

# Dokumentácia k programu Tic-Tac-Toe

Norbert Horváth, I. ročník, MFF UK, IPP

Zimný semester, 2021/2022,

Programování 1, NPR030

Jazyk: Python

## Zadanie programu:

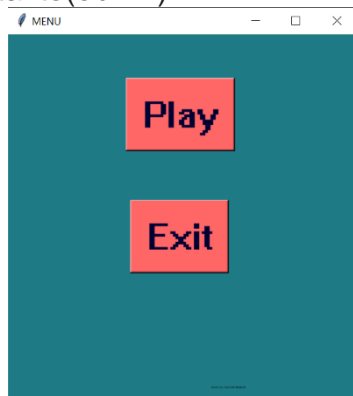
**Základ programu je hra Tic-Tac-Toe.**

UI programu vytvorím s pomocou knihovni **Tkinter**. Hráč bude vždy začínať a dostane X, počítač bude dostane O, a bude nasledovať hráča. V druhom kole, roly sa vymenia, a počítač dostane X, a začne, a hráč dostane O, bude nasledovať ťah počítača. Po prvých 2 kolách, hra je to isté, hráč a počítač sa vymieňajú.

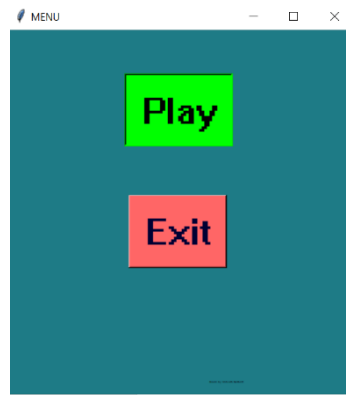
Program bude mať jeden menu, kde budú dva opcie: **START** a **STOP**. S tlačidlom **START** začnem hru. S tlačidlom **STOP** ukončím aplikáciu. Program bude mať tzv. Game Screen, kde bude tlačidlo **STOP** a **NEXT ROUND**. S tlačidlom **STOP** ukončím aplikáciu. S tlačidlom **NEXT ROUND** začnem druhé kolo. Na obrazovke ešte nájdeme skóre, ktoré bude počítat koľko má hráč a počítač víťazstiev, a koľko je remíz.

## UI (User Interface) programu:

Využíval som modul **Tkinter**, ktorý je v **pythonu** vpred nainštalovaný. Menu vyzeral takto(obr. 1):



Obrázok 1



Obrázok 2

Na obrázku vidíme 2 tlačidlá, ktorých význam už som vysvetľoval v kapitole “Základ programu”. Obrazovka má nadpis **MENU**, ktorý som nastavil pomocí:

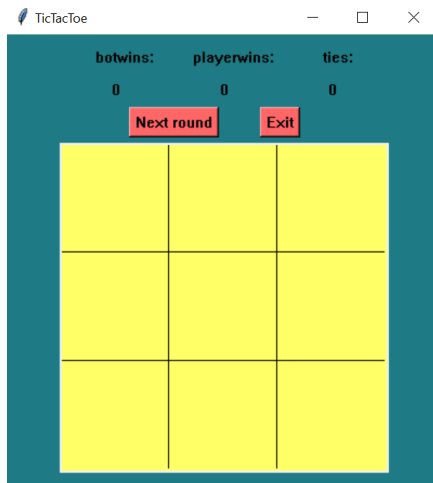
```
menu.title('MENU')
```

Tlačidlá som nastavil tak, aby na stlačenie pozadie tlačidla sa zmenil na

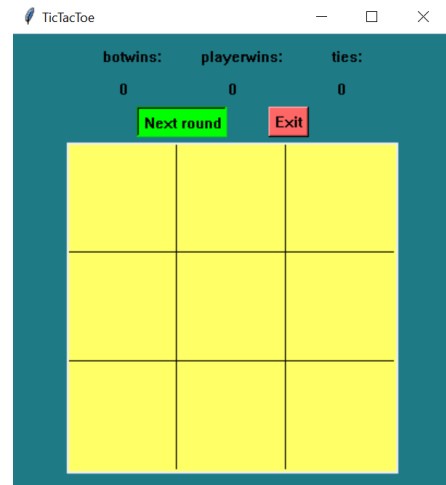
zelenú(obr. 2), čo som nastavil pomocí:

```
activebackground="#00FF00"
```

Na stlačenie tlačidla dostaneme na tzv. Game Screen:



Obrázok 3



Obrázok 4

Na obrázku môžeme vidieť nasledujúce(obr. 3):

*botwins* – spočíta koľko krát vyhral počítač.

*playerwins* – spočíta koľko krát vyhral hráč (kvôli algoritmu minimax, nikdy nenastane)

*ties* – spočíta koľko krát bola remíza

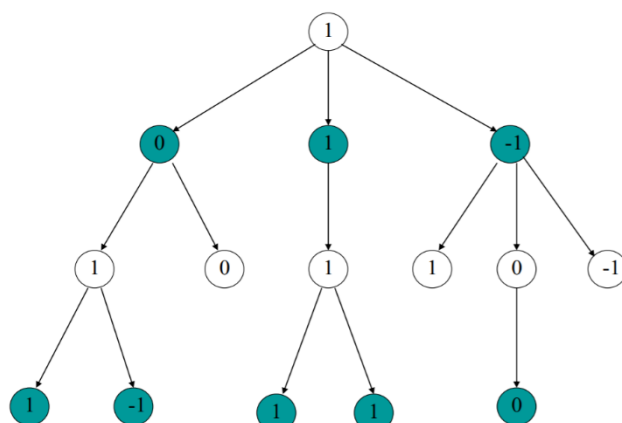
**Next round** – začne nasledujúce kolo

**Exit** – skončí program

Ako pri menu, aj tu je implementované zmena farby tlačidla pri stisknutí(obr. 4).

## Popis algoritmu:

Vybral som algoritmus minimax, s ktorým počítač pracuje. Minimax je algoritmus, ktorý má určiť najlepšiu stratégiu hry. Minimax dopredu zahrá rôzne postupnosti ťahov a snaží sa nájsť takú postupnosť, ktorá vedie k výhre počítača. Môj program najprv kontroluje kto je na rade, potom, počítač dostane body podľa výherca. Ak počítač vyhrá, tak dostane 1 bod, a ak nevyhrá, dostane -1 bod. Počítač bude vždy sa snažiť dostať čo najlepší výsledok, ktorý je v tomto prípade 1 bod. Urobíme strom hry, ktorý bude vyzeráť napríklad takto(obr. 5):



Obrázok 5

Zdroj: <https://ksvi.mff.cuni.cz/~topfer/>

Počítač si zvolí tú cestu, ktorý je ten najbenefiční pre počítača. V hre, ktorý som vybral, tic - tac – toe, strom hry, je tak malý, že počítač môže prejsť všetky možné výsledky, a tak môže nájsť ten výsledok, v ktorej počítač ma najlepší výsledok, a hráč ten najhorší. V našej prípade to bude znamenať, že hráč nebude nikdy vyhrať, pre hráča, najlepší výsledok je remíza. Ak hráč hrá vždy perfektne, tak vždy bude len remíza.

## Program:

### Funkcie – Ich význam

**Gameobrazovka** - definuje celý Game Screen, ktoré je hlavná obrazovka.

**ai** – prepne AI na True, a urobí prvý ťah pre počítača.

**dostupne\_pohyby** – urobí list s dostupnými štvorcami.

**skoncit** – kontroluje ak hra sa skončila

**vyherca** – vráti výherca

**pozice** – vráti všetky pozície, kde hráč má X alebo O

**tah** – pridá to štvorca X alebo O

**minimax** – definuje algoritmus minimax

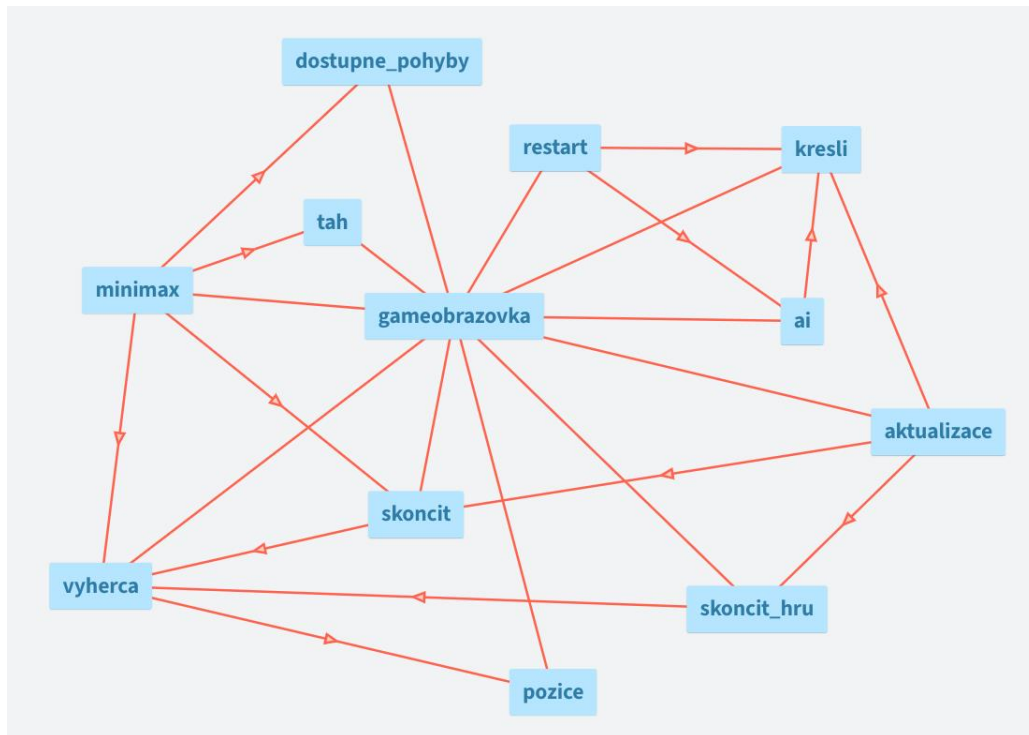
**aktualizace** – aktualizuje tabuľku podľa toho, kto je na rade

**kresli** – nakreslí aktualizácie, podľa funkcie aktualizace

**skoncit\_hru** – skončí hru, a hráč dostane správu podľa toho, že kto vyhral.

**restart** – reštartuje hru, a zavolá funkciu ai, ak hráč začal.

Pomocí *orgpadu*, som vizualizoval spojenie medzi funkciami(obr.6), šípky zobrazujú že ktorá funkcia zavolaala ktorú



Obrázok 6

### User Input:

Ja som použil koordináty, s ktorým som pracoval, čiže, ak hráč klikne na obrazovku, tak kontroluje program, že kde má ten klik koordináty, a podľa toho aktualizuje obrazovku. Obrazovku aktualizuje aj podľa toho, že kto začal. Ak hráč začal, tak obrazovku pri kliknutí nájde vhodný štvorec, a vypíše tam X. Ak hru začal počítač, tak ak hráč je na ťahu, tak podľa toho bude obrazovka aktualizovaná s O. Po skončení kola, môžeme stlačiť tlačidlo „Next round“, s ktorým začneme nasledujúce kolo. Je možné, aby hráč stlačil tlačidlo „Next round“, ale tak nedostaneme vyhodnotenie a rovno sa začne nasledujúce kolo,

### Priebeh práce:

Najväčšiu časť zobrať študovanie minimax algoritmu, a implementovanie ho. Potom s využitím Tkinter, som vypracoval hlavnú obrazovku. Po hlavnej obrazovke, som urobil menu, ktoré odstránim, ak hráč stiskne tlačítko „START“. Nakoniec som implementoval striedavosť programu, to znamená, že program strieda, že kto je na rade ako prvý.

### Na záver:

Ja si myslím, že som naprogramoval celkom dobrý program. Hlavne som veľa vecí nacvičil týkajúce Tkinter, o ktorom pred zápočtovým programom skoro nič nevedel. Algoritmus minimax mi veľakrát spôsobil bolesť hlavy, ale urobil som čo najlepšie čo

som mohol urobiť. Veľa vecí som sa naučil počas práce s mojim zápočtovým programom, a tieto skúsenosti určite využijem pri práci s niektorým iným projektom.