

---

Kutinfo

---

---

Kő-papír-olló...

---

Lukovics Márk, Marx Pál Fülöp, Pórfy Barnabás



## Kivonat

Egyszerű kő-papír-olló játék C++-ban írva, konzolos bevitellel.

# 1. Bevezetés

## 1.1. Mi ez?

A kő-papír-olló játékot két ember játszhatja a kezével.

A játékosok hangosan háromig számolnak vagy a játék nevét, a "kő – papír – olló"-t mondják, minden számolásnál meglendítve a felemelt és ökölbe szorított kezüket. A harmadik számolás után vagy az utána következő ütemben a játékosok kezükkel felveszik a három alakzat egyikét és megmutatják az ellenfelüknek.

A másik változat a csoportkör, amit 3-6 játékos játszhat, ennek során mindenki megszámolja, hány más játékoson aratott győzelmet az adott körben (mindenki egyszerre mutat). Győzelmeit egy-egy pontként jegyezzük, aki előbb szerez minimum 20 pontot (változó, de ennyi az ajánlott), az a nyertes.

- kő: a zárt ököl
- olló: a kinyújtott, szétnyitott mutató és középső ujj
- papír: a nyitott tenyér



A játék a legtöbb országban kő, papír, olló-ként népszerű, noha több változat létezik még (pl.: nyúl, puska, répa), és Japánban dzsankennek nevezik, ami tulajdonképpen a kígyó, béka, csiga játék.

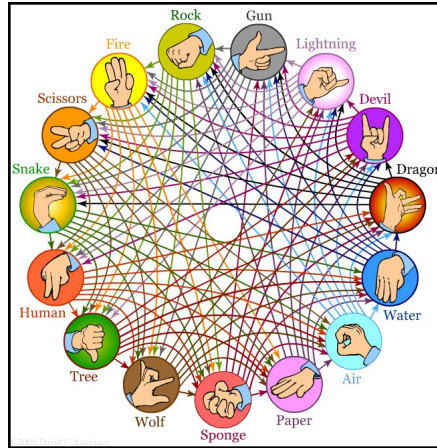
Egyes sportokban ezzel a játékkal határozzák meg, melyik csapat kezdi a meccset. Eső esetén néha így döntik el, megtartsák-e a játszmát. Élő szerepjátékokban is használják véletlenek létrehozására, mivel semmiféle felszerelés nem szükséges hozzá. Néha fogadást is kötnek rá, a szerencsejátékokhoz hasonlóan.

## 1.2. Valószínűségek

A kő-papír ollót néha a pénzfeldobáshoz, a kockadobáshoz vagy a pálcátöréshez hasonlóan egy személy kiválasztásban is használják. Elméletileg mindenkinek ugyanannyi esélye van a győzelemre, de a véletlen választásoktól eltérően azonban itt lehet fejleszteni a játéktechnikát ha több kört játszanak, mert a tapasztalt játékos kiismerheti ellenfele nem-véletlenszerű taktikáját.

### 1.3. Miért éppen ez?

A választás azért esett erre a játékra, mert ismertünk egy továbbfejlesztett verziót a játékról, amit le akartunk programozni. Ebben a verzióban 15 különböző lehetőség van.



1. ábra. Kiterjesztett kő-papír-olló

## 2. Tervezet

### 2.1. Hogyan működjön? (Első verzió)

Az első ötletünk az volt, hogyha ez a játék kiterjeszthető 15-re, akkor ebből következik, hogy az összes egynél nagyobb pozitív páratlan számra használható.

Ezért mi egy olyan játékot akartunk megvalósítani, aminek az elején a játékos kiválaszthatja, hány elemű játékkal akar játszani.

## 2.2. Megvalósítás

A megvalósításhoz C++ nyelvet használtuk.

```
#include <iostream>
#include <cstdlib>
#include <limits>
#include <ctime>
#include <vector>

using namespace std;

int main()
{
    int N = 0;
    bool ans = false;

    cout << "Give me the # of N options: ";
    while(!ans)
    {
        cin >> N;
        if(N%2 == 0)
        {
            cerr << "PLEASE ENTER A VALID NUMBER WHICH IS ODD" << endl;
            ans = false;
        }
        else
        {
            ans = true;
        }
    }

    vector< vector<int> > RPS(N);
    for (int iusr = 0; iusr < N; iusr++)
    {
        for (int ielem = iusr; ielem < ((N-1)/2+iusr); ielem++)
        {
            RPS[iusr].push_back((ielem+1)%N);
            cout << iusr << ":" << (ielem+1)%N << endl;
        }
    }

    return 0;
}
```

2. ábra.  $2n+1$ -es kő-papír-olló

Amikor itt tartottunk, akkor rájöttünk, hogy ezen az úton elindulva a játék megvalósítása túl sok időt venne igénybe. Emiatt úgy döntöttünk, hogy egy másik irányba terjesztjük ki a játékot. Nevezetesen a többkörös játéokra.

## 2.3. Hogyan működjön? (Második verzió)

A tervünk az volt, hogy az eredeti három lehetséges kő-papír-olló játékot akárhányszor lehessen játszani egymás után. Ehhez előlről kezdtük a kódot. Ebben a verzióban számolja az eredményeket, viszont itt a lehetőségek kis száma miatt, megtehattük, hogy nevezéktant(rk, pr, sc) használunk. Ami még különbség, hogy itt, szintén a korlátozottság miatt, nem volt szükség arra, hogy a program magának generálja a "kiütési párokat", hanem előre beleírhattuk a kódba.

### 3. Kód

A fent leírt tervezetet az alábbi módon valósítottuk meg:

```
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <string>

using namespace std;

int rps(int a,int b)
{
    /*
        rock = 0
        paper = 1
        scissors = 2
    */
    if(a==b)
        return 0;
    if((a==1 && b==0) || (a==2 && b==1) || (a==0 && b==2))
        return 1; //user wins
    else
        return 2; //computer wins
}

int main()
{
    int len;
    int user;
    int draw = 0;
    int player = 0;
    int computer = 0;
    string rk = "Rock";
    string pr = "Paper";
    string sc = "Scissors";

    string us;
    string co;

    cout << "Best_out_of?" << endl;
    cin >> len;
    if(len < 3)
        len = 3;
    if(len % 2 == 0)
        len = len+1;
    cout << "Best_out_of:_ "<< len << endl;
    int len2 = len;
    int comp;
```

```

srand (time(NULL));

for(int i = 0; i < len; i++)
{
    comp = rand() % 3;

    cout << "Rock(0) , Paper(1) or Scissors(2)?" << endl;
    cin >> user;
    while(user < 0 || user > 2)
    {
        cout << "Choose 0, 1 or 2" << endl;
        cin >> user;
    }
    if(comp == 0)
    {
        co = rk;
    }
    if(comp == 1)
    {
        co = pr;
    }
    if(comp == 2)
    {
        co = sc;
    }
    if(user == 0)
    {
        us = rk;
    }
    if(user == 1)
    {
        us = pr;
    }
    if(user == 2)
    {
        us = sc;
    }

    cout << "Player: " << us << endl << "Computer: " << co << endl;

    if(rps(user, comp) == 0)
    {
        cout << "Draw!" << endl;
        len++;
        draw++;
    }
    if(rps(user, comp) == 1)

```

```

    {
        cout << "Player_wins_round!" << endl;
        player++;
    }
    if(rps(user, comp) == 2)
    {
        cout << "Computer_wins_round!" << endl;
        computer++;
    }
    cout << "_" << endl << "Score:_" << endl << "Player:_"
    << player << endl << "Computer:_" << computer << endl
    << "_" << endl;
    if(computer == len2/2 +1)
    {
        cout << "——Computer_wins!——" << endl;
        cout << "_" << endl << "End_score:_" << endl << "Player:_"
        << player << endl << "Computer:_" << computer << endl
        << "_" << endl;
        cout << "——Better_luck_next_time!——";
        return 0;
    }
    if(player == len2/2 +1)
    {
        cout << "——Player_wins!——" << endl;
        cout << "_" << endl << "End_score:_" << endl;
    }
}
return 0;
}

```

## 4. Ellenőrzés

Ez a kód a terveinket megvalósítja. Az viszont egyértelműen nem valósul meg, hogy a gép kitalálja, hogy mit fogunk választani. Tehát ha a gép ellen játszunk, akkor tényleg egyenlő esélyeink van. Ellenben azzal, mint ahogy emberek ellen játszunk.

## 5. Diskusszió

Amint azt a bevezetésben írtuk, ezt a játékot több irányba tovább lehet fejleszteni. Az első változat egy érdekesebb megközelítése a játéknak, viszont a megvalósítása nagyobb időigénnyel rendelkezik. A második verzió az egyik egyszerűbb változat, viszont könnyebben megvalósítható.

## 6. Felhasznált források

- <https://hu.wikipedia.org/wiki/Kő-pap%C3%ADr-olló>
- <https://tex.stackexchange.com/>
- <http://www.umop.com/images/rps15.jpg>