

SPRAV HRU PODĽA ZADANIA

ARKADIUS

PYTHON HRA

**Naprogramuj si vlastnú textovú
adventúrku v Pythone.**

STREET {  }
OF CODE { **I** }

Obsah

Obsah	1
Aku hru vytvoríme?	4
Čo už treba vedieť?	4
Ako programovať podľa zadania?	5
Implementácia	6
1. Začiatok hry	6
Čo treba spraviť	6
Popis	6
Pomôcka	6
2. Vypýťanie a uloženie mena bojovníka	8
Čo treba spraviť	8
Popis	8
Pomôcka	8
3. Definujeme schopnosti bojovníka	10
Čo treba spraviť	10
Popis	10
Pomôcka	11
4. Prvotné rozdelenie schopností	13
Čo treba spraviť	13
Popis	13
Pomôcka	15
5. Refaktor (veľmi dôležité)	16
Čo treba spraviť	16
Popis	16
Pomôcka	18
6. Medzikrok	19
Čo treba spraviť	19
Popis	19
Pomôcka	20
7. Upravenie hrdinu	21
Čo treba spraviť	21
Popis	21
Pomôcka	22
8. Pridanie možnosti opísania jednotlivých schopností	23
Čo treba spraviť	23
Popis	23
Pomôcka	24
9. Uloženie hry (save)	25
Čo treba spraviť	25
Popis	25

Pomôcka	26
10. Načítanie hry (load)	27
Čo treba spraviť	27
Popis	27
Pomôcka	28
11. Vytvorenie nepriateľov	29
Čo treba spraviť	29
Popis	29
12. Príprava na bitku	31
Čo treba spraviť	31
Popis	31
Pomôcka	32
13. Simulácia bitky	34
Čo treba spraviť	34
Popis	34
Pomôcka	36
14. Prechod medzi bitkami (levelmi)	37
Čo treba spraviť	37
Popis	37
Pomôcka	39
15. Vytvorenie predmetov	40
Čo treba spraviť	40
Popis	40
Pomôcka	44
16. Získavanie predmetov	45
Čo treba spraviť	45
Popis	45
Pomôcka	46
Diagram hry	47
Návrhy na zlepšenia	48
Záverečné slová	50

Aku hru vytvoríme?

Podľa tohto zadania budeme vytvárať textovú adventúrku. Bude to python program, ktorý hráč spustí, a podľa textu v konzole sa bude hra rozvíjať. Hráč bude hru ovládať len pomocou inputu, resp. pomocou zadávania čísel, ktoré budú reprezentovať nejakú voľbu.

Na začiatku hry bude hráč vyzvaný, aby vytvoril meno pre svojho bojovníka. Hráč bude s bojovníkom počas hry bojovať proti rôznym príšerám. Bojovník má nasledujúce schopnosti: útočná sila, obrana, obratnosť, skill, život a šťastie. Každá schopnosť je dobrá pre niečo iné.

Ešte pred prvým súbojom hráč dostane 7 bodov schopností, ktoré bude musieť prerozdeliť svojmu bojovníkovi. Následne ide bojovník bojovať proti prvej príšere.

Po každom súboji sa môžu stať dve veci, v závislosti od toho, či bojovník bitku vyhrá alebo prehrá. Ak bojovník bitku prehrá, tak sa mu minie život, ktorý si ale bude môcť doplniť. Po prehraní súboju môže bojovník taktiež stratiť určité nadobudnuté predmety.

Ak bojovník zvíťazí, bude odmenený dvoma vecami. Jednak budú hráčovi ponúknute predmety, ktoré si môže aktivovať, a tiež dostane ďalšie body schopností, ktoré môže prerozdeliť do schopností.

Boje sú jadrom hry. Medzi bitkami si môže hráč bojovníkovi prerozdeliť body schopností, ako aj uložiť či ukončiť hru.

Posledný súper je najsilnejší, a ak aj toho bojovník porazí, tak je koniec hry.

Čo už treba vedieť?

Čo treba vedieť na to, aby si bol/a schopný/á naprogramovať takúto hru? Treba vedieť základy Pythonu. Za základy považujeme:

- premenné a dátové typy
- načítanie vstupu
- podmienené príkazy a logické operátory
- funkcie
- globálne a lokálne premenné
- for cyklus a while cyklus
- vlastné a vstavané moduly
- čítanie a zapisovanie do súboru
- list
- dictionary

A samozrejme treba mať nainštalovaný Python a vedieť spustiť program. Všetky tieto základy sú vysvetlené v kurze [Informatika 101](#). Objektovo orientované programovanie netreba vedieť.

Ak je toto tvoj prvý väčší program, ktorý ideš urobiť, tak sa pravdepodobne dosť potrápiš. Ale neboj, pomôžem ti. Či už textom na nasledujúcich stránkach, alebo si budeš pomáhať videami.

Jeden z najväčších problémov začínajúcich programátorov je štruktúra projektu. Ako si rozložiť funkcionality programu do viacerých zložiek? Do viacerých súborov a do viacerých funkcií? Nechceme mať predsa jeden 3000 riadkový súbor main.py, v ktorom budeme mať všetko. To by bolo neprehľadné. Na toto treba proste skúsenosti. Každým projektom to budeš vedieť o niečo lepšie.

Okrem vecí, ktoré sme sa učili v Informatika 101, resp vecí, ktoré sú popísané vyššie, netreba ozaj nič iné vedieť. Každopádne to nie je úplne jednoduchý projekt a všetky koncepty si budeš musieť znova prekopať zľava aj zprava.

Ako programovať podľa zadania?

Zadanie je rozdelené do 16 krokov. Každý krok je rozdelený na 3 časti: Čo treba spraviť, Popis a Pomôcka.

Čo treba spraviť

V tejto sekcii v skratke popisujem, čo ideme spraviť.

Popis

Tu rozpišem jednotlivé body tak, aby to bolo možné podľa tohto popisu naimplementovať. V tejto časti vkladám aj screenshoty hry alebo snippety kódu.

Pomôcka

Ak ti nestačí popis na to, aby si vedel/a daný krok naprogramovať, tak ti možno pomôže táto pomôcka. Na konci každej pomôcky je link na github commit, resp kód, ktorý som ja naprogramoval pre spravenie daného kroku. Ak sa chceš čo najviac naučiť, tak na ten kód neklikaj, až kým si nebudeš vedieť rady.

Celý kód je na verejnom github [projekte](#) a je rozdelený do 16 commitov, ktoré reprezentujú dané kroky v tomto zadání. Ak si napríklad na 3. kroku a nevieš, ako niečo spraviť, tak vždy si môžeš pozrieť ako som to naprogramoval ja a inšpirovať sa. Pozrel by si commit s názvom "third-step". Každopádne aj keby tvoje riešenie vyzeralo úplne inak ako moje, ale je rovnako funkčné, tak paráda. Je milión spôsobov ako urobiť jednu vec.

Ak máš nejaké otázky tak nám napíš na [Discord](#) alebo na mail info@streetofcode.sk

Implementácia

1. Začiatok hry

Čo treba spraviť

V tomto 1. kroku chceme vypísať text, ktorý sa vypíše na obrazovku, keď sa zapne hra. Hneď po vypísaní textu sa spýtame hráča, či je pripravený hrať. Celé to ale musíme zabaliť do cyklu, v ktorom bude samotná hra bežať.

Popis

Keď hráč zapne hru, vypíše sa mu v konzole nasledovné.

```
Práve si započ hru Arkadius, v ktorej budeš bojovať proti prišerám a pri tom si zlepšovať svojho hrdinu. Si na to pripravený?
0 - Nie, bojím sa.
1 - Áno, poďme na to.
Aká je tvoja voľba?:
```

Hráčovi dáme na výber:

- môže napísať 0, čo bude znamenať, že chce ukončiť hru
- môže napísať 1, čo znamená, že chce pokračovať

Čo sa ale stane, ak napíše hocičo iné? - Budeme mu musieť napísať, že netrafil ani jednu možnú voľbu a teda spýtať sa ho znova to isté, až pokiaľ sa trafi. Vyzerá to bude nejak takto.

```
Práve si započ hru Arkadius, v ktorej budeš bojovať proti prišerám a pri tom si zlepšovať svojho hrdinu. Si na to pripravený?
0 - Nie, bojím sa.
1 - Áno, poďme na to.
Aká je tvoja voľba?: xx
Netrafil si ani jednu možnú voľbu. Musím sa ťa spýtať ešte raz.
Práve si započ hru Arkadius, v ktorej budeš bojovať proti prišerám a pri tom si zlepšovať svojho hrdinu. Si na to pripravený?
0 - Nie, bojím sa.
1 - Áno, poďme na to.
Aká je tvoja voľba?: 10101
Netrafil si ani jednu možnú voľbu. Musím sa ťa spýtať ešte raz.
Práve si započ hru Arkadius, v ktorej budeš bojovať proti prišerám a pri tom si zlepšovať svojho hrdinu. Si na to pripravený?
0 - Nie, bojím sa.
1 - Áno, poďme na to.
Aká je tvoja voľba?: 1
Výborne, máš odvalu. To sa mi páči.
```

Pomôcka

Nižšie je zdrojový kód, ktorý sa dokola pýta používateľa, aby napísal číslo 7, až kým ho konečne nenapíše, kedy mu program zagrataluje a následne sa vypne.

Kód

```
should_continue = True
while should_continue:
    number = input("Napis číslo 7: ")
    if number != "7":
```

```
        continue
    else:
        print("Super, zvladol si to")
        break

print("Koniec")
```

Priebeh pomôcky v konzole

```
Napis cislo 7: xx
Napis cislo 7: 6
Napis cislo 7: 7
Super, zvladol si to
Koniec
```

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#).

2. Vypytanie a uloženie mena bojovníka

Čo treba spraviť

Pokračujeme v hre. Teraz sa budeme pýtať hráča, aby zvolil meno svojho bojovníka. Keď hráč meno zvolí, pre istotu sa ho spýtame, či si je istý daným menom (vypíšeme mu meno). Hráč vtedy musí potvrdiť svoju voľbu, alebo si bude voliť znova iné meno.

Keďže celé toto je v rámci hlavného while cyklu hry, potrebuje vytvoriť ďalší while cyklus pre tento krok, ktorý bude v rámci hlavného while cyklu.

Popis

Ak spravíme všetko, čo treba, tak aj so začiatkom hry môžeme na obrazovke vidieť nasledovné.

```
Práve si započal hru Arkadius, v ktorej budeš bojovať proti prišerám a pri tom si zlepšovať svojho hrdinu. Si na to pripravený?
0 - Nie, bojím sa.
1 - Áno, poďme na to.
Aká je tvoja voľba?: 1
Výborne, máš odvalu. To sa mi páči.
Ako sa bude volať tvoj hrdina?: Superman
Si si istý, že sa tvoj hrdina bude volať Superman?
0 - Nie, chcem zmeniť meno
1 - Áno
Aká je tvoja voľba?: 0
Ako sa bude volať tvoj hrdina?: Spiderman
Si si istý, že sa tvoj hrdina bude volať Spiderman?
0 - Nie, chcem zmeniť meno
1 - Áno
Aká je tvoja voľba?: 1
Ahoj Spiderman
```

Pomôcka

Ako pomôcku máme vonkajší a vnútorný while cyklus.

Kód

```
while True:
    print("Bezi vonkajsi cyklus")
    while True:
        print("Bezi vnutorny cyklus")
        inner_choice = input("stlac 1 pre ukoncenie vnutorneho cyklu: ")
    )
        if inner_choice == "1":
            break

        outer_choice = input("stlac 1 pre ukoncenie vonkajsieho cyklu: ")
        if outer_choice == "1":
            break

print("Ukoncene cykly")
```


Priebeh pomôcky v konzole

```
Bezi vonkajsi cyklus  
Bezi vnutorny cyklus  
stlac 1 pre ukoncenie vnutorneho cyklu: 1  
stlac 1 pre ukoncenie vonkajsieho cyklu: 1  
Ukoncene cykly
```

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#).

3. Definujeme schopnosti bojovníka

Čo treba spraviť

V tomto kroku si zadefinujeme dictionary (slovník), v ktorom budeme mať jednotlivé schopnosti bojovníka, ako aj ich body a popis.

Popis

Skopíruj si do tvojho súboru (pravdepodobne *main.py*) nasledujúci kód, ktorý predstavuje definíciu schopností.

```
abilities = {
    "Útočná sila": {
        "points": 1,
        "description": "Sila je potrebna k útoku, do ktorého okrem
sily vstupuje aj obratnosť a skill."
    },
    "Obrana": {
        "points": 1,
        "description": "Celková obrana sa ráta z bodov obrany +
obratnosti."
    },
    "Obratnosť": {
        "points": 1,
        "description": "Obratnosť je dôležitá aj pre obranu aj pre
útok."
    },
    "Skill": {
        "points": 1,
        "description": "Skill je dôležitý pri normálnom útoku ako aj
kritickom útoku"
    },
    "Život": {
        "points": 50,
        "description": "Život je dôležitý pri bitke. Život sa dá
doplniť po každom súboji."
    },
    "Šťastie": {
        "points": 1,
        "description": "Šťastie je dôležité pre kritický útok"
    }
}
```

V tejto hre budeme mať nasledujúce schopnosti: útočná sila, obrana, obratnosť, skill, život a šťastie.

Inšpiroval som sa online hrou Gladius. Neskôr uvidíme, na čo sú jednotlivé schopnosti dobré, hoci už teraz je to popísané v *description*.

abilities je dátová štruktúra slovník (dictionary). Každý slovník môže mať viacero itemov. Jeden item sa skladá z kľúča a hodnoty. Pre *abilities* je kľúčom vždy názov schopnosti, a každá schopnosť má ako hodnotu tiež slovník, v ktorom je *points* a *description*. V podstate si pre každú schopnosť vytvárame objekt, ktorý má *points* a *description*.

Pomôcka

Na pomôcke uvidíme, ako funguje takýto komplexnejší slovník, ktorý má ako hodnoty ďalšie slovníky.

Kód

```
outer_dictionary = {
    "inner_key_1": {
        "points": 1
    },
    "inner_key_2": {
        "points": 2
    }
}
print(outer_dictionary)

print("-----")

inner_key_1 = outer_dictionary["inner_key_1"]
print(inner_key_1)
print(inner_key_1["points"])

print("-----")

inner_key_2 = outer_dictionary["inner_key_2"]
print(inner_key_2)
print(inner_key_2["points"])
```

Priebeh pomôcky v konzole

```
{'inner_key_1': {'points': 1}, 'inner_key_2': {'points': 2}}
-----
{'points': 1}
1
-----
{'points': 2}
2
```

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#).

4. Prvotné rozdelenie schopností

Čo treba spraviť

Teraz nasleduje prvá ťažšia úloha. Na začiatku hry bude mať bojovník 7 bodov, ktoré si môže prerozdeliť naprieč schopnosťami. Čiže si vlastne 7 krát vyberie, ktorú schopnosť si chce zlepšiť. Vždy keď si nejakú schopnosť vyberie, tak mu ju treba aj reálne zlepšiť v slovníku *abilities* a to tak, že danej schopnosti zväčšíme *points* o 1 (V prípade života zväčšujeme život o 5 pre každý bod).

Popis

Budeme mať teda ďalší vnútorný while cyklus, v ktorom povieme bojovníkovi, koľko bodov môže rozdeliť a následne mu ponúkneme možnosti. Potom budeme počúvať jeho input a následne ak napísal správne nejakú možnosť, tak tej možnosti pridáme bod a znova zopakujeme tento proces, až pokým nevyčerpá všetkých 7 počiatočných bodov. Nezabudni, že vždy treba ošetriť hráčov vstup - čiže ak sa netrafí medzii možnosti, tak to treba zopakovať.

Ako vyzerá tento krok v konzole (obrázok začína vypýtaním mena):

```

Ako sa bude volať tvoj hrdina?: Spiderman
Si si istý, že sa tvoj hrdina bude volať Spiderman?
0 - Nie, chcem zmeniť meno
1 - Áno
Aká je tvoja voľba?: 1
Ahoj Spiderman
Spiderman, Tvoje schopnosti sú momentálne na tom takto:
Útočná sila - 1, Obrana - 1, Obratnosť - 1, Skill - 1, Život - 50, Šťastie - 1,

Máš 7 bodov, ktoré si rozdeľ naprieč schopnostiam podľa svojich preferencií.
1 - Útočná sila
2 - Obrana
3 - Obratnosť
4 - Skill
5 - Život - jeden bod pridá 5 života
6 - Šťastie
Máš 7 možností na zlepšenie. Ktorú schopnosť chceš vylepšiť?: 1

Vybral si si schopnosť - Útočná sila. Pridávam ti bod.
Teraz tvoje body vyzerajú takto:
Útočná sila - 2, Obrana - 1, Obratnosť - 1, Skill - 1, Život - 50, Šťastie - 1,

1 - Útočná sila
2 - Obrana
3 - Obratnosť
4 - Skill
5 - Život - jeden bod pridá 5 života
6 - Šťastie
Máš 6 možností na zlepšenie. Ktorú schopnosť chceš vylepšiť?: 77
Netrafil si výber, musím sa ťa spýtať ešte raz.

1 - Útočná sila
2 - Obrana
3 - Obratnosť
4 - Skill
5 - Život - jeden bod pridá 5 života
6 - Šťastie
Máš 6 možností na zlepšenie. Ktorú schopnosť chceš vylepšiť?: |

```

Pozorne si preštuduj obrázok. Všimnime si, že pred každým výberom schopnosti, hráčovi tieto schopnosti vypíšeme v ich aktuálnej podobe. V tomto obrázku teda sme najprv vylepšili útočnú silu. Potom nám konzola vypísala, že sme si vybrali útočnú silu, a že nám dáva bod. Následne boli schopnosti vypísané na obrazovku. Potom sme v ďalšom kole napísali 77 a teda netrafili sme sa do žiadnej možnosti, čo nám aj konzola vypísala, a teda zopakovala otázku.

Keď hráč určí bojovníkovi poslednú, 7. schopnosť, tak zrekapitulujeme schopnosti a následne napíšeme, či je hráč pripravený na prvý súboj.

```
Výborne Ferko. Dokončil si pridávanie schopnosti. Pre rekapituláciu, teraz vyzerajú tvoje schopnosti takto.
Útočná sila - 2, Obrana - 2, Obratnosť - 2, Skill - 3, Život - 55, Šťastie - 2,

Si pripravený na prvý súboj?
```

Pomôcka

Na vypísanie aktuálnych schopností a tiež na vypísanie možností na zlepšenie si vytvor funkcie `print_abilities_points()` a `print_abilities_options()`

Keďže sme ešte na relatívnom začiatku hry, pomôžem ti s týmito funkciami.

```
def print_abilities_options():
    for i, ability in enumerate(abilities.keys()):
        ability_option = str(i + 1) + ' - ' + ability
        if ability == "Život":
            ability_option += " " + "- jeden bod pridá 5 života"
        print(ability_option)

def print_abilities_points():
    for k, v in abilities.items():
        print(k + " - " + str(v["points"]), end=", ")
    print("\n")
```

Ak si nevieš spomenúť, ako sa pracuje s dictionary, tak si to buď vygoogli alebo pozri [video](#) z kurzu Informatika 101

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#).

5. Refaktor (veľmi dôležité)

Čo treba spraviť

Ak sa v tejto chvíli pozrieš na kód, tak to nie je najkrajší pohľad. Určite nemôžeme mať jeden súbor, ktorý bude mať niekoľko tisíc riadkov a bude úplne nečitateľný. Síce teraz nemá tak veľa, ale smeruje to k tomu. Treba refaktorovať. A to znamená vytvorenie viacerých súborov a rozdelenie kódu na viacero menších častí.

Popis

Táto časť je asi najviac subjektívna zo všetkých. Keby to robilo 1000 programátorov, tak bude 1000 rôznych verzií. Mali by mať však nejaké spoločné črty.

Ako to teda spravíme my? Navrhujem, aby sme si hru rozdelili na tzv. fázy. V hlavnom cykle hry budeme ifovať premennú *current_phase*. Hru si rozdelíme zatiaľ na fázy: INTRO, END, NAME, INTRO_ABILITIES. Môžeme si vytvoriť nový súbor *phase_constants.py*, v ktorom budeme mať tieto konštanty.

```
INTRO = "INTRO"
END = "END"
NAME = "NAME"
INTRO_ABILITIES = "INTRO_ABILITIES"
```

Taktiež si vytvorme nasledujúce súbory: *abilities.py*, *intro.py*, *name.py*. Tieto súbory aj s *phase_constants.py* si uložíme do priečinka *phase*.

Tieto súbory budú reprezentovať jednotlivé fázy, ktoré zatiaľ máme.

V *intro.py* môžeme mať funkciu *intro_phase()* s úvodným textom a výzvou, či je hráč pripravený na hru.

V *name.py* budeme mať funkciu *name_phase()* s pýtaním sa mena.

V *abilities.py* si môžeme vytvoriť funkciu *abilities_update()*, ktorá bude riešiť počiatočné pridávanie schopností.

Funkcie *intro_phase()*, *name_phase()* a *abilities_update()* budú vracat' string, ktorý bude reprezentovať ďalšiu fázu, ktorá nasleduje.

Okrem toho si vytvorme ešte súbor *utility.py* (do ktorého dáme naše print funkcie), vytvoríme si súbor *hero_data.py*, ktorý bude obsahovať dáta bojovníka (v tejto chvíli sú to *abilities* a *hero_name*). Taktiež si vytvorme súbor *game_constants.py*, v ktorom budeme mať nateraz len pomocnú konštantu.

```
DIVIDER = "-----"
```


Súbor *main.py* bude vyzerat' nejak takto:

```
import game_constants
import phase.phase_constants as phase_constants
from phase.abilities import abilities_update
from phase.intro import intro_phase
from phase.name import name_phase

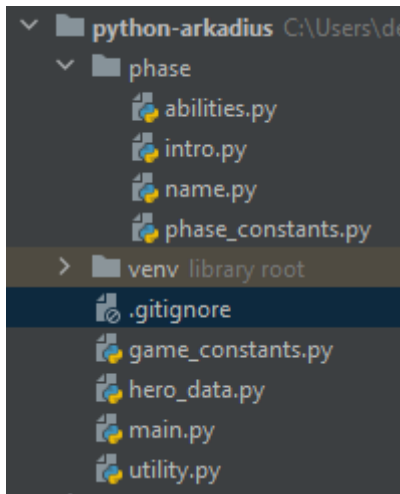
current_phase = phase_constants.INTRO

continue_game = True
while continue_game:

    if current_phase == phase_constants.INTRO:
        current_phase = intro_phase()
    elif current_phase == phase_constants.END:
        print(game_constants.DIVIDER)
        print("Dovidenia")
        continue_game = False
    elif current_phase == phase_constants.NAME:
        print(game_constants.DIVIDER)
        current_phase = name_phase()
    elif current_phase == phase_constants.INTRO_ABILITIES:
        print(game_constants.DIVIDER)
        abilities_update()
        print(game_constants.DIVIDER)
        print("Si pripravený na prvý súboj?")
        break
```

Pomôcka

Ako vyzerá štruktúra projektu po refaktore? (venv a .gitignore si nevšímaj)



Keď si chceš vytvoriť viaceré súbory, tak si budeš robiť de facto vlastné moduly. Ako sa to robí si môžeš pozrieť v tomto [videu](#).

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commit](#)

6. Medzikrok

Čo treba spraviť

Momentálne keď hráč spustí hru, tak si najprv volí meno, potom počiatkové schopnosti a následne sa už bude bojovať. Ale chýba tam niečo medzi nemyslíš?

Chceme dať "ovládač" do ruky hráčovi, takže mu vypíšeme menu, podľa ktorého zvolí, čo chce ďalej robiť.

Popis

Hráčovi po priradení počiatkových schopností dáme nasledujúce možnosti:

- pokračovať na súboj (po zvolení sa prepne fáza hry na FIGHT)
- upraviť hrdinu (keď zvolí túto možnosť, tak bude môcť pridať alebo odstrániť body hrdinovi)
- uložiť hru (hra sa uloží do súboru, hráč ale pokračuje v hre)
- ukončiť hru (hra sa ešte hráča spýta, či chce naozaj hru ukončiť)

V tejto časti budeme implementovať v tomto menu pokračovanie súboju a ukončenie hry. Pre upravenie hrdiny a uloženie hry si napíšeme TODO komentár funkčnosť budeme programovať v nasledujúcich krokoch..

Menu v hre by malo vyzeráť nejak takto.

```
Výborne Spiderman. Dokončil si pridávanie schopnosti. Pre rekapituláciu, teraz vyzerajú tvoje schopnosti takto.
Útočná sila - 4, Obrana - 4, Obratnosť - 1, Skill - 1, Život - 55, Šťastie - 1,

-----
0 - Pokračovať na FIGHT
1 - Upraviť hrdínu
2 - Uložiť hru
3 - Ukončiť hru
Aký je tvoj ďalší krok? |
```

Ako to môžeme spraviť? Vytvoríme si nový súbor *check.py* a môžeme ho vložiť do priečinka *phase*. V tomto súbore si spravíme funkciu *phase_check(next_phase)*, ktorá má parameter názov fázy, ktorá má pokračovať. V tejto funkcii budeme mať znova while cyklus v ktorom budeme vypisovať tieto možnosti. Ak hráč zvolí možnosť 0, tak jednoducho vrátime *next_phase*. Ak zvolí 1 alebo 2, tak dáme do podmienky *continue*. Ak hráč zvolí možnosť 3, tak implementujeme overovaciu časť. Pozrieme sa, ako to vyzerá v konzole.

```

-----
0 - Pokračovať na FIGHT
1 - Upraviť hrdinu
2 - Uložiť hru
3 - Ukončiť hru
Aký je tvoj ďalší krok? 3
Netrafil si ani jednu možnú voľbu. Musím sa ťa spýtať ešte raz.
0 - Pokračovať na FIGHT
1 - Upraviť hrdinu
2 - Uložiť hru
3 - Ukončiť hru
Aký je tvoj ďalší krok? 3
Si si istý že chceš ukončiť hru?
0 - Späť
1 - Áno, ukončiť.
Naozaj ukončiť? 0

0 - Pokračovať na FIGHT
1 - Upraviť hrdinu
2 - Uložiť hru
3 - Ukončiť hru
Aký je tvoj ďalší krok? 3
Si si istý že chceš ukončiť hru?
0 - Späť
1 - Áno, ukončiť.
Naozaj ukončiť? 1
-----
Dovidenia

```

Pomôcka

Funkciu `check_phase` použi v `main.py` nasledovne.

```

elif current_phase == phase_constants.INTRO_ABILITIES:
    print(game_constants.DIVIDER)
    abilities_update(game_constants.INTRO_ABILITIES_COUNT)
    print(game_constants.DIVIDER)
    current_phase = phase_check(phase_constants.FIGHT)
elif current_phase == phase_constants.FIGHT:
    print("Si pripravený na prvý súboj?")
    break

```

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#).

7. Upravenie hrdinu

Čo treba spraviť

Ideme implementovať časť upravenia hrdiny, resp. čo sa stane, keď po minulom kroku spustíme v menu možnosť 1 - upravenie hrdiny.

V skratke ide o to, že hráč bude môcť pridať schopnosť alebo odstrániť schopnosť hrdinovy. Budeme ale potrebovať pridať do *hero_data* premennú *available_points*, ktorá bude vyjadrovať, koľko bodov môže hráč hrdinovi pridať.

Vždy keď hrdinovi pridáme schopnosť, tak sa toto číslo zmenší o jedna a naopak, ak hrdinovi odoberieme schopnosť, tak sa *available_points* zväčší o jedna. Samozrejme sa hrdinovi nedá pridať žiadny bod, ak má *available_points* 0, a zároveň nemôže hráč odstrániť schopnosť, ktorej hodnota je už 0.

Popis

Ukážeme si rovno na obrázku, ako sa má táto funkcionálna správať. Nižšie máme dva obrázky vedľa seba. Je to rozdelené, aby to nebol jeden veľký obrázok. Čiže kde ľavý obrázok končí, tam pravý začína.

```

-----
0 - Pokračovať na FIGHT
1 - Upraviť hrdinu
2 - Uložiť hru
3 - Ukončiť hru
Aký je tvoj ďalší krok? |

Ferko, Tvoje schopnosti sú momentálne na tom takto:
Útočná sila - 3, Obrana - 3, Obratnosť - 3, Skill - 2, Život - 50, Šťastie - 1,

Máš 0 bodov na pridelenie schopností.

0 - Späť
1 - Upraviť schopnosti hrdiny
Čo chceš robiť? |
0 - Späť
1 - Pridať body (máš 0 bodov na pridanie)
2 - Odstrániť body zo schopnosti
Aká je tvoja voľba? |
Nemáš žiadne body, ktoré by si mohol pridať.
0 - Späť
1 - Pridať body (máš 0 bodov na pridanie)
2 - Odstrániť body zo schopnosti
Aká je tvoja voľba? |
0 - Už nechcem odoberať body. (Späť)
1 - Útočná sila
2 - Obrana
3 - Obratnosť
4 - Skill
5 - Život - jeden bod pridá 5 života
6 - Šťastie
Ktorej schopnosti chceš odoberať bod? |

Vybral si si schopnosť - Útočná sila. Odoberám ti bod.
Teraz tvoje body vyzerajú takto:
Útočná sila - 2, Obrana - 3, Obratnosť - 3, Skill - 2, Život - 50, Šťastie - 1,

0 - Už nechcem odoberať body. (Späť)
1 - Útočná sila
2 - Obrana
3 - Obratnosť
4 - Skill
5 - Život - jeden bod pridá 5 života
6 - Šťastie
Ktorej schopnosti chceš pridať bod? |

Vybral si si schopnosť - Skill. Pridávam ti bod.
Teraz tvoje body vyzerajú takto:
Útočná sila - 2, Obrana - 3, Obratnosť - 3, Skill - 3, Život - 50, Šťastie - 1,

Už nemáš žiadne body, ktoré by si mohol pridať.
0 - Späť
1 - Pridať body (máš 0 bodov na pridanie)
2 - Odstrániť body zo schopnosti
Aká je tvoja voľba? |
0 - Späť
1 - Upraviť schopnosti hrdiny
Čo chceš robiť? |
0 - Pokračovať na FIGHT
1 - Upraviť hrdinu
2 - Uložiť hru
3 - Ukončiť hru
Aký je tvoj ďalší krok? |

```

Určite si to rozdeľ do funkcií. Ja osobne som si vytvoril funkciu *hero_check()*, ktorú som umiestnil do súboru *check.py*. Taktiež som vytvoril súbor *hero_update.py*, v ktorých mám funkcie *hero_update()*, *hero_add_point()* a *hero_substract_point()*

Ako som písal vyššie, treba si vytvoriť novú premennú, ktorá bude reprezentovať voľné body schopností.

```
available_points = 0
```

Pomôcka

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#).

8. Pridanie možnosti opísania jednotlivých schopností

Čo treba spraviť

Nám je možno jasné, načo sú dané schopnosti dobré, ale hráč nevidí náš dictionary *abilities* a ich description. Takže chceli by sme, aby pri pridávaní schopností sme dali hráčovi nultú možnosť, ktorá po spustení vypíše jednotlivým schopnostiam ich description.

Popis

Pridaj 0. možnosť v menu pridávania schopnosti, ktorá po zvolení vypíše schopnostiam ich description (popis). Nech to vyzerá nejak takto.

```
Máš 7 bodov, ktore si rozdel naprieč schopnostiam podľa svojich preferencií.
0 - Vysvetlivky - načo su dobré jednotlivé schopnosti?
1 - Útočná sila
2 - Obrana
3 - Obratnosť
4 - Skill
5 - Život - jeden bod pridá 5 života
6 - Šťastie
Máš 7 možnosti na zlepšenie. Ktorú schopnosť chceš vylepšiť? 0
-----
Útočná sila - Sila je potrebna k útoku, do ktorého okrem sily vstupuje aj obratnosť a skill.
Obrana - Celkový obrana sa ráta z bodov obrany + obratnosti.
Obratnosť - Obratnosť je dôležitá aj pre obranu aj pre útok.
Skill - Skill je dôležitý pri normálnom útoku ako aj kritickom útoku
Život - Život je dôležitý pri bitke. Život sa dá doplniť po každom súboji.
Šťastie - Šťastie je dôležité pre kritický útok
-----
0 - Vysvetlivky - načo su dobré jednotlivé schopnosti?
1 - Útočná sila
2 - Obrana
3 - Obratnosť
4 - Skill
5 - Život - jeden bod pridá 5 života
6 - Šťastie
Máš 7 možnosti na zlepšenie. Ktorú schopnosť chceš vylepšiť? |
```

Pomôcka

Už máš funkciu, ktorá vypisuje tie možnosti, ale treba ju rozšíriť. U mňa sa táto funkcia volá `print_abilities_options()`, a rozšírim ju tak, že jej pridam parameter `with_help_option`, ktorú defaultne nastavím na `False`. Čiže definícia funkcie vyzerá takto

```
def print_abilities_options(with_help_option=False):
```

Vo funkcii potom mám len jeden `if`, kde čeknem či je `with_help_option` `True`, a ak áno, tak tam hodím jeden `printík`.

Aby to celé fungovalo, treba aj mierne upraviť metódu `abilities_update`, aby počítala s 0 možnosťou.

Defaultný parameter som v kurze Informatika 101 nevysvetloval. Pre pochopenie si prečítaj tento [článok](#).

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#).

9. Uloženie hry (save)

Čo treba spraviť

Uloženie hry - save game. Na prvý pohľad to môže byť odstrašujúce, ale fakt je to veľmi jednoduché. Chceme uložiť hru, to znamená, že chceme uložiť tie informácie, ktoré súvisia s bojovníkom a progresom hry. Čiže všetko, čo máme v *hero_data.py* (*hero_name*, *available_points*, *abilities*).

Okrem toho však ešte treba uložiť názov aktuálnej fázy, aby sme po načítaní vedeli, kde sa v hre nachádzame (aby to šlo do správneho ifíku vo while cykle).

Popis

Veľmi dôležité je poradie, akým zapíšeme dáta. Pretože v tom istom poradí ten súbor budeme neskôr načítavať. Ja som zvolil poradie nasledujúce.

V prvom riadku zapíšeme meno bojovníka. V ďalších riadkoch napíšeme schopnosti tak, že pre každú schopnosť vytvoríme nový riadok, v ktorom bude "názov schopnosti - počet bodov". V poslednom riadku napíšeme v akej fáze hry sa aktuálne nachádzame.

Menu po zvolení uloženia hry bude vyzerat' nejak takto.

```
0 - Pokračovať na FIGHT
1 - Upraviť hrdinu
2 - Uložiť hru
3 - Ukončiť hru
Aký je tvoj ďalší krok? 2
Pod akým názvom chceš uložiť hru? (Názov nesmie obsahovať čísla, špeciálne znaky ani medzery)
Názov - ferko
Úspešne som uložil hru
```

V názvoch súborov nemôžu byť špeciálne znaky, čísla či medzery, čiže to treba ošetriť. Uložené hry ukladáme do .txt súborov pod novo vytvorený priečinok *saved*. Samozrejme si priečinok nazvi ako chceš, ale určite si priečinok vytvor, pretože budeš mať neskôr problém, ak to tak nespravíš.

Obsah uloženého súboru vyzerá takto.

```
Ferko
Útočná sila - 5
Obrana - 3
Obratnosť - 1
Skill - 2
Život - 50
Šťastie - 1
FIGHT
```

Funkcionalitu ukladania hry naprogramujeme v novom súbore *save_load.py* vo funkcii *save_game(phase)*.

Pomôcka

Ak si nepamätáš, ako sa píše do súboru, tak pozri toto [videjko](#) z kurzu Informatika 101.

Kvôli diakritike budeš musieť špecifikovať encoding. To čo chceme je encoding [utf-8](#).

```
file_handler = open(file_path, "w", encoding="utf-8")
```

Čo sa týka validácie názva súboru, resp aby neobsahoval špeciálne znaky, čísla či medzery - môžeš jednoducho použiť vstavanú funkciu nad stringom *isalpha()*, ktorá vráti True, ak všetky znaky v texte sú alfabetické (sú to len písmená z abecedy).

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#)

10. Načítanie hry (load)

Čo treba spraviť

Potrebuje na začiatku hry dať možnosť hráčovi načítať hru. Program ponúkne možnosti uložených hier, ktoré má v priečinku *saved*. Po zvolení súboru sa hra načíta a bude pokračovať tam, kde ju hráč uložil s dátami, ktoré boli vtedy aktuálne.

Popis

Doteraz sme mali ako prvý krok po zapnutí hry INTRO fázu, v ktorej sme sa pýtali hráča, či je pripravený hrať hru. Teraz spravíme nultú fázu, ktorá sa bude vyhodnocovať ešte skôr, ako INTRO fáza. Túto fázu nazveme START.

Správanie načítania hry z pohľadu hráča hneď po zapnutí hry vidíme v nasledujúcom obrázku.

```
0 - Začať novú hru
1 - Načítať uloženú hru
Aká je tvoja voľba?: 1
-----
0 - Späť
1 - ferko
Ktorú hru chceš načítať? 1
Načítavam hru uloženú pod názvom - ferko

Hra sa načítala...Vitaj späť Ferko
0 - Pokračovať na FIGHT
1 - Upraviť hrdinu
2 - Uložiť hru
3 - Ukončiť hru
Aký je tvoj ďalší krok? |
```

Logiku načítania hry spravíme v novej funkcii *load_game()*, ktorú prídáme do existujúceho súboru *save_load.py*.

Vo funkcii najprv preskenujeme všetky súbory v priečinku *saved* (ak uložené súbory neukladáš do zvlášť priečinku, tak máš problém), a tieto súbory ponúkneme hráčovi na načítanie.

Pri čítaní súboru treba dávať pozor na to, aby sme čítali informácie z riadkov po sebe tak, ako sme ich aj zapísali. Čiže ak to máš tak ako ja, tak najprv budeme pozerieť na meno hrdiny, potom na jednotlivé schopnosti, potom na názov ďalšej fázy a napokon na voľné body.

Pomôcka

Takto naskenuješ súbory, ktoré sú pod priečinkom *saved*.

```
from os import listdir
from os.path import isfile, join

saved_files = []
for file in listdir("saved"):
    if isfile(join("saved", file)):
        saved_files.append(file)
```

Daj si pozor, aby si otvoril/a súbor pomocou utf-8 encodingu. Taktiež si daj pozor, aby si načítané čísla (napr body schopnosti a voľné body) prekonvertoval/a na integer.

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#)

11. Vytvorenie nepriateľov

Čo treba spraviť

V tomto kroku vytvoríme nový súbor *enemy_data.py* a vložíme doň nepriateľov.

Popis

V tejto hre budeme mať 5 súbojov. Dokopy teda budeme mať 5 nepriateľov, pre každý level jedného. Vytvoríme si nový súbor *enemy_data.py* v hlavnej zložke a vložíme doň tento dictionary enemies.

```
enemies = {
    1: {
        "name": "Harpya (LVL 1)",
        "Útočná sila": 3,
        "Obrana": 1,
        "Obratnosť": 2,
        "Skill": 2,
        "Šťastie": 1,
        "Život": 25,
    },
    2: {
        "name": "Golem (LVL 2)",
        "Útočná sila": 4,
        "Obrana": 3,
        "Obratnosť": 1,
        "Skill": 2,
        "Šťastie": 1,
        "Život": 35,
    },
    3: {
        "name": "Mocny demon (LVL 3)",
        "Útočná sila": 4,
        "Obrana": 4,
        "Obratnosť": 2,
        "Skill": 3,
        "Šťastie": 5,
        "Život": 50,
    },
    4: {
        "name": "Meduza (LVL 4)",
        "Útočná sila": 7,
        "Obrana": 7,
        "Obratnosť": 5,
```

```
    "Skill": 5,  
    "Šťastie": 1,  
    "Život": 40,  
  },  
  5: {  
    "name": "KRAKEN - Boh temnoty",  
    "Útočná sila": 7,  
    "Obrana": 7,  
    "Obratnosť": 7,  
    "Skill": 7,  
    "Šťastie": 7,  
    "Život": 80,  
  }  
}
```

Nepriateľov rozdeľujeme podľa kľúčov, ktoré vyjadrujú level nepriateľa. Každý kľúč ma ako hodnotu objekt, resp slovník, ktorý má viacero atribútov. V podstate ale prvý atribút je názov nepriateľa a potom všetky ďalšie sú jeho schopnosti.

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#).

12. Príprava na bitku

Čo treba spraviť

Čaká nás kľúčová funkcionálnosť. Bitka. V tomto kroku síce ešte neurobíme priamo bojovanie, resp simuláciu bitky, ale nachystáme všetko tak, aby sme to mohli v ďalšom kroku urobiť.

Pripravíme všetky dáta potrebné na bitku. Spravíme reprezentáciu bojovníka a nepriateľa. Vypočítame útoky, obranu a šancu na kritický útok.

Popis

V tomto kroku spravíme nasledovné:

- nový súbor *battle.py* v novom priečinku *fight*
- vytvorenie novej premennej *fight_level* do súboru *hero_data.py*. Táto premenná bude reprezentovať level nepriateľa, proti ktorému bojujeme. Tento level sa bude po každej vyhrajenej bitke zväčšovať, až pokým nepríde posledný boss.
- v súbore *battle.py* budeme mať funkcionálnosť bitky.

Obranu aj útok budeme mať ako interval (minimum, maximum). Chceme potom v bitke v každom kole vypočítať útok, ktorý bude random číslo v tomto intervale. Zároveň aj obranu.

Útok

- minimum = útočná sila
- maximum = útočná sila + obratnosť + skill

Obrana

- minimum = obrana
- maximum = obrana + obratnosť

Šanca na kritický útok v percentách = $(\text{skill} * \text{šťastie}) / 2$

Čo presne znamená kritický útok sa dozvieš v ďalšej kapitole.

Prečo práve takéto výpočty? Skúšal som viaceré možnosti a nebolo to dobre vybalansované, až kým som neskúsil túto možnosť. Určite sa to dá aj lepšie, ale funguje to celkom dobre a zároveň pri bitke je aj faktor náhody, čo je v hrách dôležité.

Čiže v *battle.py* si môžeme vytvoriť funkciu *batte(fight_level)*, v ktorej spravíme reprezentáciu bojovníka a nepriateľa. Potom vypíšeme tieto dáta a ďalej už budeme bojovať, ale to nie ešte v tomto kroku, takže keď vypíšeme dáta, tak môžeme ukončiť hru. Takto by to malo vyzeráť.

```

-----
0 - Pokračovať na FIGHT
1 - Upraviť hrdinu
2 - Uložiť hru
3 - Ukončiť hru
Aký je tvoj ďalší krok? 0
-----

Tvoj hrdina ide do súboja pripravený nasledovne:
Útok: minimum - 3, maximum - 7
Šanca na kritický útok - 1%
Obrana: minimum - 3, maximum - 5
Život - 55

Oproti nemu stojí prišera Harpya (LVL 1) s nasledovnými schopnosťami:
Útok: minimum - 3, maximum - 7
Šanca na kritický útok - 1%
Obrana: minimum - 1, maximum - 3
Život - 25
-----

ZAČÍNAME SÚBOJ. AKO PRVÝ ÚTOČIŠ TY.

Process finished with exit code 0

```

Pomôcka

Moja funkcia `battle(fight_level)` vyzerá nasledovne.

```

def battle(fight_level):
    enemy_data = enemies[fight_level]

    hero = {
        "attack": calculate_hero_attack(),
        "critical_hit": min(100, (abilities["Skill"]["points"] *
abilities["Šťastie"]["points"]) // 2),
        "defence": (abilities["Obrana"]["points"],
abilities["Obrana"]["points"] + abilities["Obratnosť"]["points"]),
        "health": abilities["Život"]["points"]
    }

    enemy = {

```



```
    "name": enemy_data["name"],
    "attack": calculate_enemy_attack(enemy_data),
    "critical_hit": min(100, (enemy_data["Skill"] *
enemy_data["Šťastie"]) // 2),
    "defence": (enemy_data["Obrana"], enemy_data["Obrana"] +
enemy_data["Obratnosť"]),
    "health": enemy_data["Život"]
}

return simulate_battle(hero, enemy)
```

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#)

13. Simulácia bitky

Čo treba spraviť

V minulom kroku som v súbore *battle.py* mal aj funkciu *simulate_battle(hero, enemy)*, ktorá ma na vstupe dáta bojovníka a dáta nepriateľa. Teraz budeme s týmito dátami pracovať.

Vytvoríme simuláciu bitky, budeme vypočítavať útoky, obranu, zisťovať či je kritický útok. Bitka končí vtedy, keď sa niekomu minie život.

Po bitke napíšeme, kto vyhral, a koľko života prípadne zostalo bojovníkovi. Čiže aby som to zhrnul, ideme spraviť simuláciu bitky a začneme riešiť aj to, čo sa stane, keď bojovník vyhrá alebo prehrá.

Popis

Ukážme si kúsok z bitky.

```
-----
Tvoj hrdina ide do súboja pripravený nasledovne:
Útok: minimum - 4, maximum - 8
Šanca na kritický útok - 1%
Obrana: minimum - 3, maximum - 5
Život - 50

Oproti nemu stojí prišera Harpya (LVL 1) s nasledovnými schopnosťami:
Útok: minimum - 3, maximum - 7
Šanca na kritický útok - 1%
Obrana: minimum - 1, maximum - 3
Život - 25
-----

ZAČÍNAME SÚBOJ. AKO PRVÝ ÚTOČÍŠ TY.

Zaútočil si útočnou silou - 7
Harpya (LVL 1) po tvojom útoku stále žije. Súperov zvyšok života - 18

Súper zaútočil útočnou silou - 1
Stále žiješ, zostáva ti 49 Života

Zaútočil si útočnou silou - 5
Harpya (LVL 1) po tvojom útoku stále žije. Súperov zvyšok života - 13

Súper zaútočil útočnou silou - 4
Stále žiješ, zostáva ti 45 Života

Zaútočil si útočnou silou - 4
Harpya (LVL 1) po tvojom útoku stále žije. Súperov zvyšok života - 9

Súperov útok ťa netrafil.

Zaútočil si útočnou silou - 2
Harpya (LVL 1) po tvojom útoku stále žije. Súperov zvyšok života - 7

Súper zaútočil útočnou silou - 3
Stále žiješ, zostáva ti 42 Života
```

V tejto základnej verzii hry to spravíme tak, že vždy bude ako prvý útočiť bojovník. Potom nepriateľ, následne znova bojovník a tak ďalej, až kým niekto nezvíťazí.

Funkcia *simulate_battle* vracia tuple dvoch hodnôt (Boolean, Integer). Prvá hodnota hovorí, či bojovník súboj vyhral, a druhá hodnota reprezentuje, koľko života ostalo bojovníkovi. Pomocou týchto návratových hodnôt budeme vedieť ďalej fungovať v programe - v hlavnom while cykle, vo fáze bitky.

Ak bojovník vyhral, vypíšeme koľko mu ostalo života a okrem toho spravíme ešte dve ďalšie veci (nateraz). Zvýšime *fight_level* premennú o 1, aby v ďalšom kole sme už bojovali proti inej príšere, a tiež dáme bojovníkovi voľný bod (konkrétne toľko voľných bodov, aký bol *fight_level*) a teda teda spustíme zlepšovanie schopnosti.

Ak bojovník prehral, tak nateraz len napíšeme, že si potrebuje oddýchnuť, lebo má 0 života. A teda samozrejme bojovníkovi v každom prípade aktualizujeme bojovníkov život.

Ukážeme si ešte obrázok, ako to vyzerá po vyhrajenej bitke.

```
Zaútočil si útočnou silou - 3
Harpya (LVL 1) po tvojom útoku stále žije. Súperov zvyšok života - 9

Súperov útok ťa netrafil.

Zaútočil si útočnou silou - 4
Harpya (LVL 1) po tvojom útoku stále žije. Súperov zvyšok života - 5

Súper zaútočil útočnou silou - 3
Stále žiješ, zostáva ti 39 Života

Zaútočil si útočnou silou - 5
ZVÍŤAZIL SI!

-----
Po víťaznej bitke ti ostal život 39/50
-----
Po tvojej 1. výhre, ti pridávam 1 body, ktoré môžeš využiť na upravenie tvojho hrdinu.
Ferko, Tvoje schopnosti sú momentálne na tom takto:
Útočná sila - 5, Obrana - 3, Obratnosť - 1, Skill - 2, Život - 50, Šťastie - 1,

Máš 1 bodov, ktore si rozdeľ naprieč schopnostiam podľa svojich preferencií.
0 - Vysvetlivky - načo su dobré jednotlivé schopnosti?
1 - Útočná sila
2 - Obrana
3 - Obratnosť
4 - Skill
5 - Život - jeden bod pridá 5 života
6 - Šťastie

Máš 1 možnosti na zlepšenie. Ktorú schopnosť chceš vylepšiť? |
```

Pri každom útoku, či už od bojovníka alebo nepriateľa treba zistiť, či nastal kritický útok.

Kritický útok sme si rátali ako číslo, ktoré vyjadruje percentá, čiže teraz musíme spraviť funkciu, ktorá zistí (s použitím *random* knižnice), či je útok kritický. Ak áno, tak vynásobíme útok trojnásobne.

Keď sme zistili útok (či už kritický alebo nie), tak odpočítame obranu (random číslo z intervalu).

Proces útoku vyzerá teda nejak takto:

1. Vypočítam útok ako random číslo z intervalu (čo je v intervale sme riešili v 12. kroku)
2. Zistím, či je útok kritický - ak áno tak vynásobím útok tromi
3. Vypočítam obranu protivníka ako random číslo z intervalu (čo je v intervale sme riešili v 12. kroku)
4. Vypočítam finálny útok ako útok - obrana
5. Zmenším život protivníka o výsledný útok. Ak sme protivníka zabili, tak je koniec, inak vymeň kto bude útočiť a brániť a opakuj proces

Pomôcka

Kód, ktorý sa vykoná, keď je na rade útočiť bojovník.

```

if hero_turn:
    min_attack, max_attack = hero["attack"]
    attack = random.randint(min_attack, max_attack)
    if is_critical_hit(hero["critical_hit"]):
        attack *= 3
        print("ÚTOČIŠ KRITICKÝM ÚTOKOM")

    min_defence, max_defence = enemy["defence"]
    defence = random.randint(min_defence, max_defence)
    final_attack = max((attack - defence), 0)

    if final_attack > 0:
        print("Zaútočil si útočnou silou - ", final_attack)
        enemy["health"] -= final_attack

        if enemy["health"] > 0:
            print(enemy["name"] + " po tvojom útoku stále žije. Súperov
zvyšok života - ", enemy["health"])
        else:
            time.sleep(1)
            print("ZVÍŤAZIL SI!\n")
            print(game_constants.DIVIDER)
            return True, hero["health"]
    else:
        print("Netrafil si...")

```

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#)

14. Prechod medzi bitkami (levelmi)

Čo treba spraviť

Momentálne keď nám skončí bitka, tak chýba nejaký medzikrok. Chceme dať po každej bitke hráčovi pár možností. Možnosť oddýchnuť si a doplniť život. Možnosť uložiť alebo ukončiť hru, a tiež možnosť upraviť hrdinu. Vytvárali sme už *phase_check*, tak teraz vytvoríme *battle_check*, ktorý sa zavolá ako medzikrok po každej bitke, či už vyhratej alebo prehranej.

Popis

Na konci tohto kroku už bude hra plne funkčná. Len to bude ešte nemožné prejsť, pretože nemáme predmety, čiže bojovník bude príliš slabý už v 3. alebo 4. kole. (Tak som navrhol *enemies* resp ich dáta).

Okrem iného si vytvoríme konštantu do *phase_constants* `WON_GAME = "WON_GAME"`
 A do *game_constants* si vytvoríme `BOSS_FIGHT_LEVEL = 5 # Last fight with boss`

Medzi bitkami dáme možnosť hráčovi oddýchnuť a dobiť život. To by sme mohli vyriešiť napríklad tak, že dáme *sleep* na 3 sekundy (pauzujeme hru v podstate) a potom mu zväčšíme život o 20 bodov (nie viac ako jeho maximum).

Uloženie a načítanie hry treba aktualizovať o *fight_level*.

Ukážme si, ako to vyzerá v konzole po vyhratej bitke.

```

Zaútočil si útočnou silou - 4
ZVÍŤAZIL SI!

-----

Po víťaznej bitke ti ostal život 36/55
Po pridávaní bodov a voľby predmetu si budeš môcť doplniť život.
-----

Po tvojej 1. výhre, ti pridávam 1 body, ktoré môžeš využiť na upravenie tvojho hrdinu.
Petko, Tvoje schopnosti sú momentálne na tom takto:
Útočná sila - 3, Obrana - 2, Obratnosť - 2, Skill - 2, Život - 55, Šťastie - 2,

Máš 1 bodov, ktoré si rozdeľ naprieč schopnostiam podľa svojich preferencií.
0 - Vysvetlivky - načo su dobré jednotlivé schopnosti?
1 - Útočná sila
2 - Obrana
3 - Obratnosť
4 - Skill
5 - Život - jeden bod pridá 5 života
6 - Šťastie
Máš 1 možností na zlepšenie. Ktorú schopnosť chceš vylepšiť? █

Vybral si si schopnosť - Obratnosť. Pridávam ti bod.
Teraz tvoje body vyzerajú takto:
Útočná sila - 3, Obrana - 2, Obratnosť - 3, Skill - 2, Život - 55, Šťastie - 2,

Výborne Petko. Dokončil si pridávanie schopnosti. Pre rekapituláciu, teraz vyzeraju tvoje schopnosti takto.
Útočná sila - 3, Obrana - 2, Obratnosť - 3, Skill - 2, Život - 55, Šťastie - 2,

-----

0 - Oddýchnuť si - Obnoviť 20 života. Do plného života ti chýba 19 bodov
1 - Bojovať proti - Golem (LVL 2)
2 - Upraviť hrdinu
3 - Uložiť hru
4 - Ukončiť hru
Aký je tvoj ďalší krok?: 0
Dobrá voľba, oddýchni si
Tvoj život je teraz na tom takto: 55/55

0 - Oddýchnuť si - Máš síce plný život, ale oddych nikdy nie je zlý
1 - Bojovať proti - Golem (LVL 2)
2 - Upraviť hrdinu
3 - Uložiť hru
4 - Ukončiť hru
Aký je tvoj ďalší krok?: |

```

Teraz si ukážme, ako to vyzerá po prehratej bitke.

```

SÚPER ÚTOČÍ KRITICKÝM ÚTOKOM
Súper zaútočil útočnou silou - 6
PREHRAL SI

-----

Potrebuješ si oddýchnuť a možno aj prehodnotiť svoje schopnosti, máš 0 života
-----

0 - Oddýchnuť si - Obnoviť 20 života. Do plného života ti chýba 55 bodov
1 - Bojovať proti - Mocny demon (LVL 3)
2 - Upraviť hrdinu
3 - Uložiť hru
4 - Ukončiť hru
Aký je tvoj ďalší krok?: |

```

Síce sa ešte asi nedá vyhrať hra (ak si si nezmenil/a dáta), ale implementuj už ukončenie celej hry. To sa stane vtedy, keď *fight_level* je rovný *BOSS_FIGHT_LEVEL* a zároveň bola bitka vyhnaná. Môžeš potom presunúť hry do novej fázy *WON_GAME*.

Pomôcka

Pauznúť hru, resp spraviť sleep vieš urobiť nasledovne.

```
import time

time.sleep(3)
```

Tento kód zastaví program na 3 sekundy.

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#)

15. Vytvorenie predmetov

Čo treba spraviť

Do dokončenia hry nám chýbajú ešte predmety. V tomto kroku ich iba vytvoríme, resp vytvoríme nový súbor a pacneme tam predmety podobne, ako sme to spravili s nepriateľmi.

Popis

Nasledujúce riadky skopíruj a vlož do novovytvoreného súboru *item_data.py*. Pre tento súbor som aj vytvoril zvlášť adresár *item*, kám som súbor umiestnil.

```
items = {
    1: {
        1: {
            "name": "Nápoj zdravia (LVL 1)",
            "description": "Pridáva 15 zdravia. Ak pri súboji zomrieš,
tak ti nápoj vyprchá. Môžeš mať viacero nápojov naraz.",
            "ability": {
                "Život": 15
            },
            "tag": "drink",
            "replaceable": False,
            "destroy_when_lost": True
        },
        2: {
            "name": "Látkové brnenie (LVL 1)",
            "description": "Pridáva 2 obrany. Zostáva ti do konca hry,
jedine že by si ho nahradil iným brnením.",
            "ability": {
                "Obrana": 2
            },
            "tag": "armor",
            "replaceable": True,
            "destroy_when_lost": False
        },
        3: {
            "name": "Topánky (LVL 1)",
            "description": "Pridáva 1 Obratnosť. Zostáva ti do konca
hry, jedine že by si ich nahradil inými topánkami.",
            "ability": {
                "Obratnosť": 1
            },
            "tag": "boots",
            "replaceable": True,
```



```

        "destroy_when_lost": False
    },
},
2: {
    1: {
        "name": "Nápoj šťastia (LVL 2)",
        "description": "Pridáva 5 stastia. Ak pri súboji zomrieš,
tak ti nápoj vyprchá. Môžeš mať viacero nápojov naraz.",
        "ability": {
            "Šťastie": 5
        },
        "tag": "drink",
        "replaceable": False,
        "destroy_when_lost": True
    },
    2: {
        "name": "Krátky meč (LVL 2)",
        "description": "Pridáva 3 útočnej sily. Zostáva ti do
konca hry, jedine že by si ho nahradil iným mečom.",
        "ability": {
            "Útočná sila": 3
        },
        "tag": "weapon",
        "replaceable": True,
        "destroy_when_lost": False
    },
    3: {
        "name": "Tréning gladiátora (LVL 2)",
        "description": "Pridáva 2 body skillu. Schopnosti ti
nevyprchajú.",
        "ability": {
            "Skill": 2
        },
        "tag": "training",
        "replaceable": False,
        "destroy_when_lost": False
    },
},
},
3: {
    1: {
        "name": "Gumidžús (LVL 3)",
        "description": "Pridáva 2 body útočnej sile, 2 body obrane
a 2 body skillu. Ak pri súboji zomrieš, tak ti nápoj vyprchá. Môžeš
mať viacero nápojov naraz.",

```

```

    "ability": {
      "Útočná sila": 2,
      "Obrana": 2,
      "Skill": 2,
    },
    "tag": "drink",
    "replaceable": False,
    "destroy_when_lost": True
  },
  2: {
    "name": "Dlhý meč (LVL 3)",
    "description": "Pridáva 5 útočnej sily. Zostáva ti do
konca hry, jedine že by si ho nahradil iným mečom.",
    "ability": {
      "Útočná sila": 5
    },
    "tag": "weapon",
    "replaceable": True,
    "destroy_when_lost": False
  },
  3: {
    "name": "Kožené topánky (LVL 3)",
    "description": "Pridáva 2 Obratnosťi a 2 body obrany.
Zostávajú ti do konca hry, jedine že by si ich nahradil inými
topánkami",
    "ability": {
      "Obratnosť": 2,
      "Obrana": 2,
    },
    "tag": "boots",
    "replaceable": True,
    "destroy_when_lost": False
  },
  4: {
    "name": "Ocelove brnenie (LVL 3)",
    "description": "Pridáva 5 obrany. Zostáva ti do konca hry,
jedine že by si ho nahradil iným brnenim.",
    "ability": {
      "Obrana": 5
    },
    "tag": "armor",
    "replaceable": True,
    "destroy_when_lost": False
  },
},

```

```

},
4: {
  1: {
    "name": "Všehodžús (LVL 4)",
    "description": "Pridáva 30 Života a 10 stastia. Ak pri
súboji zomrieš, tak ti nápoj vyprchá. Môžeš mať viacero nápojov
naraz.",
    "ability": {
      "Život": 30,
      "Šťastie": 10,
    },
    "tag": "drink",
    "replaceable": False,
    "destroy_when_lost": True
  },
  2: {
    "name": "Dvojhľavá sekera (LVL 4)",
    "description": "Pridáva 10 útočnej sily. Zostáva ti do
konca hry, jedine že by si ju nahradil inou zbraňou",
    "ability": {
      "Útočná sila": 10
    },
    "tag": "weapon",
    "replaceable": True,
    "destroy_when_lost": False
  },
  3: {
    "name": "Tréning Spartana (LVL 4)",
    "description": "Pridáva 4 body skillu. 2 body útočnej sile
a 2 body obrane. Schopnosti ti nevyprchajú.",
    "ability": {
      "Skill": 3,
      "Útočná sila": 3,
      "Obrana": 3
    },
    "tag": "training",
    "replaceable": False,
    "destroy_when_lost": False
  },
  4: {
    "name": "Rímsky štít (LVL 4)",
    "description": "Pridáva 16 obrany, ale odoberá 2 body
obratnosti. Zostáva ti do konca hry",
    "ability": {

```

```

        "Obrana": 16,
        "Obratnosť": -2
    },
    "tag": "armor",
    "replaceable": False,
    "destroy_when_lost": False
},
}
}

```

Pozorne si všimni, ako sú navrhnuté predmety. Jednak akú majú dátovú štruktúru, ale aj ich vlastnosti.

Itemy majú dátovú štruktúru nasledovnú.

- jednotlivé kľúče predstavujú level po ktorom dané predmety môžeme dostať
- ďalej každý tento kľúč má vlastný slovník, v ktorom máme znova ďalšie vnútorné kľúče (1, 2 alebo 3). Tieto kľúče predstavujú možnosti predmetov, medzi ktorými si budeme vyberať.

Čiže v podstate máme 4 levely predmetov a každý level má 3 rôzne predmety na výber. Čiže hráč bude vyberať predmet po 1. vyhranom kole, 2., 3. a 4. Po poslednom, 5. kole už nebude vyberať predmet, pretože hra skončila.

Čo znamenajú jednotlivé vlastnosti predmetov?

name - názov predmetu

description - popis predmetu

ability - čo robí daný predmet (pridáva tieto schopnosti)

tag - typ predmetu (dôležité, ak je nahraditeľný)

replaceable - nahraditeľnosť predmetu (napr ak si v 2. kole vyberieš meč ako predmet, a potom v 4. si budeš chcieť vybrať sekeru, tak tieto predmety sú toho istého typu a nemôžeš mať 2 rôzne zbrane. Čiže sa ti vymaže meč a namiesto toho dostaneš sekeru. To si samozrejme, ale ešte budeme musieť overiť u hráča, či je s tým zmierený.)

destroy_when_lost - Ak je toto True (nápoje sú toho príkladom), tak hráč tieto predmety po prehranom súboji stratí. Síce sú zaujímavejšie čo sa týka ich schopností, ale hráč pri nich riskuje.

Dáta sa dali samozrejme navrhnuť rôzne a ak máš nejaké nápady, tak ich neváhaj implementovať.

Pomôcka

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#).

16. Získavanie predmetov

Čo treba spraviť

Jéj posledný krok :). Ideme zakomponovať do hry predmety. V podstate po každej vyhratej bitke dáme hráčovi na výber predmety (podľa toho aký level prešiel). Hráč si musí jeden predmet vybrať, a ten predmet zapíšeme do novo vytvorenej premennej *hero_items*.

Okrem iného treba tieto itemy zakomponovať do ukladania a načítavania hry. A tiež treba tieto predmety sem tam vypisovať, aby hráč vedel, ako je na tom.

Popis

Vytvoríme si nový súbor *items.py* v zložke *item*.

V tomto súbore treba naprogramovať tieto funkcionality:

- výber predmetu po výhre
- priradenie predmetu
- kontrola, či predmet nie je replaceable, a ak áno a zároveň už má hráč predmet s rovnakým tagom, tak ho treba upozorniť, či chce predmet nahradiť
- po prehratej bitke implementovať stratu predmetov, ktoré majú *destroy_when_lost* ako True

Tieto funkcionality potom voláme vo fáze FIGHT, keď vyhráme alebo prehráme.

Ukážme si obrázok z hry po vyhranom súboji.

```
Zaútočil si útočnou silou - 6
ZVÍŤAZIL SI!

-----

Po víťaznej bitke ti ostal život 47/55
Po pridávaní bodov a voľby predmetu si budeš môcť doplniť život.

Po výhre máš na výber z 3 predmetov
1 - Nápoj zdravia (LVL 1) -> Pridáva 15 zdravia. Ak pri súboji zomrieš, tak ti nápoj vyprchá. Môžeš mať viacero nápojov naraz.
2 - Látkové brnenie (LVL 1) -> Pridáva 2 obrany. Zostáva ti do konca hry, jedine že by si ho nahradil iným brnením.
3 - Topanky (LVL 1) -> Pridáva 1 Obratnosť. Zostáva ti do konca hry, jedine že by si ich nahradil inými topánkami.
Čo si vyberáš?: 1

Tvoje predmety sa aktualizovali. Teraz vyzerajú takto.
Látkové brnenie (LVL 1) -> Pridáva 2 obrany. Zostáva ti do konca hry, jedine že by si ho nahradil iným brnením.

-----

Po tvojej 1. výhre, ti pridávam 1 body, ktoré môžeš využiť na upravenie tvojho hrdinu.
Ferko, Tvoje schopnosti sú momentálne na tom takto:
Útočná sila - 3, Obrana - 5, Obratnosť - 2, Skill - 2, Život - 55, Šťastie - 1,

Máš 1 bodov, ktoré si rozdeľ naprieč schopnostiam podľa svojich preferencií.
0 - Vysvetlivky - načo su dobré jednotlivé schopnosti?
1 - Útočná sila
2 - Obrana
3 - Obratnosť
4 - Skill
5 - Život - jeden bod pridá 5 života
6 - Šťastie
Máš 1 možnosti na zlepšenie. Ktorú schopnosť chceš vylepšiť? 1
```

Pomôcka

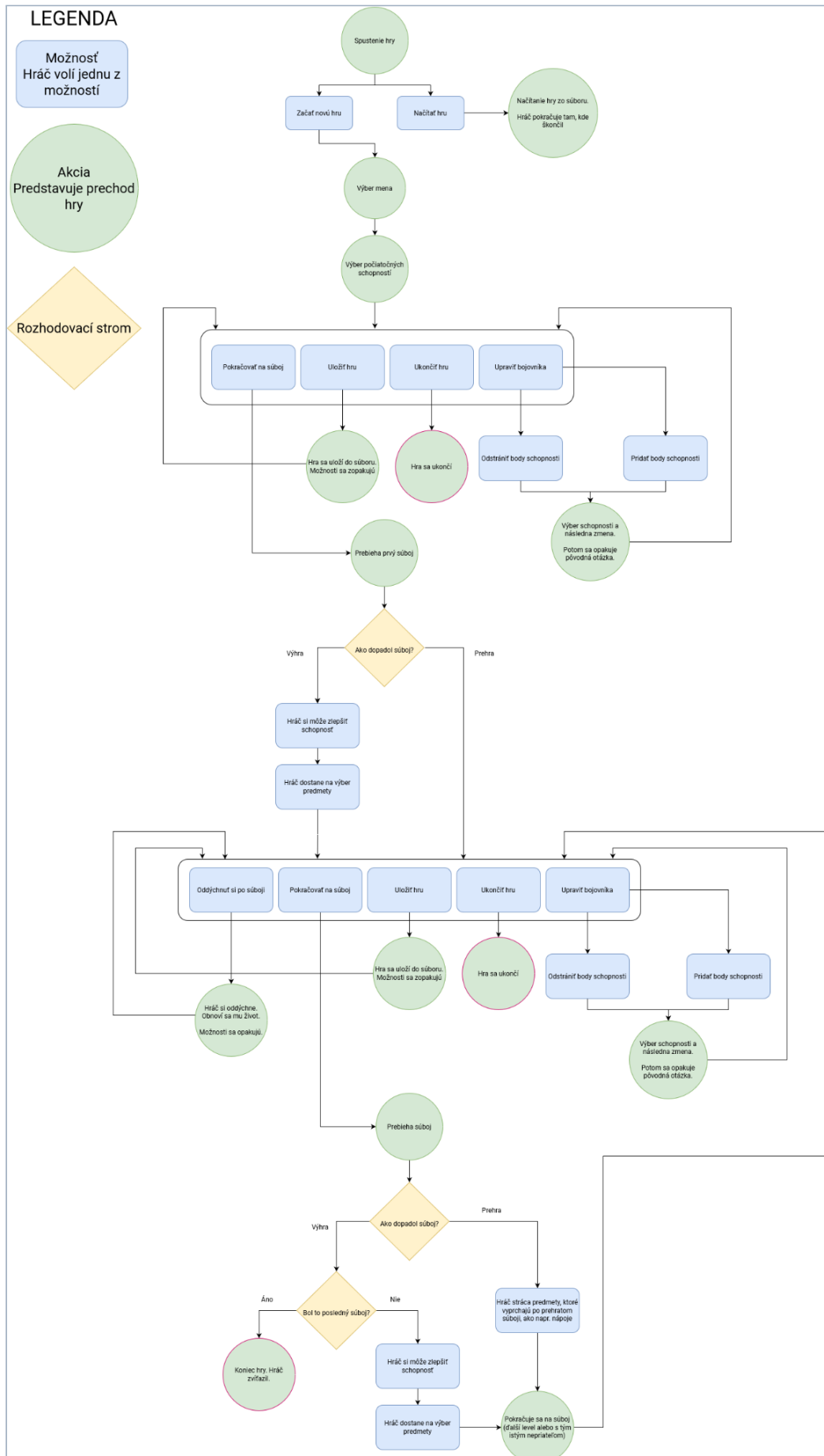
Ako vyriešiť ukladanie a načítavanie hry zo súboru? Navrhujem, aby sme ako posledné riadky súboru mali názvy predmetov, podľa ktorých pri načítavaní budeme hľadať predmety.

Ako vyzerá uložený súbor po vyhratom 1. kole s 0 voľnými bodmi a látkovým brnením?

```
Ferko
Útočná sila - 4
Obrana - 5
Obratnosť - 2
Skill - 2
Život - 47
Šťastie - 1
2
FIGHT
0
Látkové brnenie (LVL 1)
```

Ako som naprogramoval tento krok ja, si môžeš pozrieť v tomto [github commite](#).

Diagram hry



Návrhy na zlepšenia

Počas implementácie hry ti určite napadlo viacero vecí, čo by sa tam dalo ešte vylepšiť. Takisto aj mne. Veľmi by som chcel vidieť, ako sa s tým dokážeš vyhrať. Takže ak spraviš vlastnú verziu hry, kde pridáš všetky tieto zlepšenia, alebo si teda vymyslíš svoje, tak pošli kód zabalený na info@streetofcode.sk, alebo pošli link na svoj github projekt.

Ďalej idem popísať moje nápady na zlepšenia.

Nápoj pred bitkou

V hre by mohli byť špeciálne nápoje, ktoré by mohol hráč použiť iba pred bitkou. Napríklad nápoj agresivity (zväčšil by útok) a podobne.

Zlatky (coins)

Po vyhratej bitke by mohol hráč dostať nejaké coins (a po prehratej stratíť). Za coins by sa dali kupovať predmety.

Špeciálne schopnosti predmetov

Predmety majú teraz dané statické vlastnosti, ktoré ti pridávajú schopnosti. Ale mohli by mať aj nejaké špeciálne, ako napríklad 50% šanca imúnosti voči kritickým útokom, alebo ak máš tento predmet, tak vždy ideš v súboji prvý (alebo druhý, ak ten predmet má iné výhody). Prípadne itemy môžu byť rozdelené podľa kategórií: common, rare, epic, legendary. Každá kategória má inú pravdepodobnosť, že ti padne po súboji daný predmet. A samozrejme so stúpajúcou kategóriou by boli predmety silnejšie.

Globálne nastavenia hráča

Možno by bolo super pridať nastavenia pre hráča. Napríklad konfigurovať rýchlosť vypisovania súbojov, alebo automatické uloženie po každej bitke, prípadne nastaviť obtiažnosť hry.

Interaktívny boj

Teraz akonáhle hráč spustí boj, tak už nad ním nemá kontrolu. Je milión spôsobov, čo by sa dalo vymyslieť, aby sme pridali istý spôsob kontroly nad súbojom. Napríklad pred každým kolom sa hráč môže rozhodnúť, či chce útočiť ostražito alebo agresívne.

Ďalšia celkom srandovná vec by bola, že počas súboja hra vyzve hráča, aby rýchlo napísal nejaké random slovo a bude mať na to napríklad 3 sekundy. Ak hráč stihne to slovo napísať správne, tak bude mať silnejší útok.

Znížiť šťastie po prehre

Po prehranom súboji by sa mohlo znižovať šťastie. Aj keď na to by sa skôr hodila nová schopnosť - morálka. Ako by morálka fungovala už nechám na teba.

Začína v bitke ten, kto má väčšie šťastie

Momentálne vždy začína ako prvý útočiť hráč, ale to nie je veľmi fér voči nepriateľovi, či? Chcelo by to zaviesť nejaké pravidlo. Napríklad, že začína ten, kto má väčšie šťastie, prípadne tam zaviesť aj nejakú randomizáciu do toho. Napríklad ak hráč má šťastie 3 a nepriateľ 2, tak tým pádom by mal hráč 60% pravdepodobnosť, že pôjde ako prvý a nepriateľ 40%.

Viacero nepriateľov, random nepriateľ pre boj

Chcelo by to pridať viacej nepriateľov. Viacej kôl, ako aj viacero možností nepriateľov pre každú úroveň. Voľba nepriateľova pred súbojom by bola buď náhodná, alebo si hráč vyberie, alebo by sme tam zakomponovali nejaký iný faktor.

Fixnúť problém s predmetmi

Momentálne hra obsahuje bug s predmetmi. Hráč môže získať predmety, ktoré sa mu automaticky zarátajú do schopností. Avšak tie schopnosti si môže hráč potom upraviť a premiestniť inam. Čiže fixli by sme to nejak tak, že by sme nerátali hneď predmety k schopnostiam, ale mali to ako zvlášť, aby sa to nedalo takto okašľať.

Závěrečné slová

A to je všetko. Dúfam, že si si pri tom dobre precvičil/a Python, a naučil/a si sa veľa nových vecí.

Chceli by sme ťa v mene Street of Code poprosiť o zanechanie review, ktoré môžeš napísať priamo ku kurzu. Veľmi nás zaujíma, čo si myslíš o tomto kurze, ako sa ti to páčilo, ako aj čo si myslíš o tomto samotnom zadaní.

Dokázal/a si sa z toho vysomáriť? Bolo to pochopiteľné? Nebolo to príliš dlhé? Má takéto zadanie zmysel? Čo by trebalo zlepšiť nabudúce? Bolo to príliš náročné, alebo ľahké?

Prípadne nám prosím, napíš mail, či už na info@streetofcode.sk alebo jakub@streetofcode.sk. Alebo na [Discord](#).

Ďakujeme!