

# demo\_explorer

November 23, 2020

```
[243]: import pickle
import blosc
import numpy as np
import matplotlib.pyplot as plt

with open('demos/demos.pkl', 'rb') as f:
    data = pickle.load(f)
```

```
[244]: print(f"number of demos: {len(data)}")
```

number of demos: 100

```
[245]: mission_id = np.random.randint(0, len(data))
print(f"Selected mission {mission_id}")
```

Selected mission 11

```
[246]: # the instruction (in this case it is always the same)
data[mission_id][0]
```

```
[246]: 'open the door'
```

```
[247]: # this is the numeric state of the mission
numeric_state = blosc.unpack_array(data[mission_id][2])

# shape: number_of_steps x grid_size X grid_size X object_encoding
numeric_state.shape
```

```
[247]: (15, 22, 22, 3)
```

```
[253]: # Example initial numeric state (step=0)
print(numeric_state[0])
plt.imshow(numeric_state[0]/10)
plt.show()
```

```
[[[2 5 0]
    [2 5 0]
    [2 5 0]
```

...  
 [2 5 0]  
 [2 5 0]  
 [2 5 0]]

[[2 5 0]  
 [1 0 0]  
 [1 0 0]

...  
 [1 0 0]  
 [1 0 0]  
 [2 5 0]]

[[2 5 0]  
 [1 0 0]  
 [1 0 0]

...  
 [1 0 0]  
 [1 0 0]  
 [2 5 0]]

...

[[2 5 0]  
 [1 0 0]  
 [1 0 0]

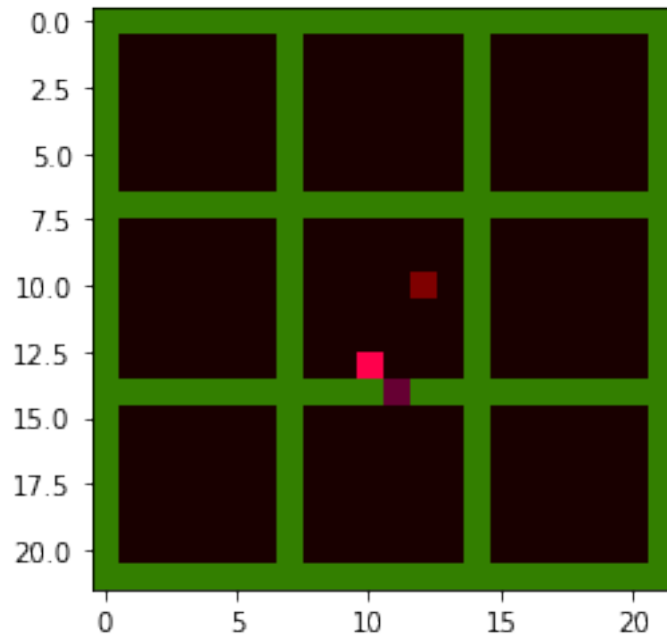
...  
 [1 0 0]  
 [1 0 0]  
 [2 5 0]]

[[2 5 0]  
 [1 0 0]  
 [1 0 0]

...  
 [1 0 0]  
 [1 0 0]  
 [2 5 0]]

[[2 5 0]  
 [2 5 0]  
 [2 5 0]

...  
 [2 5 0]  
 [2 5 0]  
 [2 5 0]]]



```
[249]: # this is an additional compass-sensor (called direction) which the agent could
        ↪ use to navigate
        # not sure if this is useful at this point
        data[mission_id][3]
```

```
[249]: [2, 3, 2, 1, 1, 1, 2, 2, 2, 2, 3, 3, 0, 0, 0]
```

```
[250]: # This is the sequence of actions the agent took
        actions = list(map(lambda x: x.name, data[mission_id][4]))
        actions
```

```
[250]: ['right',
        'left',
        'left',
        'forward',
        'forward',
        'right',
        'forward',
        'forward',
        'pickup',
        'right',
        'forward',
        'right',
        'forward',
        'forward',
        'toggle']
```

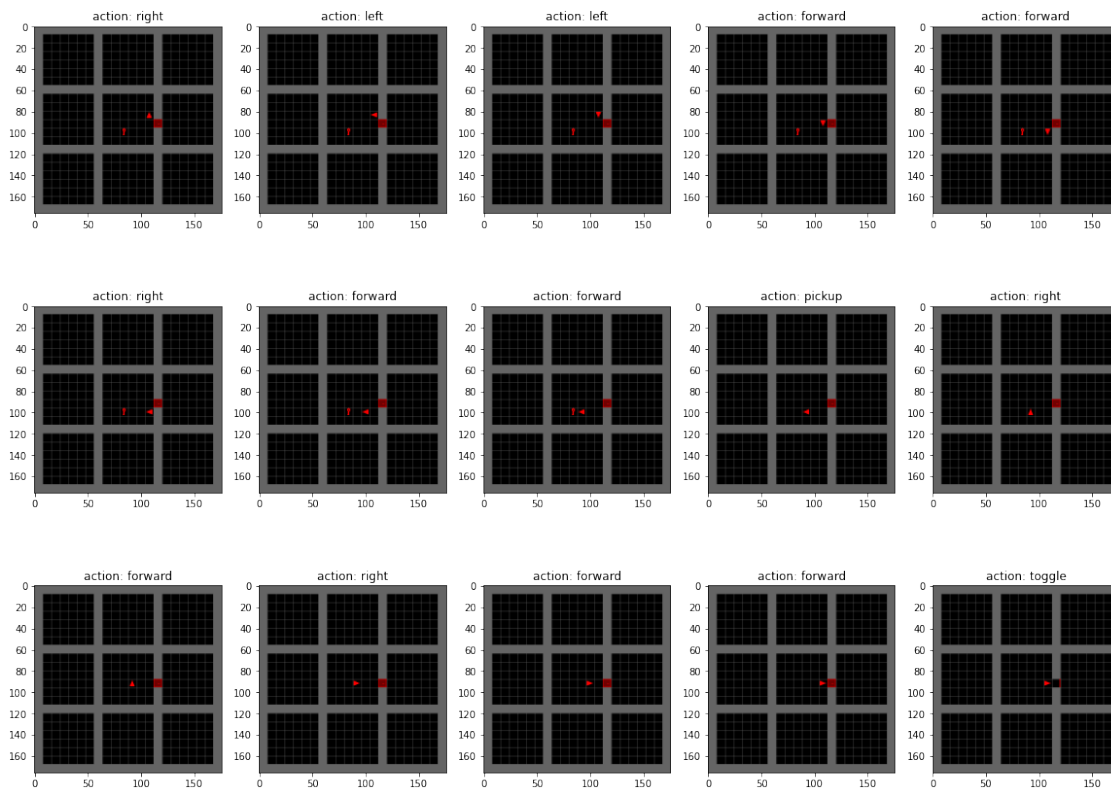
```
[251]: # In order to allow to explore what is actually done, this allows the view a
↳graphical rendering of the sequence of steps taken
img = blosc.unpack_array(data[mission_id][1])
```

```
[252]: rows = int(np.ceil(len(img)/5))

plt.figure(figsize=(20,5*rows))

for i in range(len(img)):
    plt.subplot(rows,5,i+1)
    plt.title(f"action: {actions[i]}")
    plt.imshow(img[i])

plt.show()
```



```
[ ]:
```