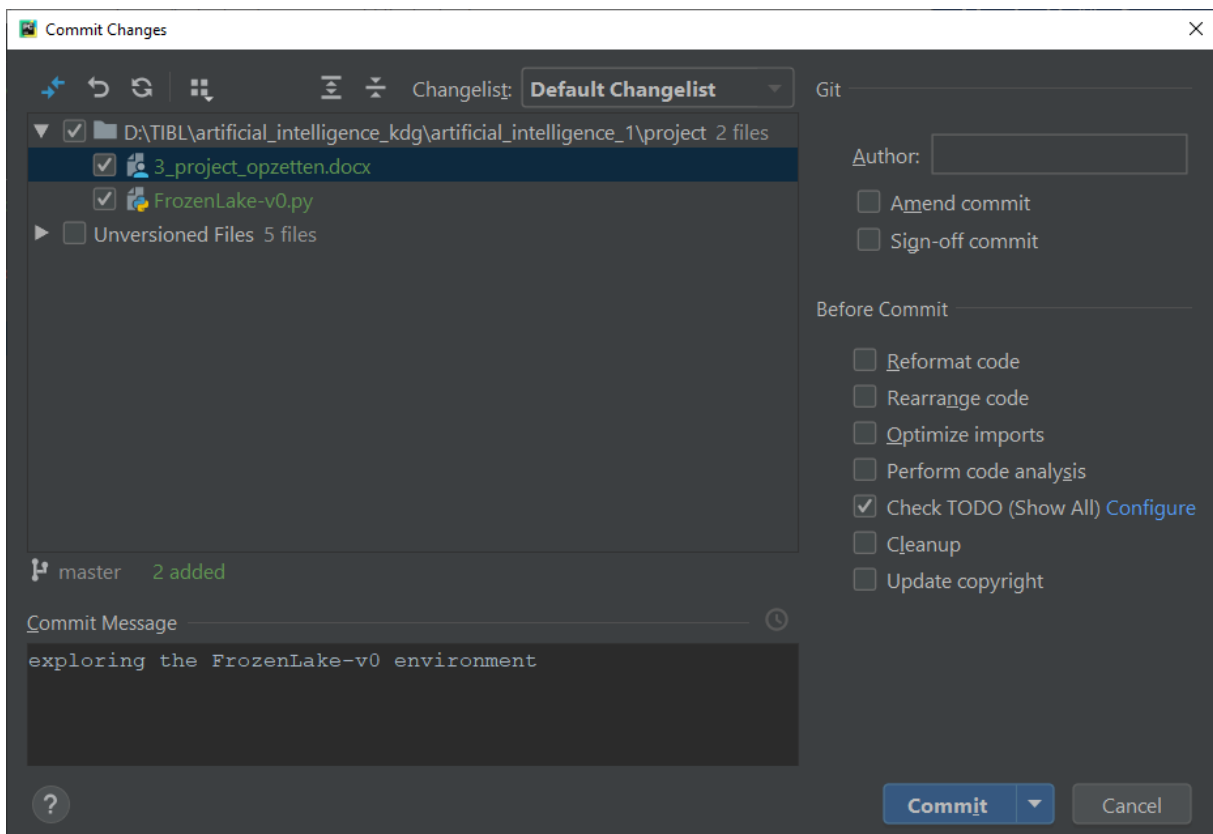


We gaan dit document (3\_project\_opzetten.docx) en ons script (FrozenLake-v0.py) toevoegen door de files te selecteren, rechts te klikken, en in het menu dat verschijnt te kiezen voor “Git > +Add”. De files zijn nu groen geworden.

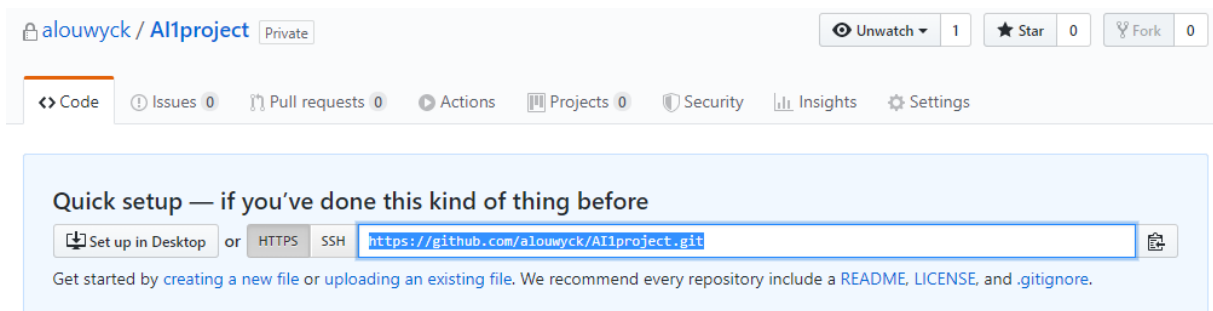
Nu gaan we een commit uitvoeren via het menu “VCS > Commit...” of door op het groene vinkje te klikken in het snelmenu “Git:”

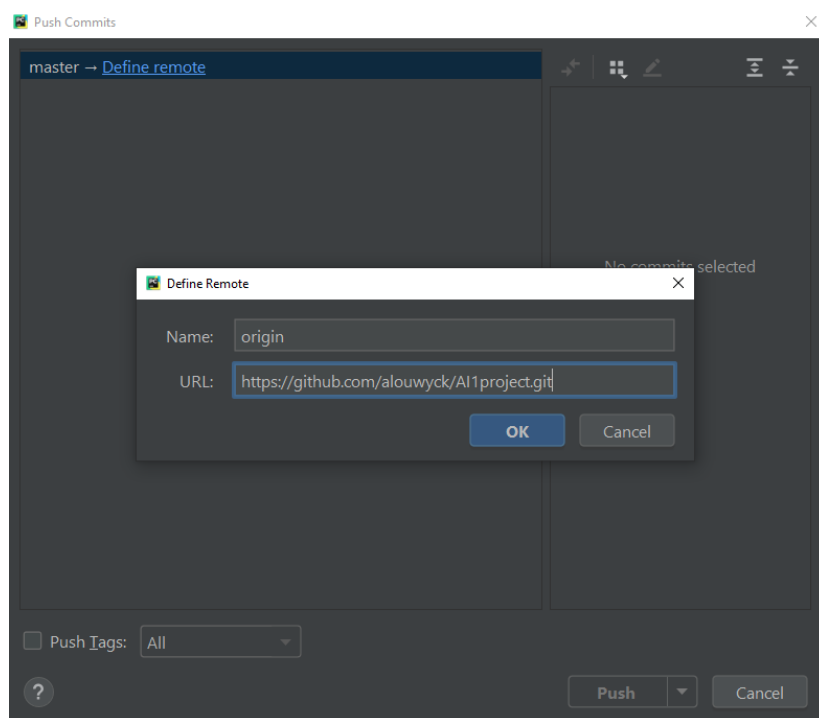
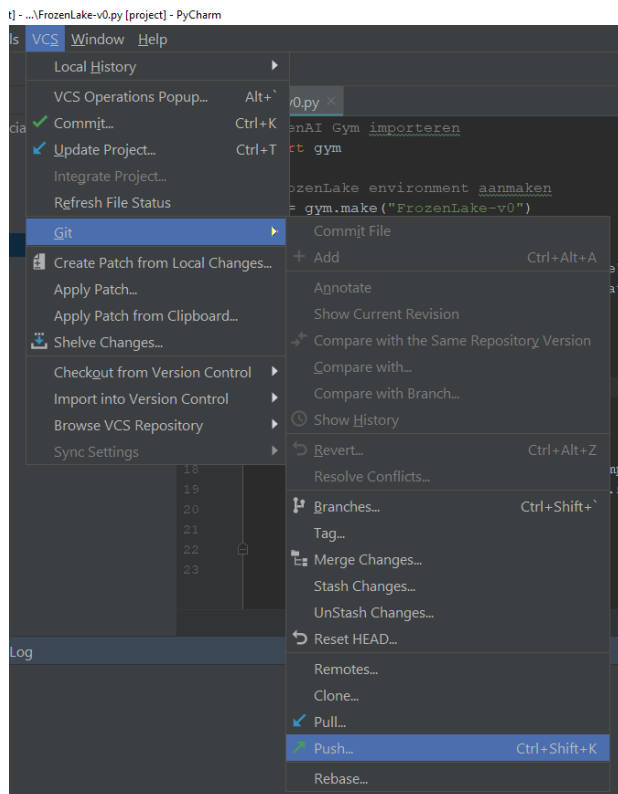


In het venster dat verschijnt vullen we een “Commit Message” in en we klikken op “Commit”:



Nu hebben we de twee files toegevoegd aan de lokale repo in de projectfolder. Die repo gaan we nu pushen naar de remote GitHub repo. We gaan naar menu “VCS > Git > Push...” en in het venster dat verschijnt klikken we op “define remote” en we vullen de url van onze GitHub repo in en klikken op “Push”. De url vinden we terug op de startpagina van ons GitHub project:





In GitHub zien we nu onder tabblad “Code” dat de twee files effectief zijn toegevoegd:

alouwyck / AI1project Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Security Insights Settings

Project - course Artificial Intelligence 1 - TIE

Manage topics

1 commit

Branch: master New pull request

Create new file Upload files Find file Clone or download

alouwyck exploring the FrozenLake-v0 environment Latest commit e2b8ab9 11 minutes ago

|                         |   |                |
|-------------------------|---|----------------|
| 3_project_opzetten.docx | exploring the FrozenLake-v0 environment | 11 minutes ago |
| FrozenLake-v0.py        | exploring the FrozenLake-v0 environment | 11 minutes ago |

Add a README with an overview of your project. Add a README

Tenslotte voegen we de docent toe als collaborator via het tabblad “Settings” en menu “Collaborators”:

alouwyck / AI1project Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Security Insights Settings

Options

Collaborators

Branches

Webhooks

Notifications

Integrations & services

Deploy keys

Collaborators Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

DekwoKybon Add collaborator

0 of 3 collaborators