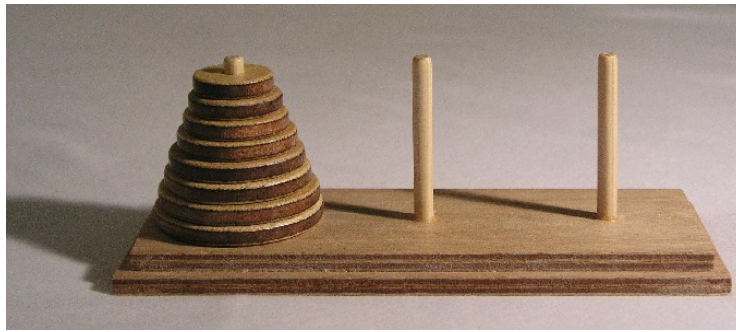


# Hanojské věže



Dřevěný model hry

## Princip hry:

Hanojské věže (Tower of Hanoi) je matematický hlavolam, který vymyslel francouzský matematik Édouard Lucas v roce 1883. Skládá se ze tří kolíků (věží). Na začátku je na jednom z nich nasazeno několik kotoučů různých poloměrů, seřazených od největšího (vespod) po nejmenší (nahore). Úkolem hry je přemístit všechny kotouče na třetí věž (druhou věž přitom využije jako pomocnou pro dočasné odkládání) podle následujících pravidel:

- V jednom tahu lze přemístit jen jeden kotouč.
- Jeden tah sestává z vzetí vrchního kotouče z některé věže a jeho položení na vrchol jiné věže.
- Je zakázáno položit větší kotouč na menší.

*Převzato a upraveno z wikipedie.*

Jako zdrojovou hru jsem zvolil zdrojový kód dostupný na stránce [https://github.com/coderford/hanoi\\_pygame](https://github.com/coderford/hanoi_pygame)

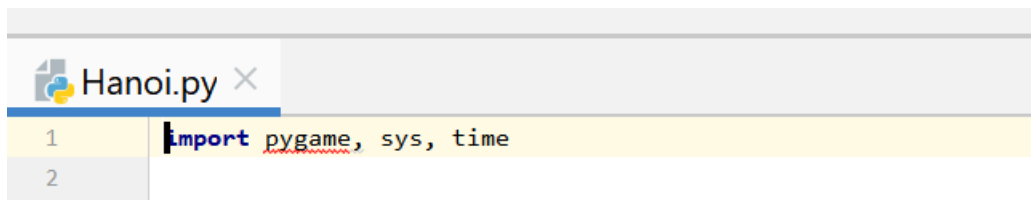
## Postup zprovoznění:

Pro zprovoznění a úpravu hry jsem zvolil vývojové prostředí PyCharm verze 2019.2.

Po spuštění vývojového prostředí vytvořím Nový projekt a pojmenuji ho „Hanoi“.

V projektu vytvořím nový soubor „Hanoi.py“ do kterého ze zdrojové stránky přepokopíruji zdrojový kód hry.

Pokusím se o první spuštění hry, která se nepodaří, protože překladač postrádá jednu z knihoven nutných pro běh hry.

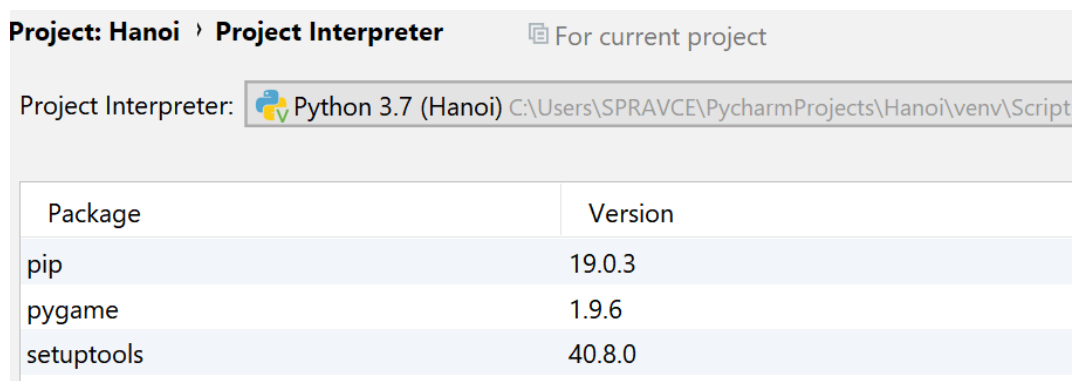


*Nenalezená knihovna Pygame*

Pro běh hry je potřeba doinstalovat knihovnu Pygame, což je multiplatformní sada modulů jazyka Python určená k tvorbě počítačových her. Obsahuje knihovny pro práci s grafikou, zvukem a vstupními zařízeními určené pro použití v jazyce Python.

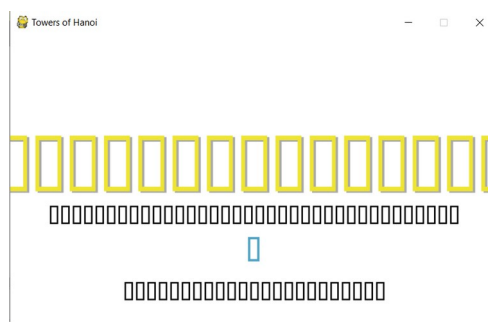
Instalace knihovny se provádí přímo ve vývojovém prostředí PyCharm podle následujícího postupu:

1. File – Settings
2. Project: Hanoi
3. Project Interpreter, kde v dialogu vyhledám knihovnu „Pygame“ a nainstaluji ji do projektu Hanoi.



*Nainstalovaná knihovna Pygame*

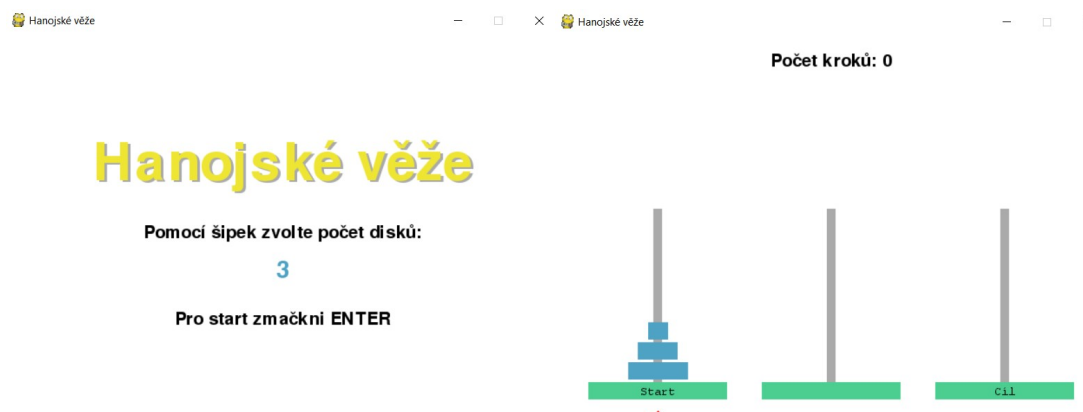
Nyní se pokusíme hru spustit znovu, spuštění proběhne bez problému, ale místo textu se zobrazí pouze obdélníčky, což signalizuje pravděpodobně problém s příslušným fontem.



*Ve hře se nezobrazuje text*

Po analýze zdrojového kódu je problém s fontem pro písmo „sans serif“, které pravděpodobně nemám v počítači nainstalované, nebo se špatně zobrazuje. Problém

vyřeším nastavením výchozího systémového písma. Při nastavování výchozího písma, přeložím i jednotlivé dialogy použité ve hře.



*Ukázka úvodního menu a hry*

### Průběh hry:

Hra se ovládá pomocí klávesnice (kurzorové šipky, Enter a klávesa Esc).

Po spuštění hry si v úvodním menu nastavíme pomocí šipek nahoru a dolů obtížnost od 1 do 6 (počet disků). Klávesou Enter spustíme hru. Ve vlastní hře přesouváme jednotlivé disky pomocí šipek, nejdříve vybereme věž, ze které budeme přesouvat disk, šipkou nahoru zvedneme disk, vybereme cílovou věž a šipkou dolů disk uložíme (jeden krok hry). Samotná hra kontroluje, zda jsou dodržena příslušná pravidla hry a nedovolí nepovolené kroky (např. na menší disk uložit větší disk).

Po přesunutí všech disků na cílovou věž se hra ukončí a vyhodnotí počet kroků použitých během hry a porovná s nejmenším možným počtem kroků.

Hra je omezena na 6 disků a to z toho důvodu, že počet kroků s každým novým diskem roste exponenciálně.

Počet disků	3	4	5	6	7	8	9	10	11
Počet tahů	7	15	31	63	127	255	511	1023	2047

Hru lze kdykoliv resetovat klávesou Esc.

### Analýza kódu:

Hra je naprogramovaná pomocí imperativního programování, kde jsou v kódu využity především cykly (for, while), větvení (if) a přiřazení hodnot do proměnných.

Program je pro větší názornost rozčleněn do několika uživatelsky definovaných funkcí, které jsou navzájem volány.

Uvádím jejich definici a účel

```
# funkce, která slouží k zobrazení textu na příslušných souřadnicích, velikosti, fontu a barvě)
```

```
def blit_text(screen, text, midtop, aa=True, font=None, font_name=None, size=None, color=(255, 0, 0)):
```

```
#funkce, která je spuštěna před začátkem hry, zde si hráč nastavuje obtížnost hry
```

```
def menu_screen():
```

```
#funkce, volána po ukončení hry, slouží k zobrazení výsledku a ukončení hry
```

```
def game_over():
```

```
#funkce, která vykreslí tři věže
```

```
def draw_towers():
```

```
#funkce, která vykreslí tři věže
```

```
def draw_towers():
```

```
#funkce, která vytvoří počáteční umístění disků na první věži
```

```
def make_disks():
```

```
#funkce, která vykreslí jednotlivé disky na jednotlivé věže
```

```
def draw_disks():
```

```
#funkce, která vykreslí ukazatel (červený trojúhelník) k vybrané věži
```

```
def draw_ptr():
```

```
#funkce, která zjistí, zda jsou přesunuty všechny disky na cílový disk
```

```
def check_won():
```

```
#funkce, která resetuje všechny proměnné a spustí hlavní menu
```

```
def reset():
```

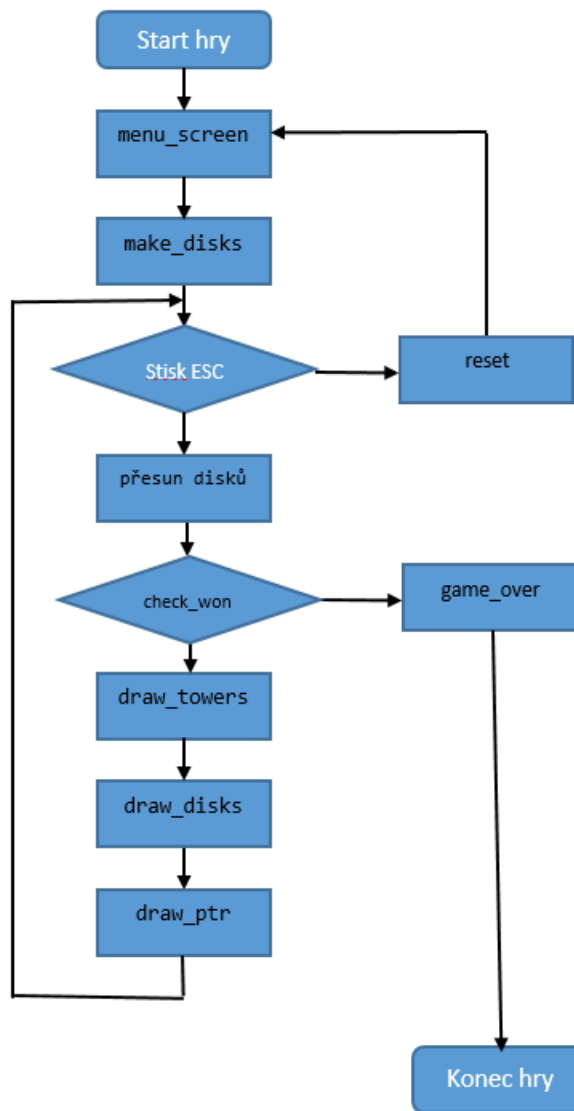
```
# hlavní herní smyčka (opakuje se tak dlouho, dokud je v globální proměnné game_done hodnota false)
```

```
while not game_done:
```

Podrobný popis jednotlivých částí kódu, je uveden v komentářích přímo v kódu hry.

## Analýza běhu programu

Pomocí vývojového diagramu níže naznačuji běh hry, který odpovídá průběhu hry, který je popsán v úvodu.



Hra je přehledně naprogramována a umožňuje jednoduše změnit klíčové vlastnosti hry.

Například zvětšit maximální použitelný počet disků z 6 na 8:

```
if event.key in [pygame.K_RIGHT, pygame.K_UP]:    #stisk šipky nahoru
    n_disks += 1
    if n_disks > 8:
        n_disks = 8
```

Upravit barevné schéma hry a tvar disků.

Hra „Hanojské věže“ může žákům usnadnit vstup do řešení exponenciálních složitostí.

Kdy nalézt řešení např. pro 2 až 4 disky lze zvládnout ve velice krátkém čase, ale při použití např. 6 disků, už řešení není tak jednoduché.