Programátorská dokumentace

## Použité technologie

* **Python** - skriptovací programovací jazyk, který podporuje objektový přístup k programování. V současné době existují dvě nekompatibilní verze, Python 2 a Python 3. Pythonu 2 bude v blízkém budoucnu končit podpora.
* **Tkinter** – standardní pythonovská grafická knihovna. Nabízí základní nástroje pro vytváření grafických rozhraní a dodává se standardně s každou instalací Pythonu. Tkinterem je v programu vytvořeno celé grafické rozhraní.
* **PIL** – Python Imaging Library je nadstavbová knihovna pro práci s obrázky. Je kompatibilní pouze s Pythonem 2, pro Python 3 se používá knihovna Pillow, která nahrazuje PIL a je mu velmi podobná

## Rozvržení programu

* **body.py** – zdrojový kód aplikace, je spustitelný v Pythonu 2, obsahuje všechen kód použivaný aplikací
* **reditelna.jpg** – obrázek ředitelny, který se v aplikaci objevuje na hlavní obrazovce
* **licensing** – textový soubor, obsahující plné znění licence GNU General Public License 3.0

## Hlavní třída hry – class root

Ve hře je jediná třída, která se stará o všechno. Jsou v ní nadefinovány všechny funkce i proměnné.

### Funkce, které spouštějí okna

Funkce, kde lze naleznout definice nějakých oken. Jsou uvedeny v pořadí, v jakém se objevují ve zdrojovém kódu.

* **\_\_init\_\_(self)** -v konstruktoru třídy root se definuje hlavní okno a veškerý jeho obsah, taky se zde definují fonty a některé seznamy a proměnné, které se používají ve zbytku programu
* **popupmsg(self,msg)** – funkce popupmsg obstarává malé okno, které hlásí konec hry a nebo vrcholnou nespokojenost některé skupiny lidí. Proměnná *msg* je zpráva, kterou má oko napsat.
* **trophies()** –funkce, která definuje okno s trofejemi. Pomocí cyklu for vytvoří *trophies()* label pro každý prvek ze seznamu *self.trophy\_list* (nadefinovaný jako prázdný v konstruktoru), který se plní úspěchy školy v meziškolních soutěžích.
* **upgrades()** - ve funkci upgrades se definuje okno s vylepšeními. Stiskem tlačítka u jednotlivých oblastí vylepšení se spouští funkce, které vylepšují vybavení (*sport\_upgrade(),veda\_upgrade(),vedomosti\_upgrade()*).
* **zmenit\_skolne()** – v rámci této funkce se řeší okno, které spouští funkci *s\_zmen()*, která provádí samotné výpočty týkající se změn školného.
* **zmenit\_platy()** – tato funkce funguje velmi podobně jako *zmenit\_skolne()*, ale vyvolává funkci *p\_zmen()*
* **competition()** – rozsáhlá funkce, která zařizuje meziškolní soutěže. V rámci funkce existuje seznam *self.pos\_values*, ve které jsou, čísla od jedné do sta. Na začátku meziškolní soutěže se pro školu hráče vybere náhodné číslo mezi 0 a 100. Číslo je poté zvýšeno o modifikátor, který je specificky vypočítaný pro každou oblast soutěží (level vylepšení x 2.75). Výsledné číslo je skóre hráčovy školy a odebere se z *self.pos\_values*. Dále se stejným způsobem vytvoří náhodná skóre pro šest dalších škol a všechna čísla se vloží do seznamu *self.results*. Dále se jednoduchým cyklem while určí, kolikátá skončila hráčova škola. Vždy se projede celý seznam a odebere se nejvyšší hodnota. Za každou odebranou hodnotu se o jedno zvýší proměnná *self.place*, která uchovává umístění hráčovy školy. Když se ukáže, že nejvyšší skóre v seznamu je skóre hráčovy školy, cyklus skončí a umístění školy se pošle do okna s výsedky. Přes seznam se funkce dělá, protože je to nejjednodušší způsob, jak zajistit, aby neměly dvě školy stejné skóre a nevznikaly zmatky.
* **event\_class1,2,3,4(argumenty)** – *event\_class* přijímá vstup od funkce *random\_*event a dává hráči grafickou informaci o náhodných událostech. Dělí se do 4 podfunkcí, každá pracuje samostatně. Funkce jsou pojmenovány podle toho, kolik přijímají následků náhodných událostí. *Event\_class1* přijímá jeden následek, *event\_class2* přijímá dva následky, *event\_class3* bere tři a *event\_class4* bere čtyři následky. Funkce potom vytvoří okno, kde napíše co se stalo a všechny následky události.

### Pomocné funkce

V této sekci jsou popsány pomocné funkce, které nějak zjednodušují práci programu, nebo provádějí drobné úpravy. Jsou uvedeny v pořadí, v jakém se objevují ve zdrojovém kódu.

* **refresh()** – funkce, která aktualizuje všechno info na hlavní obrazovce. Obsahuje jak změny v nápisech, tak grafické úpravy ukazatelů spokojeností. Volá se ve většině funkcí, vždy když se mění proměnná, která je vypsána na hlavní obrazovce.
* **center(jmeno\_okna)** – grafická funkce. Zajišťuje že okno, které jí volá, se bude při vytvoření objevovat uprostřed obrazovky.
* **game\_over\_duvod()** – funkce, která se vyvolá, když končí hra. Ať už se jedná o konec dobrý či špatný, funkce vyvolá okno popupmsg
* **end()** – funkce zavírající popupmsg i hlavní okno, volá se skrze popupmsg a ukončuje hru

### Negrafické hlavní funkce

Jedná se o funkce, které v sobě obsahují výpočty a úpravy, které zásadně ovlivňují chod hry. Jediné úpravy, které v těchto funkcích souvisí s grafikou jsou úpravy nápisů. Jsou uvedeny v pořadí, v jakém se objevují ve zdrojovém kódu.

* **expand()** – navýšení kapacity školy. Každá expanze stojí více podle vztahu *level žáků\*850 000*. *Expand* také snižuje spokojenost učitelů, protože se musí starat o víc žáků a mají víc starostí.
* **Funkce *oblast\_upgrade()*** – tyto tři funkce realizují vylepšení školního vybavení (sport, věda, vědomosti). Každé vylepšení má dynamickou cenu, která se odvíjí od současného levelu konkrétního vybavení (*(vybaveni\_level+1)\*18 000*). Každé vylepšení taky zvyšuje spokojenosti o 5 u dvou skupin, ale u jedné skupiny o 10 (sport u studentů, věda u učitelů a učební pomůcky u rodičů).
* **s\_zmen()** – funkce, která mění školné. To se děje tak, že se do proměnné *s\_output* uloží hráčem napsaný obsah pole v okně změny školného. Dále se zkontroluje, že je školné v povoleném rozmezí od 500 korun žák/měsíc až do 3500 korun žák/měsíc (když se hodnota nepohybuje v daném rozmezí, v okně změny školného se objeví chybová hláška). Společně se školným se mění i spokojenost rodičů. Změna v částce školného se vydělí dvěma sty a výsledné číslo představuje změnu ve spokojenosti, která se uloží do proměnné *self.i*. Když se školné snižuje, spokojenost se zvyšuje a naopak. Pro případ, že by hráč posunul školným o méně než 200 je v *s\_zmen()* jednoduchá pojistka. Ta by *self.*i nastavila na hodnotu rovné jedné v případě, že by se *self.i* rovnalo nule.
* **p\_zmen()** – *p\_zmen*  mění výšku platů. Je voláno stiskem tlačítka v okně změny platů a funguje velmi podobně, jako *s\_zmen*. Jediný rozdíl je, že platy učitelů nemají horní strop a pro výpočet změn spokojenosti se nepoužívá 200, ale 300.
* **vpred()** – nejdůležitější funkce ve hře. Spouští se stiskem tlačítka „vpřed“ a dělá velkou spoustu věcí. Na začátku zkontroluje, jestli neuběhlo 44 týdnů, po kterých přijde větší časový skok, simulující prázdniny. Potom zkontroluje, jestli neuběhlo 97 týdnů, což by znamenalo šťastný konec hry. Dále každé 4 týdny provede funkce vyplacení platů učitelům, zaplatí nájem, ale také vybere školné. Když je po těchto výpočtech škola finančně v mínusu, tak zkrachuje a nastává konec hry. Když doteď nenastal konec hry, tak funkce zkontroluje, jestli jsou v pořádku spokojenosti. Když dvě z nich klesnou pod 15, tak je ředitel odvolán a hra končí. Když je nějaká spokojenost menší než 25, tak funkce vpřed hráče varuje, že by měl něco změnit a spokojenost zvýšit. Poslední dvě činnosti v kompetenci funkce *vpred* je volání funkce meziškolních soutěží (*competition*) a funkce s náhodnými událostmi (*random\_event*). Meziškolní soutěže jsou volány upraveným náhodným výběrem (když je počet týdnů bez soutěže + náhodné číslo od 1 do 10 větší nebo rovno 13, koná se meziškolní soutěž). Pro oba testy se využívá stejná proměnná *self.calm\_week*, která vyjadřuje počet týdnů od poslední soutěže nebo události. Když se nerovná nule, může se stát buď soutěž, nebo náhodná událost, ale ne obojí.
* **random\_event(eve)** – funkce, která obsahuje všechny procesy související s náhodnými událostmi. Jako argument přijme *random\_event* od funkce vpred číslo, které se uloží do proměnné *eve.* Tato funkce vysílá argumenty pro funkce *event\_classcislo()*. Hlavní částí funkce je *if*, který zjistí, jaká náhodná událost se má stát. Když se zjistí, která událost se stala, tak se v rámci *random\_event()* provedou procesy související se změnami proměnných (spokojenosti, peníze, vylepšení). Potom se spustí funkce *event\_classcislo* podle toho, kolik má událost argumentů a jako parametry se funkci pošle, co se bude zobrazovat v okně oznámení hráči.