## Dokumentace

## TODO:

* Výpis textu: Teoreticky bychom měli mát dvojí výpis. První, popis události a její vyhodnocení, čekají na proceed, takže by nevadilo, kdyby byl zobrazen v samostatném elementu. Tento element by se objevoval jen pro zobrazení textu. Mohla by to být třeba nějaká obrazovka nebo nějaký kus papíru, pergamenu, tabule, který by obsahoval tlačítko ok. Ten delší text a tlačítko navíc tak nebudou zabírat místo. No, a druhý výpis, takový nějaký log, který by mohl vypisovat efekty, jako třeba dmg a tak, by mohl být v nějaké menší obrazovce v main screenu. Nemusel by zabírat tolik místa, vlastně by stačily dva tři řádky.
* Snížit počet akcí a zároveň přidat tlačítko pro ukončení tahu předčasně.
* Dodělat snižování a zvyšování rychlosti. Nejlépe vždy o jeden bod. A zároveň ukazatel rychlosti.

## Playstate

V konstruktoru je nastaven ukazatel tahu na CPUFIRST a jsou vytvořeny instance CPUlogic a Board. V updatu je realizován turn, kdy se střídá CPUFIRST -> PLAYER -> CPUSECOND. V tomto pořadí. Po dokončení CPUFIRST turnu se vytváří nová instance třídy PlayerLogic, což zaručí vytvoření nového timeru (nejjednodušší způsob, na jaký jsem přišel).

## CPUlogic

Je to logika pro CPU turny. Dělí se to zde dle role.

V první roli se odehrává zdánlivě jen vypsání textu a čekání na proceed. V samotném kódu ale nejprve, pokud není nastavena žádná aktivní událost, nastaví čekání na proceed, poté je vybrána a přiřazena aktivní událost pomocí funkce pickEvent, následuje restart všech hráčem ovlivňovaných statů, jako je dmg, manévry atp. Tyto změny jsou prováděny a samozřejmě i nulovány v GameStatus. Dále aktivní událost nastaví podmínky, pokud nějaké jsou (mimojiné je to i nastavení textu pro label). Je aktualizován label. Je změněn příznak aktivní události na true. Nakonec je spuštěna funkce UnEndEvent. Z nějakého důvodu je přes všechny snahy a vždy příznak Eventu, ending, true. Proto tato funkce změní ending na false, což je nutné pro správné fungování druhé role. Role vrátí true, čímž pokračuje Playstate k opakování této role. Pokud ale je nastavena aktivní událost, pouze se čeká na proceed, který by smyčku ukončil a přepnul Playstate do další role. Do té doby ale stále vrací true, čímž opakuje smyčku, dokud hráč nepotvrdí vypsaný text.

Druhá role – pokud událost ještě nekončí, tak nastaví příznak čekání na proceed na true, ubere kyslík (ubírá jeden kyslík každé kolo), odstraní podmínky původně nastavené událostí, vyhodnotí událost a provede změny nad hráčem (což i nastaví text výsledku události), aktualizuje text, nastaví příznak aktivní události na false, čímž umožní naběhnutí prvního kola, jakmile bude toho ukončeno, ukončí tento event a nechá znovu proběhnout smyčku. Pokud už událost končí, tak testuje na proceed. Možná se to může zdát trochu reduntantní, což možná i je, ale bylo to řešeno asi ve dvě ráno. Stačilo by celou roli 2 řešit jen na proceed a proceed příznak erasnout hned po ukončení role 1.

Je zde funkce pichEvent(). V této funkci je nejprve dotaz na následující scénáře. Pokud nejsou žádné nalezeny tak je náhodně ze switche vybrána událost nebo posloupnost událostí. Všechny nové události nebo posloupnosti událostí by měly být připojeny do tohoto switche.

## PlayerLogic

Toto je logika tahu hráče. Je zde definován maximální počet akcí, timer a return value. V kontruktoru je definován timer (zde je jako callback funkce vytvořena metoda, která nastaví návratovou hodnotu na false, čímž ukončí tah hráče) a zavolána funkce pro erasnutí hodnot hráče. Pozor, redundance s CPUlogic, kde je možné to smazat, ale je možné to tam nechat pro jistotu.

V samotném updatu a se nejprve nastaví hodnoty pro update layoutu a kontroluje se na počet použitých akcí, pokud je již max, tak se ukončí se tah hráče. Pokud ne, opakuje se smyčka. Zároveň ale callback funkce timeru nastaví příznak, který ukončí tah hráče.

## Event

### isEventEnding

Funkce pro CPUlogic, která vrací, zda událost končí. Prakticky to rozhoduje o tom, jestli použít SetConditions nebo UnsetConditions+MakePlayerPay. Není určeno k override.

### EndEvent

Prakticky property proměnné ending. Není určeno k override.

### UnEndEvent

Prakticky property proměnné ending. Není určeno k override.

### DoesContinue

Je určeno k override. Udává to, jestli se má CPUlogic vůbec pokoušet o získání následující události. Po ošetření vracení null při dotazu na následující událost, by se tento krok dal přeskočit a je redundantní.

### GetFollowingEvent

Určeno k override. Tato metoda vrací navazující událost.

### SetConditions

Určeno k override. Nastaví text pro label a případně podmínky prostředí.

### UnsetConditions

Určeno k override. Případně zruší způsobené podmínky prostředí.

### MakePlayerPay

Určeno k override. Vyhodnocení události a nastavení textu pro label.

## Game Status

V této třídě jsou definované statické proměnné pro udržení profilu hráče. Jsou to jednak vlastnosti (život, energie…), které jsou restartovány na začátku hry, jednak hodnoty způsobené hráčem (dmg…), které jsou aktualizovány v průběhu hráčova tahu pro vyhodnocení situace a na konci CPUSECOND jsou nulovány a nakonec jsou zde hodnoty mimo toho, jako třeba proceed.

Jsou zde funkce pro nastavení defaultních hodnot a pro restart hodnot hráče.