This notebook allows the loading or creation of *enemy* NPC entities and semi-automates a fight scene in the roleplaying game Naheulbeuk.

Dependencies

In [1]:
```
# Live Object Info
import inspect

# UNIX pathname handling
import glob

# Interactive Control
import ipywidgets as widgets
from ipywidgets import interact, interact_manual
```

Load Enemy Class

In [2]:
```
from NPC import enemy
```

Refer to Naheulbeuk User Guide for attribute details.

In [3]:
```
print(inspect.signature(enemy));
#print(inspect.getfullargspec(enemy));
```

```
(NAME, AT, PRD, EV, PR, WEAPON, DMG_DICE_N, DMG_BONUS, COU, EXP)
```

Functions

In [4]:
```
def everyone_attack():
    # Attack
    for opponent in enemies:
        opponent.attack();

    # If counter-attacked
    print("--------------------");
    print("If counter-attacked:\n");
    for opponent in enemies:
        opponent.counter();

    print("\nEnemies: ",[(bad_guy.NAME,bad_guy.EV,bad_guy.state) for bad_guy :

actions=["Defend","Kill","Heal"]
def process_action():
    # TODO: opponent should not be global, but argument.
    # TODO: Should action be argument too?
    if action not in actions:
        print("Unknown action ["+action+"].")
    if action=="Attack":
        opponent.attack();
        opponent.counter();
    if action=="Defend" and not crit:
        opponent.defend(hit);
    if action=="Defend" and crit:
        opponent.take_damage(hit);
        opponent.next_phase();
    if action=="Kill":
        opponent.EV = 0
        opponent.next_phase();
    if action=="Heal":
        opponent.gain_health(hit);
    # TODO: Add Revive

def enemy_stats():
    for opponent in enemies:
        print('\033[30m'"--- ",opponent.NAME," ---")
        # TODO: Colour depending on state
        print('\033[32m'"EV = ", opponent.EV,"PR = ",opponent.PR, "Statut = "
```

Initialization

Reading enemies

Load the NPCs defined in a file or define your own.

NOTE: Do not launch following cell again once the fight has started at the risk of losing all progress.

In [5]:
```
# TODO: Add toggle to start a new fight or load new enemies into ongoing fight

# Load an encounter from file, or
# TODO: Add interactive selection from encounters folder
filename = "Sample_Encounter_2"
# filename = "encounters/Scenario_Gardes_Entree_Chateau"
# filename = "encounters/Scenario_Gardes_Patrouilles"
# filename = "encounters/Scenario_Valtordu_Chef_de_Garde"
# filename = "encounters/Scenario_Rats_Pesteux"
# filename = "encounters/Scenario_Valtordu_Chapichapos"
# filename = "encounters/Scenario_Chnafon_Gronick"
# filename = "encounters/Scenario_Chnafon_Ruflak"

# Define your own encounter, or
# TODO: Add sample for in-line encounter definition here (instead of encounter
# TODO: Add interactive enemy adding

# Load a previously saved state (default = save_file)
# TODO: 1 save folder/file per encounter file loaded so individual encounters
# filename = "save_file"
```

In [6]:
```
# Check and prepare file for loading saved encounter
if filename in ["saved_encounter", "save_file"]:
    filename = "current_save"
    relative_path = "save_file/"
    states = glob.glob(relative_path+'saved_*')
    states = [s[len(relative_path):] for s in states]

    compiled_save = str(len(states)) + " (NAME, AT, PRD, EV, PR, WEAPON, DMG_D
    for save in states:
        with open(relative_path+save, 'r') as file:
            compiled_save += "\n" + file.readline()

    with open(filename, 'w') as file:
        file.write(compiled_save)

# Create list of enemies
enemies = []

with open(filename, 'r') as file:
    x = file.readline().split();
    try:
        nb_enemies = int(x[0])
    except ValueError:
        print("First line should be number of enemies followed by legend.")

    for i in range(nb_enemies):
        # Read enemy (1 line per enemy)
        NAME, AT, PRD, EV, PR, WEAPON, DMG_DICE_N, DMG_BONUS, COU, EXP = file
        # Convert specific variables to int for further use
        AT, PRD, EV, PR, DMG_DICE_N, DMG_BONUS, COU, EXP = \
            int(AT), int(PRD), int(EV), int(PR), int(DMG_DICE_N), int(DMG_BONU
        # Create enemy instance
        myVars = vars()
        myVars[NAME] = enemy(NAME, AT, PRD, EV, PR, WEAPON, DMG_DICE_N, DMG_BO
        enemies.append(myVars[NAME])

print("These enemies were loaded:\n")
enemy_dict = {}
for opponent in enemies:
    enemy_dict.update({opponent.NAME:opponent})
    print(opponent.info(),"\n")
```

```
These enemies were loaded:

Bertrand a 11 en attaque, 10 en parade, 32 en EV, et une protection de 2.
Armé de Rapiere, qui fait des dégâts 1D6+3.
Ils ont un courage de 13.
Ils donnent 15 XP en mourant.

Garde_1 a 10 en attaque, 8 en parade, 18 en EV, et une protection de 3.
Armé de Lance/Bouclier, qui fait des dégâts 1D6+3.
Ils ont un courage de 12.
Ils donnent 10 XP en mourant.

Garde_2 a 11 en attaque, 9 en parade, 17 en EV, et une protection de 3.
Armé de Epee/Bouclier, qui fait des dégâts 1D6+2.
Ils ont un courage de 13.
Ils donnent 10 XP en mourant.

Garde_3 a 10 en attaque, 10 en parade, 19 en EV, et une protection de 3.
Armé de Epee_Rouille, qui fait des dégâts 1D6+2.
Ils ont un courage de 11.
Ils donnent 11 XP en mourant.
```

Check Current Status

Click on "Check Current Status" to check current enemies status at any time during the fight

In [7]:
```
# Check Current Status
manual_status = interact_manual.options(manual_name="Check Current Status")
manual_status(enemy_stats);
```

Check Current Status

Fight Sequence

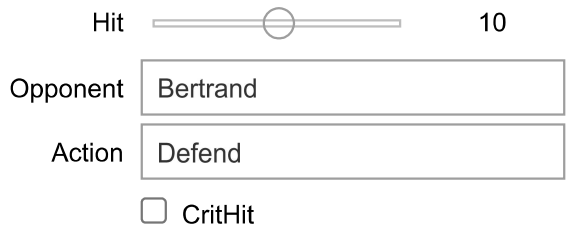
Attacking: All NPC enemies attack at once. Players need to roll dice to block or counter.
Defending: Player hit needs to be input before "Process Action".
Check Current Status: Click on "Check Current Status" to check current enemies status.

In [8]:
```
# TODO: Do dead enemies need to be removed from dropdown???
@interact
def enemy_update(Hit=(0,20,1),
                 Opponent=enemy_dict,
                 Action=actions,
                 CritHit=False):
    print(Opponent.NAME)
    if Opponent.state != "ALIVE":
        print("This opponent is", Opponent.state)
    print("EV =", Opponent.EV,"PR =",Opponent.PR)
    global hit,opponent,action,crit
    opponent=Opponent
    hit=Hit
    action=Action
    crit=CritHit

do_action = interact_manual.options(manual_name="Process Action")
do_action(process_action);

all_attack = interact_manual.options(manual_name="Everyone Attack!")
all_attack(everyone_attack);

# Check Current Status
manual_status = interact_manual.options(manual_name="Check Current Status")
@manual_status
def enemy_stats():
    for opponent in enemies:
        print('\033[30m'"--- ",opponent.NAME," ---")
        # TODO: Colour depending on state
        print('\033[32m'"EV = ", opponent.EV,"PR = ",opponent.PR, "Statut = "
```




```
Bertrand
EV = 32 PR = 2
```

Process Action

```
Bertrand : Maladresse Parade!
Maladresse Parade:
Standard: Casse son arme (selon point rupture). Combats mains nues, change
d'arme
Hand_to_Hand: Frappe un allie proche vers la droite. Degats sur allie
Dice 4: 4 Dice 6: 2 Dice 10: 1 Dice 20: 3
```

Everyone Attack!

Check Current Status

In [ ]: