In [1]:
```
from IPython.display import clear_output
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
num_of_wumpus = 1
num_of_holes = 3
```

In [3]:
```
def end_game(dangers, pos):
    alive = True
    for danger_pos in dangers.T:
        if pos[0] == danger_pos[0] and pos[1] == danger_pos[1]:
            alive = False
            clear_output()
            plt.figure(figsize = (4,4))
            plt.plot(0,0, color = 'k', marker = 's', markersize = 3000)
            plt.axis('off')
            plt.title('FAIL')
            plt.show()
    return alive
```

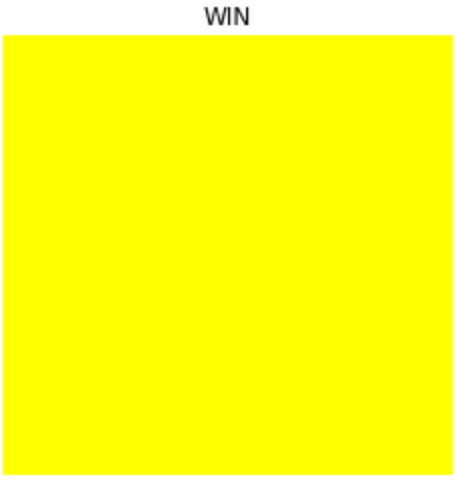
In [4]:
```
def win_game(pos, gold):
    win = False
    if sum(np.array(pos)==np.array(gold))==2:
        win = True
        clear_output()
        plt.figure(figsize = (4,4))
        plt.plot(0,0, color = 'yellow', marker = 's', markersize = 3000
        plt.axis('off')
        plt.title('WIN')
        plt.show()
    return win
```

In [9]:
```
def plot_game(pos, wumpus, holes, gold, width = 5, height = 5):
    clear_output()
    plt.figure(figsize = (4, 4))
    plt.plot([pos[0]], [pos[1]], linestyle ='', marker = '*', markersiz
    plt.plot([gold[0]], [gold[1]], linestyle ='', marker = 'o', markers
    plt.plot([wumpus[0]], [wumpus[1]], linestyle = '', marker = 'x', ma
    plt.plot([holes[0]], [holes[1]], linestyle = '', marker = 'o', mark
    plt.xlim(-width-1, width+1)
    plt.ylim(-height-1, height+1)
    plt.axis('off')
    plt.show()
```

```
GOLD
```

In [9]:

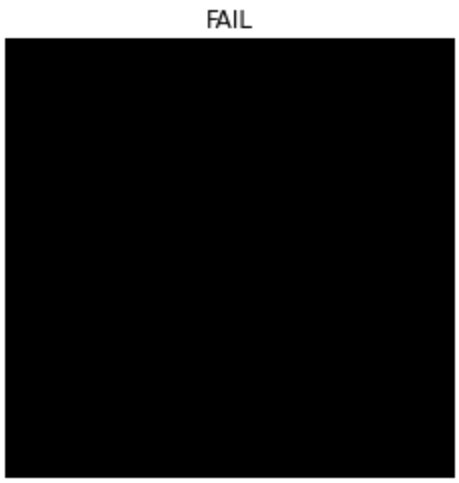
```
gold = np.random.randint(low = -5, high = 6, size = 2)
pos = [0,0]
wumpus = np.random.randint(low = -5, high = 6, size = (2, num_of_wumpus
holes = np.random.randint(low = -5, high= 6, size = (2, num_of_holes))
alive = True
win = False

while alive and not win:

    plot_game(pos, wumpus, holes, gold)

    alive = end_game(wumpus, pos) and end_game(holes, pos)
    if alive : win = win_game(pos, gold)
    if alive and not win:
        # moving the player
        direction = input('Which Direction? u/d/l/r and Enter:')
        if len(direction)<1:
            clear_output()
            continue
        else :
            direction = direction[0]

        if   direction == 'u':
            pos[1] += 1
        elif direction == 'd':
            pos[1] -= 1
        elif direction == 'l':
            pos[0] -= 1
        elif direction == 'r':
            pos[0] += 1
        else : break

```




```
PIT
```

In [12]:

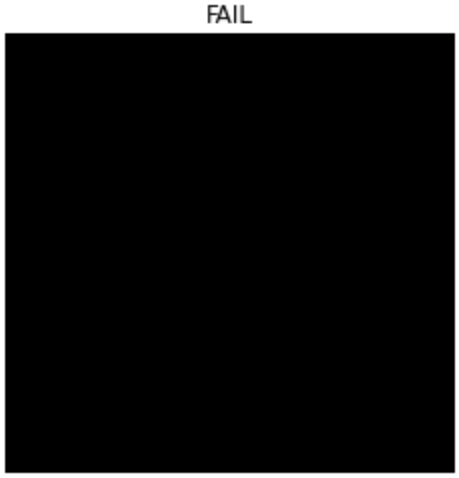
```
gold = np.random.randint(low = -5, high = 6, size = 2)
pos = [0,0]
wumpus = np.random.randint(low = -5, high = 6, size = (2, num_of_wumpus
holes = np.random.randint(low = -5, high= 6, size = (2, num_of_holes))
alive = True
win = False

while alive and not win:

    plot_game(pos, wumpus, holes, gold)

    alive = end_game(wumpus, pos) and end_game(holes, pos)
    if alive : win = win_game(pos, gold)
    if alive and not win:
        # moving the player
        direction = input('Which Direction? u/d/l/r and Enter:')
        if len(direction)<1:
            clear_output()
            continue
        else :
            direction = direction[0]

        if   direction == 'u':
            pos[1] += 1
        elif direction == 'd':
            pos[1] -= 1
        elif direction == 'l':
            pos[0] -= 1
        elif direction == 'r':
            pos[0] += 1
        else : break

```




```
WUMPUS
```

In [13]:

```
gold = np.random.randint(low = -5, high = 6, size = 2)
pos = [0,0]
wumpus = np.random.randint(low = -5, high = 6, size = (2, num_of_wumpus
holes = np.random.randint(low = -5, high= 6, size = (2, num_of_holes))
alive = True
win = False

while alive and not win:

    plot_game(pos, wumpus, holes, gold)

    alive = end_game(wumpus, pos) and end_game(holes, pos)
    if alive : win = win_game(pos, gold)
    if alive and not win:
        # moving the player
        direction = input('Which Direction? u/d/l/r and Enter:')
        if len(direction)<1:
            clear_output()
            continue
        else :
            direction = direction[0]

        if   direction == 'u':
            pos[1] += 1
        elif direction == 'd':
            pos[1] -= 1
        elif direction == 'l':
            pos[0] -= 1
        elif direction == 'r':
            pos[0] += 1
        else : break

```



In [ ]:

```

```