7. Metoda Monte Carlo

Zadání:

Metoda Monte Carlo představuje rodinu metod a filosofický přístup k modelování jevů, který využívá vzorkování prostoru (například prostor čísel na herní kostce, které mohou padnout) pomocí pseudonáhodného generátoru čísel. Jelikož se jedná spíše o filosofii řešení problému, tak využití je téměř neomezené. Na hodinách jste viděli několik aplikací (optimalizace portfolia aktiv, řešení Monty Hall problému, integrace funkce, aj.). Nalezněte nějaký zajímavý problém, který nebyl na hodině řešen, a získejte o jeho řešení informace pomocí metody Monte Carlo. Můžete využít kódy ze sešitu z hodin, ale kontext úlohy se musí lišit.

Řešení:

Mým úkolem bylo najít zajímavý problém a analyzovat jej pomocí metody Monte Carlo. Pro následující zadání jsem si zvolil problém hry Kostky, pro které analyzuji % poměr vítězství jednotlivých hráčů, jejich průměrné body za počet odehraných her.

První úkol, který mě čekal po předchozím pokusu o analýzu, bylo vytvořit si třídu, která nám bude vytvářet jednotlivé hráče. V této třídě si hráče rozlišíme a vytvoříme seznamy pro ukládání hodnot pro nadcházející analýzu.

```
# Třída hráče
2 usages
class Hrac:
    def __init__(self, jmeno):
        self.jmeno = jmeno
        self.kolo_body = []
        self.body = 0
        self.vitezstvi = []

    def __repr__(self):
        return f"Jsem tu abych pomohl se snazší implementací vyhodnocení oproti původní verzi MC"
```

Následně jsem si vytvořil funkci ”hra()“, která jako argument přijímá list se soutěžícími a počet her.

```
# Spuštění hry
1 usage
def hra(soutezici, n_her):
    # print(f"Je na tahu hráč {soutezici.jmeno}")
    for i in range(n_her):
        for hrac in soutezici:
            herni_kolo = najdi_stejne(hod_kostky(pocet_kostek))
            hrac.body += herni_kolo
            hrac.kolo_body.append(herni_kolo)
    return
```

V této funkci se zavolá funkce na hození kostky a s hodnotami kostky se zavolá funkce “najdi_stejne()”, která jako argument přijímá list s hodnotami z kostky.

```python
# Funkce pro hledání stejných čísel v hozených kostkách
1 usage
def najdi_stejne(kostka):
    seznam_hodnot = kostka
    stejne_hodnoty = []
    i = 0
    tuple_k_bodovani = []

    while i < len(seznam_hodnot):
        value = seznam_hodnot[i]
        count = seznam_hodnot.count(value)

        if count >= 2:
            stejne_hodnoty.extend([value] * count)
            for j in range(count):
                seznam_hodnot.remove(value)
        else:
            i += 1

    # print(stejne_hodnoty)
    for value in set(stejne_hodnoty):
        count = stejne_hodnoty.count(value)
        tuple_k_bodovani.append((value, count))

    # print(tuple_k_bodovani)
    return bodovani(tuple_k_bodovani, seznam_hodnot)

```

V této funkci si procházím “seznam_hodnot”, ve kterém jsou nahrané hody kostky a hledám ty hodnoty seznamu, které mají počet větši než 2 a přidám si je do listu pomocí metody “extend()”, která mi umožní přidat více hodnot najednou. Při použití metody “append()” jsem narazil na chybu, kdy mi vždy chyběl první člen z nalezeného páru, nakonec odstraním nalezenou hodnotu ze seznamu. Poté si uspořádáme seznam pomocí metody “set()” a projdeme jednotlivé hodnoty s jejich výskytem, které poté přidáme do ntice “tuple_k_bodovani”, kterou budeme spolu se seznamem hodnot bodovat.

Bodování:

```python
# Funkce pro bodování jednotlivých hodů
1 usage
def bodovani(stejna_cisla, hodnoty_hodu):
    """
    body = 0
    global postupka
    global trojice
    global ctverice
    global petice
    global sestice
    global tri_pary

    for value, count in stejna_cisla:
        if value != 1:
            if count == 3:
                body += value * 100
                trojice += 1
            elif count == 4:
                body += value * 200
                ctverice += 1
            elif count == 5:
                body += value * 400
                petice += 1
            elif count == 6:
                body += value * 800
                sestice += 1
        if value == 1:
            if count == 3:
                body += 1000
                trojice += 1
            elif count == 4:
                body += 2000
                ctverice += 1
            elif count == 5:
                body += 4000
                petice += 1
            elif count == 6:
                body += 8000
                sestice += 1

    if len(stejna_cisla) == 3 and all(count == 2 for value, count in stejna_cisla):
        body += 1000

    if set(hodnoty_hodu) == {1, 2, 3, 4, 5, 6}:
        body += 1500
        bodovani_list.append(body)
        postupka += 1
#       print(f"Body: {body}")
        return body

    if any(value in [1] and 1 <= hodnoty_hodu.count(value) <= 2 for value in hodnoty_hodu):
        body += hodnoty_hodu.count(1) * 100

    if any(value in [5] and 1 <= hodnoty_hodu.count(value) <= 2 for value in hodnoty_hodu):
        body += hodnoty_hodu.count(5) * 50

    bodovani_list.append(body)
#    print(f"Body: {body}")
    return body
```

Z funkce “bodovani()” kromě počtu bodů jednotlivého hráče, sbírám také data o hozených kombinacích pro výpočet % hozených kombinací v porovnání s celkovým počtem hodů.

Jakmile se odehrají všechny hry, následuje vyhodnocení poměru výher jednotlivých hráčů a poměru remíz.

```python
# Vyhodnocení pro grafy
1 usage
def vyhodnoceni(n_remiz):
    i = 0
    for j in range(pocet_her):
        if hrac1.kolo_body[i] > hrac2.kolo_body[i]:
            hrac1.vitezstvi.append(1)
            hrac2.vitezstvi.append(0)
            n_remiz.append(0)
            i += 1
        elif hrac1.kolo_body[i] == hrac2.kolo_body[i]:
            n_remiz.append(1)
            hrac1.vitezstvi.append(0)
            hrac2.vitezstvi.append(0)
            i += 1
        else:
            hrac1.vitezstvi.append(0)
            hrac2.vitezstvi.append(1)
            n_remiz.append(0)
            i += 1
    return
```

Závěr:

Na závěr jsem dělal analýzu % jednotlivých kombinací při různém počtu hodů:

```
***** Počet hodů kostkami: 200 *****
% pravděpodobnost hození postupky bylo 2.5%, celkově padla na kostkách 5krát.
% pravděpodobnost hození tří párů bylo 0.0%, celkově padla na kostkách 0krát.
% pravděpodobnost hození trojice bylo 32.0%, celkově padla na kostkách 64krát.
% pravděpodobnost hození čtveřice bylo 6.5%, celkově padla na kostkách 13krát.
% pravděpodobnost hození pětice bylo 0.0%, celkově padla na kostkách 0krát.
% pravděpodobnost hození šestice bylo 0.0%, celkově padla na kostkách 0krát
```

```
***** Počet hodů kostkami: 1000 *****
% pravděpodobnost hození postupky bylo 1.6%, celkově padla na kostkách 16krát.
% pravděpodobnost hození tří párů bylo 0.0%, celkově padla na kostkách 0krát.
% pravděpodobnost hození trojice bylo 30.599999999999998%, celkově padla na kostkách 306krát.
% pravděpodobnost hození čtveřice bylo 5.600000000000005%, celkově padla na kostkách 56krát.
% pravděpodobnost hození pětice bylo 0.3%, celkově padla na kostkách 3krát.
% pravděpodobnost hození šestice bylo 0.0%, celkově padla na kostkách 0krát
```

```
***** Počet hodů kostkami: 3000 *****
% pravděpodobnost hození postupky bylo 1.3%, celkově padla na kostkách 39krát.
% pravděpodobnost hození tří párů bylo 0.0%, celkově padla na kostkách 0krát.
% pravděpodobnost hození trojice bylo 31.733333333333334%, celkově padla na kostkách 952krát.
% pravděpodobnost hození čtveřice bylo 4.0%, celkově padla na kostkách 120krát.
% pravděpodobnost hození pětice bylo 0.16666666666666669%, celkově padla na kostkách 5krát.
% pravděpodobnost hození šestice bylo 0.0%, celkově padla na kostkách 0krát
```

```
***** Počet hodů kostkami: 10000 *****
% pravděpodobnost hození postupky bylo 1.68%, celkově padla na kostkách 168krát.
% pravděpodobnost hození tří párů bylo 0.0%, celkově padla na kostkách 0krát.
% pravděpodobnost hození trojice bylo 31.16%, celkově padla na kostkách 3116krát.
% pravděpodobnost hození čtveřice bylo 5.0200000000000005%, celkově padla na kostkách 502krát.
% pravděpodobnost hození pětice bylo 0.38999999999999996%, celkově padla na kostkách 39krát.
% pravděpodobnost hození šestice bylo 0.02%, celkově padla na kostkách 2krát
```

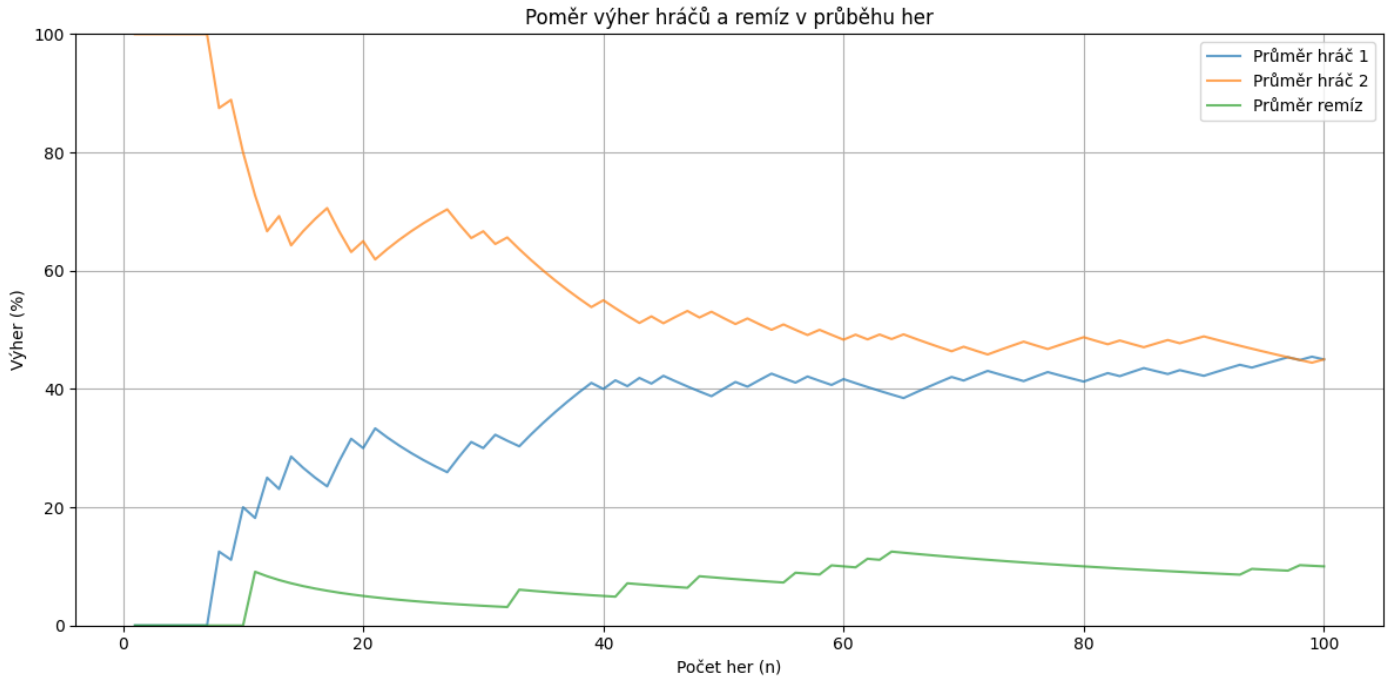
```
***** Počet hodů kostkami: 20000 *****
% pravděpodobnost hození postupky bylo 1.6%, celkově padla na kostkách 320krát.
% pravděpodobnost hození tří párů bylo 0.0%, celkově padla na kostkách 0krát.
% pravděpodobnost hození trojice bylo 31.86%, celkově padla na kostkách 6372krát.
% pravděpodobnost hození čtveřice bylo 4.99%, celkově padla na kostkách 998krát.
% pravděpodobnost hození pětice bylo 0.36%, celkově padla na kostkách 72krát.
% pravděpodobnost hození šestice bylo 0.01%, celkově padla na kostkách 2krát
```

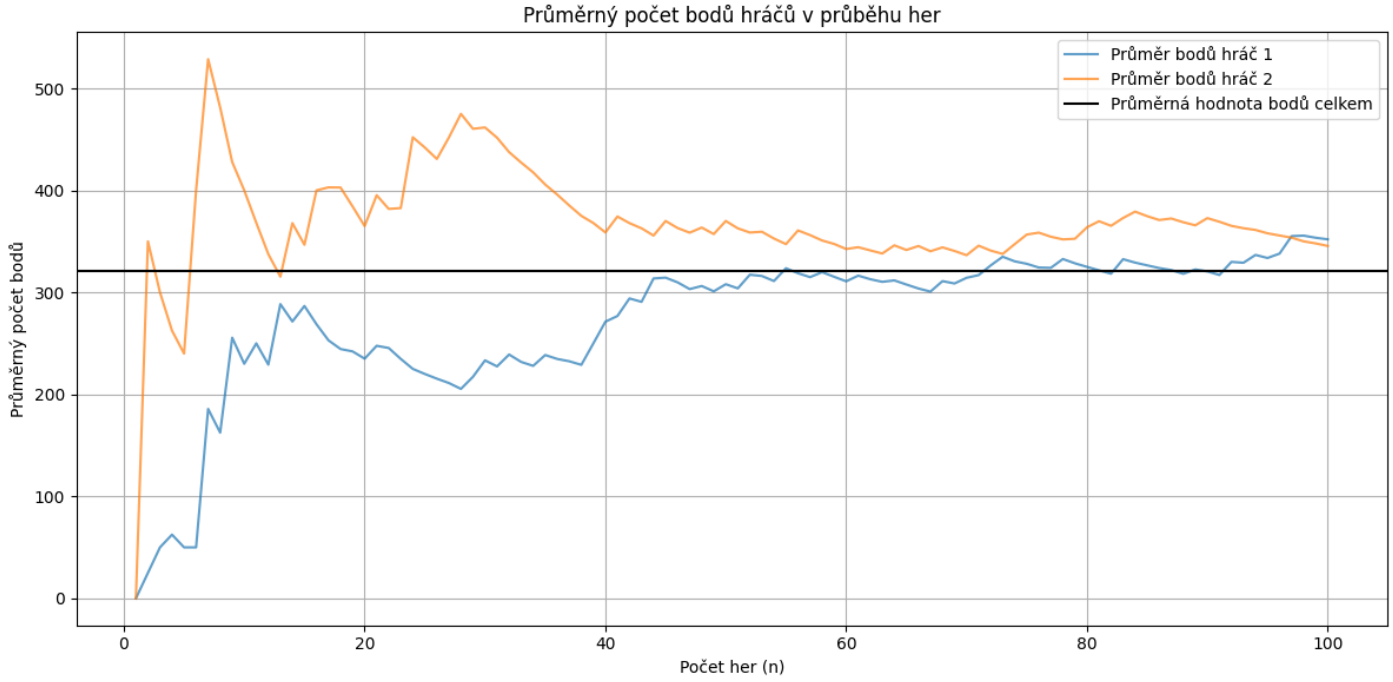
Z výsledků vyplývá, že % na hození většiny kombinací nejsou tolik ovlivněna počtem hodů kostky a jsou předpověditelná. Poté máme pravděpodobnost na hození pětice, šestice a tří párů, které vypadají opravdu jako prvek “náhody”.

Grafy:

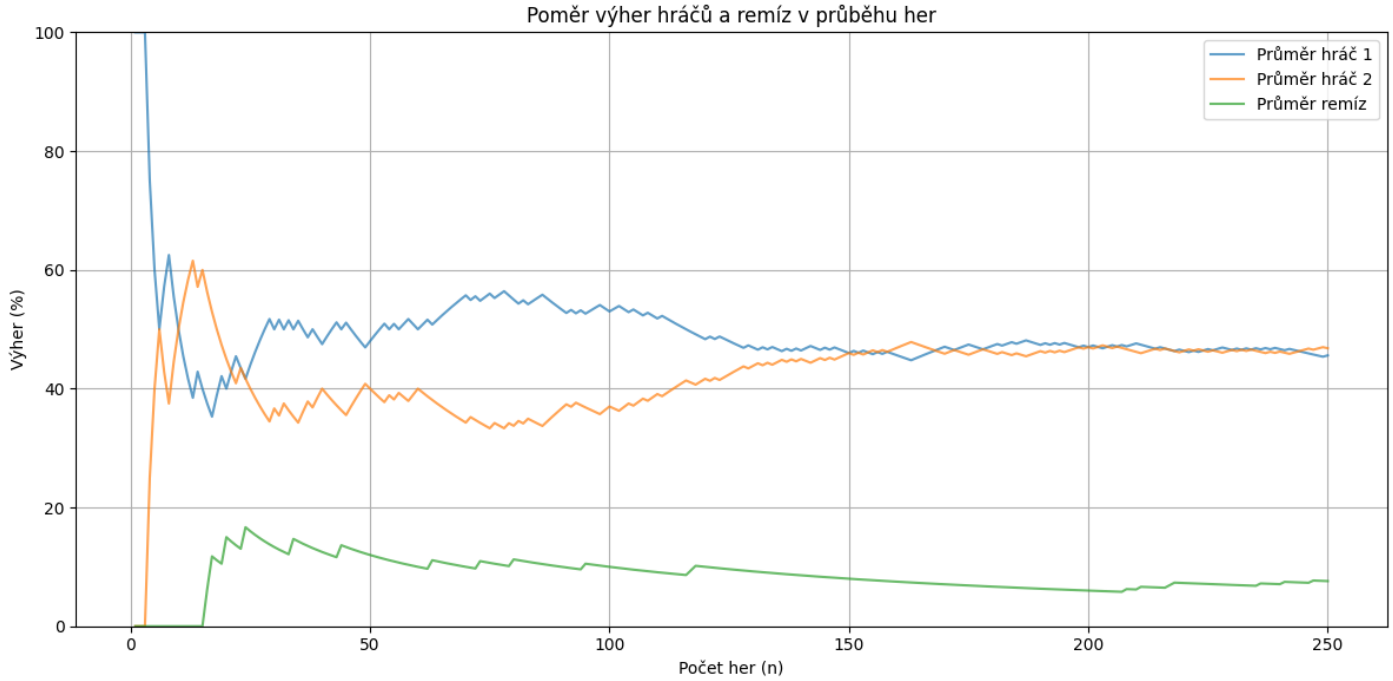
Z grafů můžeme vyčíst, že nejvíce vyhrát/prohrát můžeme hlavně v prvních kolech. Čím více kol odehrajeme, tím více k sobě hodnoty konvergují.

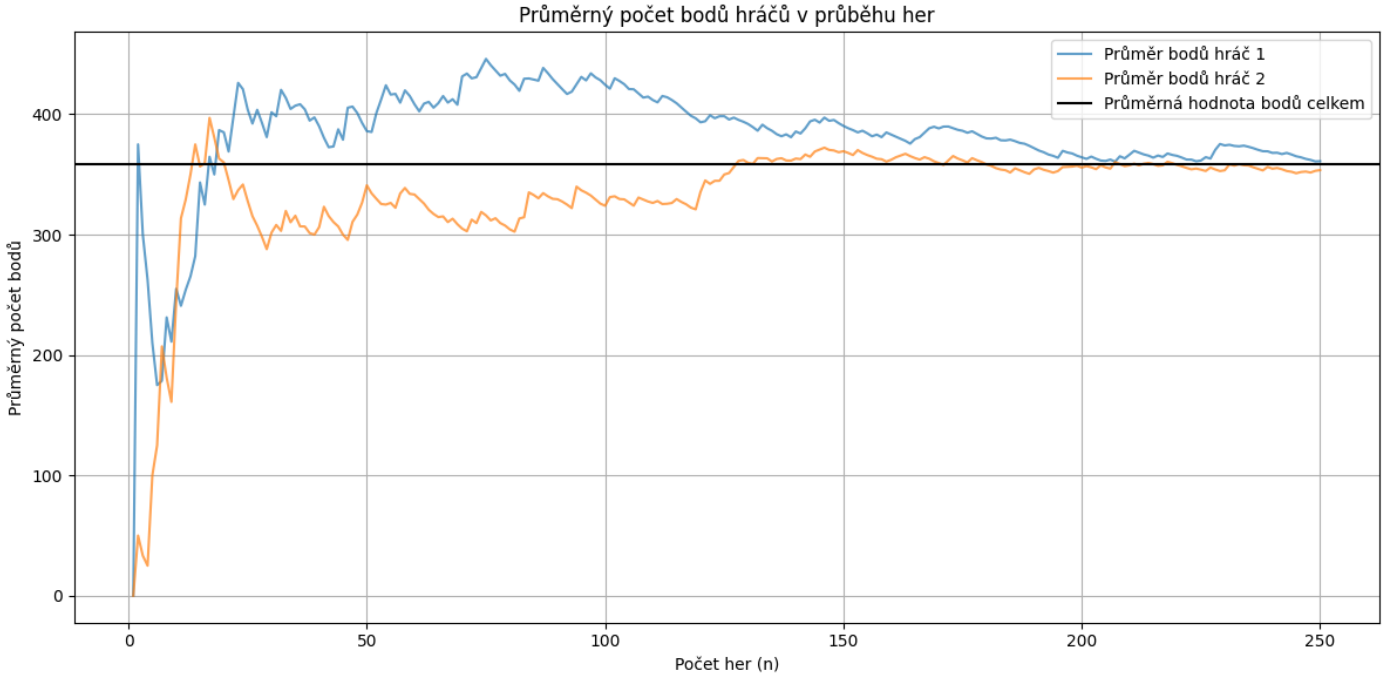
Pro 100 her:



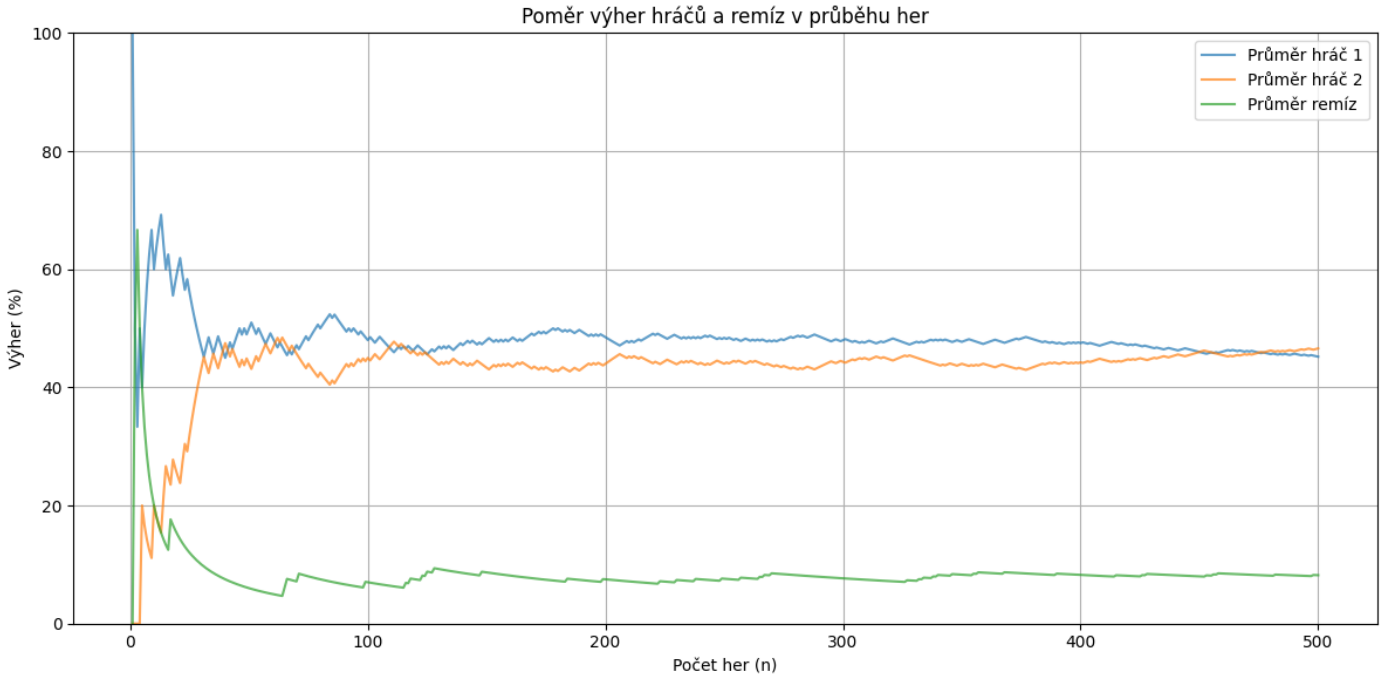

Průměrný počet bodů hráčů v průběhu her
Průměr bodů hráč 1
Průměr bodů hráč 2
Průměrná hodnota bodů celkem
Průměrný počet bodů
Počet her (n)
0
100
200
300
400
500
0
20
40
60
80
100

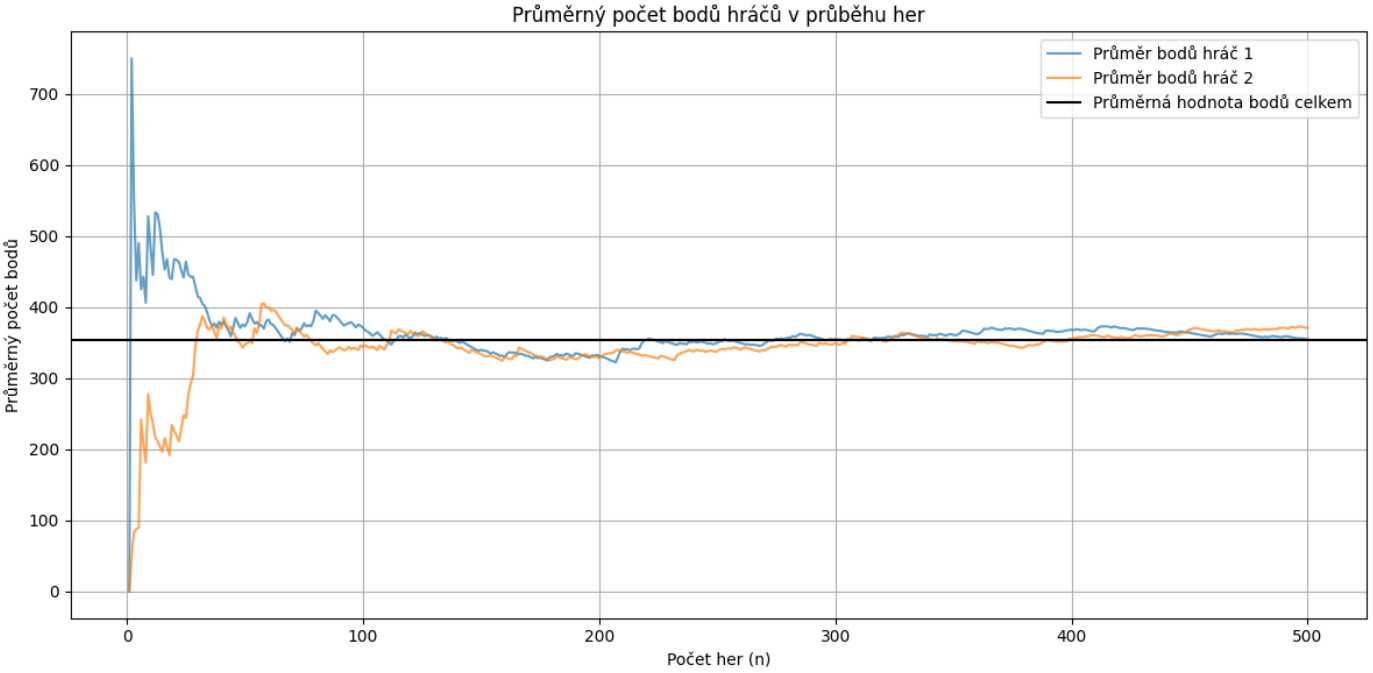
Pro 250 her:


Poměr výher hráčů a remíz v průběhu her
Průměr hráč 1
Průměr hráč 2
Průměr remíz
Výher (%)
Počet her (n)
0
20
40
60
80
100
0
50
100
150
200
250

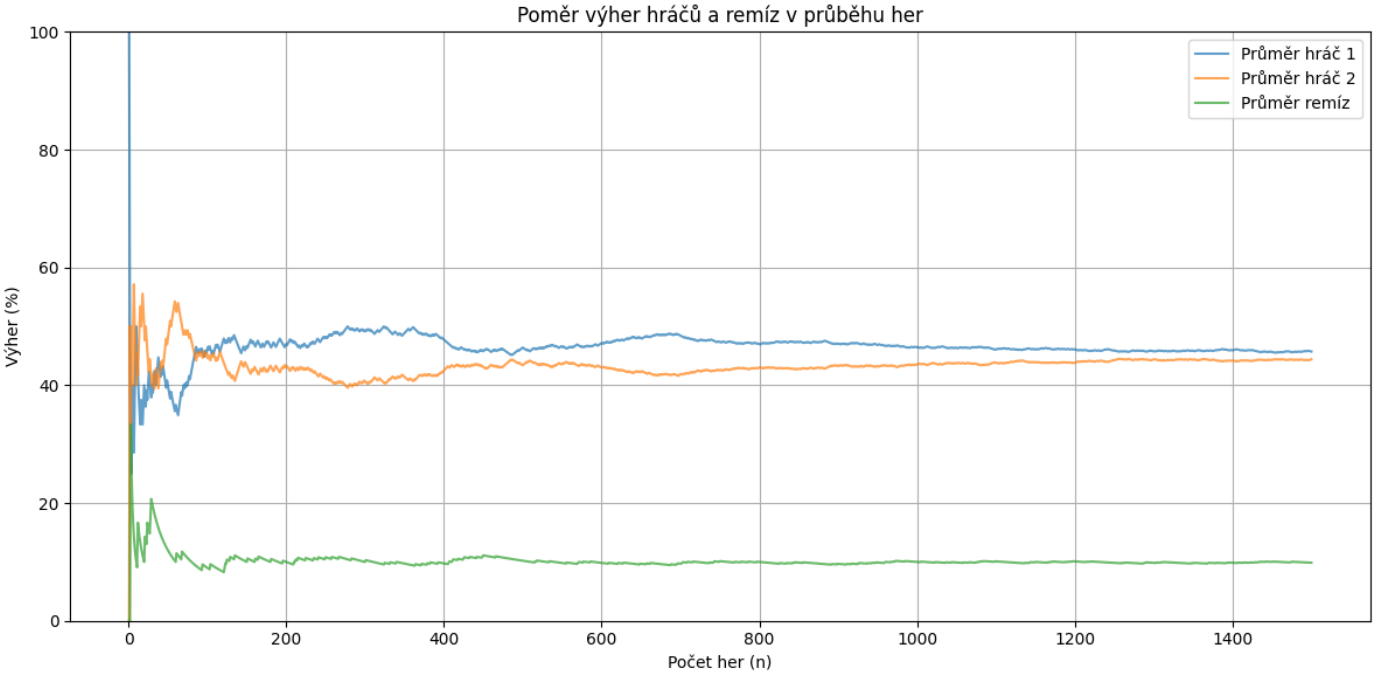

Průměrný počet bodů hráčů v průběhu her
Průměr bodů hráč 1
Průměr bodů hráč 2
Průměrná hodnota bodů celkem
Průměrný počet bodů
Počet her (n)
0
100
200
300
400
0
50
100
150
200
250

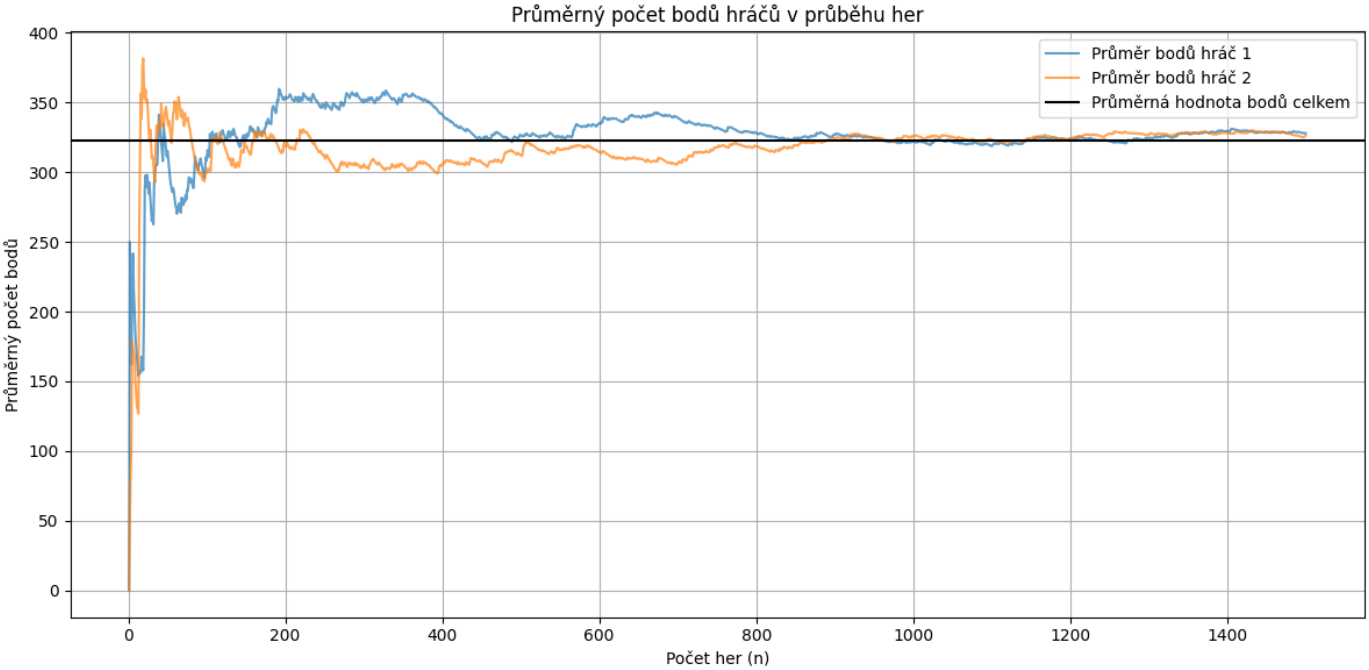
Pro 500 her:


Poměr výher hráčů a remíz v průběhu her
Průměr hráč 1
Průměr hráč 2
Průměr remíz
Výher (%)
Počet her (n)
0
20
40
60
80
100
0
100
200
300
400
500


Průměrný počet bodů hráčů v průběhu her
Průměr bodů hráč 1
Průměr bodů hráč 2
Průměrná hodnota bodů celkem
Průměrný počet bodů
Počet her (n)
0
100
200
300
400
500
600
700

Pro 1500 her:


Poměr výher hráčů a remíz v průběhu her
Průměr hráč 1
Průměr hráč 2
Průměr remíz
Výher (%)
Počet her (n)
0
20
40
60
80
100
200
400
600
800
1000
1200
1400


Průměrný počet bodů hráčů v průběhu her
Průměr bodů hráč 1
Průměr bodů hráč 2
Průměrná hodnota bodů celkem
Průměrný počet bodů
Počet her (n)
0
50
100
150
200
250
300
350
400
0
200
400
600
800
1000
1200
1400