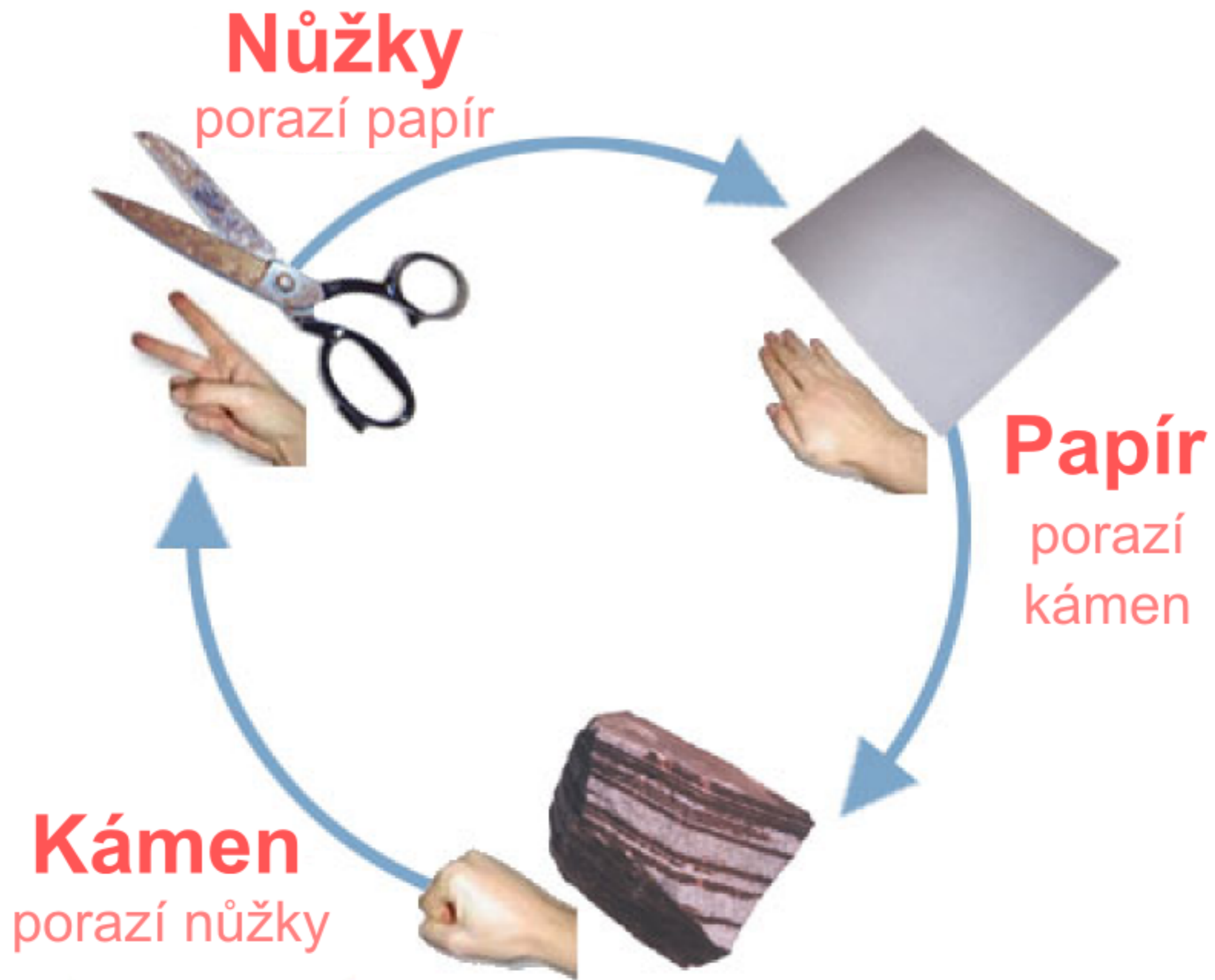


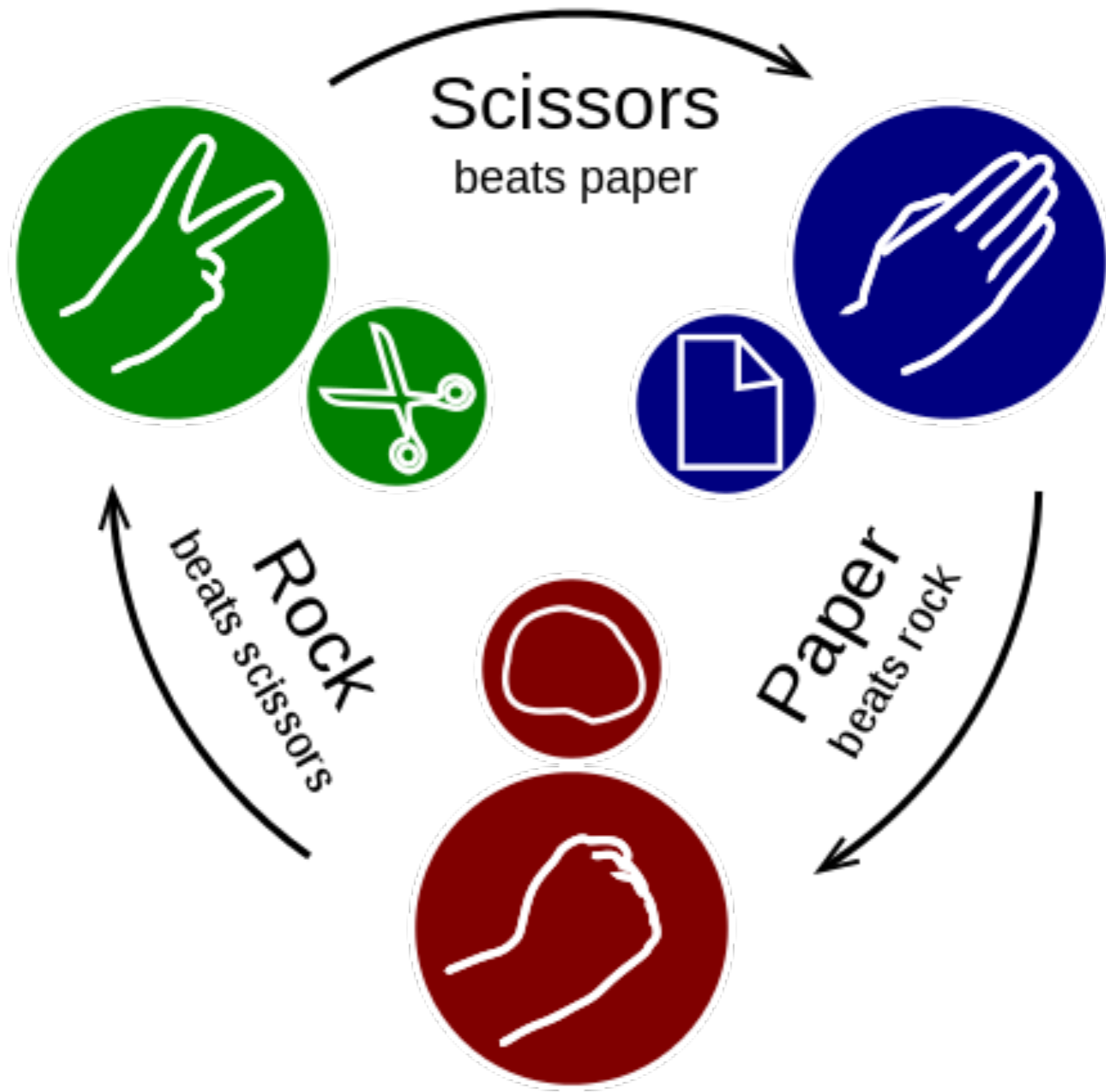
Kámen-nůžky-papír

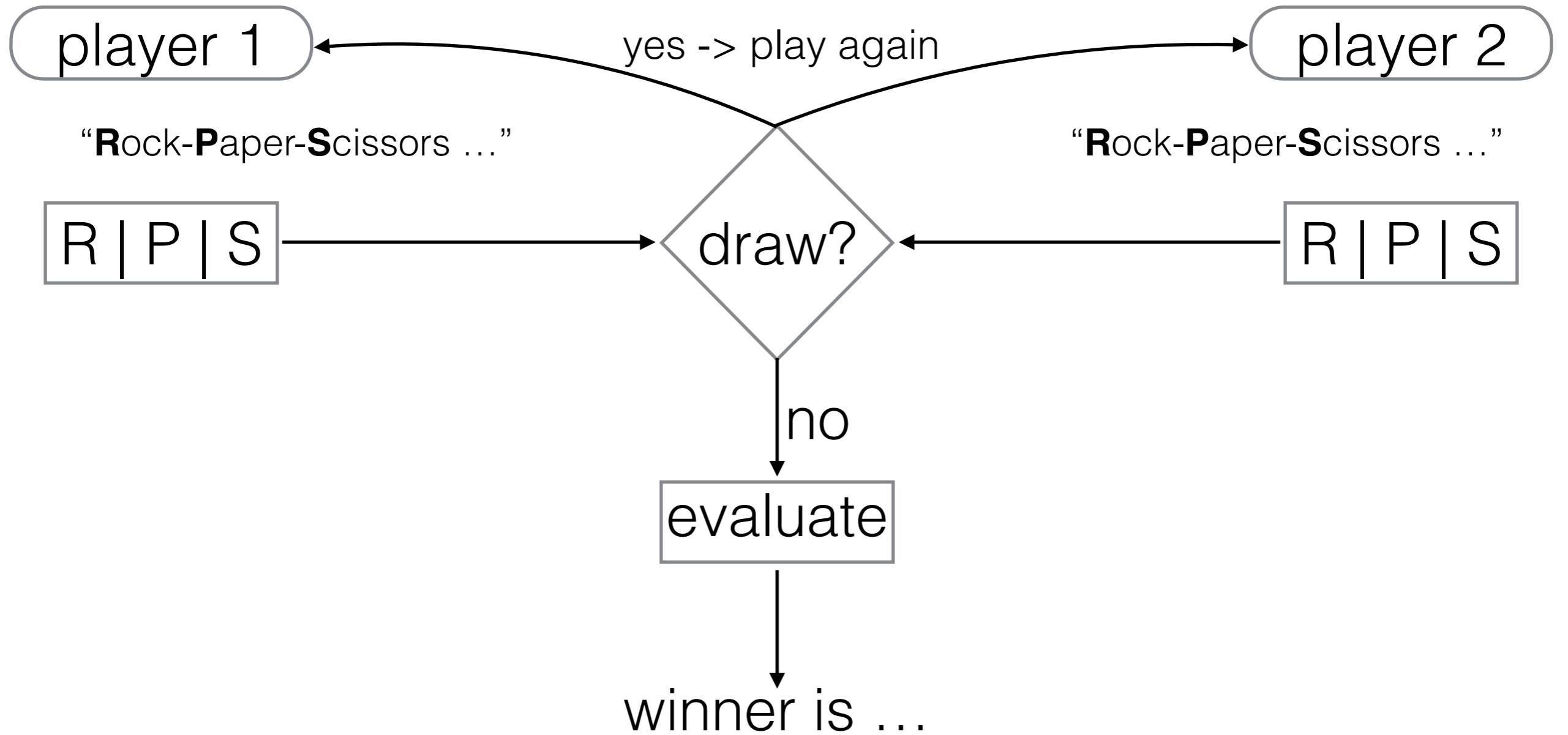
Tomáš Svoboda

Centrum strojového vnímání, Katedra kybernetiky
Fakulta elektrotechnická, České vysoké učení technické

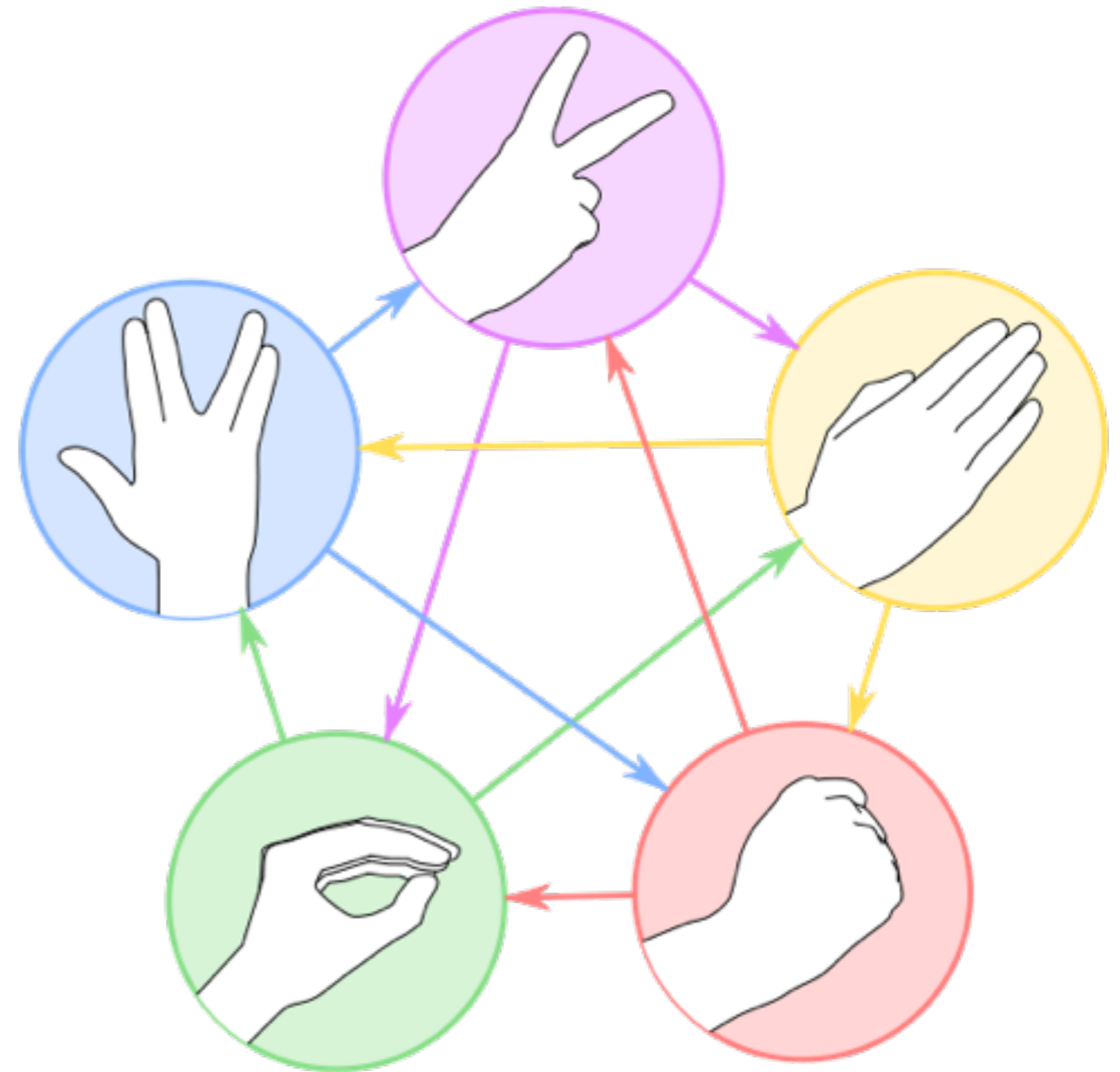
Studijní program: [Otevřená informatika](#)







Vyšší level



Player

player 1

play

R | P | S

Game

run

yes -> play again

draw?

no

evaluate

winner is ...

Player

player 2

play

R | P | S

playerdummy.py

obecná definice hráče

```
1 class MyPlayer:
2     '''A very dummy player (always returns R)'''
3
4     def play(self):
5         return 'R'
6
7 if __name__ == "__main__":
8     p = MyPlayer() # creating a player
9     print(p.play()) # showing what it plays
```

docstring - popis hráče/třídy

vždy vrat' 'R'

vykonej, pokud je spuštěno jako hlavní program

vytvoření konkrétního hráče podle vzoru

konkrétní hráč hraje

Co když potřebujeme chování hráče měnit?

playerdummys.py

```
1 class MyPlayer:
2     '''A dummy player on steroids'''
3     def __init__(self, answer='R'):
4         self.answer = answer
5
6     def play(self):
7         return self.answer
8
9 if __name__ == "__main__":
10     p1 = MyPlayer() # creating a default player
11     print(p1.play()) # showing what it plays
12     p2 = MyPlayer('P') # a better player?
13     print(p2.play()) # showing what it plays
14     # oops changed mind
15     p1.answer = 'S'
16     print(p1.play())
17
```

konstruktor objektu

přiřazení atributu objektu

konkrétní hráč bude hrát tak, jak má předepsáno

vlastnosti objektu mohou měnit

podruhé už hraje jinak

Player

player 1

play

R | P | S

Game

run

yes -> play again

draw?

no

evaluate

winner is ...

Player

player 2

play

R | P | S

Jak hrát hru?

```
p1 = Player
```

```
p2 = Player
```

```
draw = True
```

```
while draw:
```

```
    move1 = p1.play
```

```
    move2 = p2.play
```

```
    draw = (move1 == move2)
```

```
result = evaluate(move1, move2)
```

Třída Game

```
1 class Game:
```

```
2     def __init__(self, p1, p2):
```

```
3         self.p1 = p1
```

Předání hráčů do hry

```
4         self.p2 = p2
```

```
5         self.winner = None
```

Na začátku žádný vítěz není

```
7     def run(self):
```

```
8         draw = True
```

Dokud nehrají, tak je remíza

```
9         while draw:
```

a dokud je remíza, tak hrajte dál

```
10             move1 = self.p1.play()
```

```
11             move2 = self.p2.play()
```

```
12             draw = (move1 == move2)
```

```
13             result = evaluate_moves([move1, move2])
```

```
14             if result[0] > result[1]:
```

```
15                 self.winner = p1
```

Porovnej tahy podle pravidel
a přiřad' vítěze

```
16         else:
```

```
17             self.winner = p2
```

porovnej tahy

```
1 def evaluate_moves(moves):
2     '''
3     compares moves (plays) and decides about the winner
4     :param moves: 1x2 list of valid moves
5     :return: 1x2 list with points [1,0] or [0,1]
6     depending on who is winner
7     '''
8     if moves in [['P', 'R'], ['S', 'P'], ['R', 'S']]:
9         return [1, 0]
10    else:
11        return [0, 1]
12
```

Paper > Rock, Scissors > Paper,
Rock > Scissors

hlavní program

```
1 import playertom
2 import playerdummy
3 tady patří class Game: a spol. ...
4 if __name__ == "__main__":
5     p1 = playertom.MyPlayer() vytvoř hráče
6     p2 = playerdummy.MyPlayer()
7     g = Game(p1, p2) inicializuj hru, předej hráče
8     g.run() hraj
9     print('Winner is:', g.winner.__doc__)
10         oznam vítěze
```

importuj hráče z modulů - souborů
playertom.py a playerdummy.py

vytvoř hráče

inicializuj hru, předej hráče

hraj

oznam vítěze

Pro pozdější analýzu výsledku, navrátíme index vítěze

```
1 class Game:
2     def __init__(self, p1, p2):
3         self.p1 = p1
4         self.p2 = p2
5         self.winner = None
6
7     def run(self):
8         draw = True
9         while draw:
10            move1 = self.p1.play()
11            move2 = self.p2.play()
12            draw = (move1 == move2)
13            result = evaluate_moves([move1, move2])
14            if result[0] > result[1]:
15                self.winner = p1
16                return 0
17            else:
18                self.winner = p2
19                return 1
```

Návratová hodnota podle toho,
kdo vyhrál

opakovaná hra

```
1 def compute_stats(winners):
2     wins = [0, 0] počet výher každého hráče
3     for winner in winners: pro každého vítěze v seznamu
4         wins[winner] = wins[winner] + 1
5     return wins
6
7 if __name__ == "__main__":
8     p1 = playertom.MyPlayer()
9     p2 = playerdummy.MyPlayer()
10    winners = [] inicializuj prázdný seznam
11    for i in range(10):
12        hraje 10x g = Game(p1, p2)
13        winners.append(g.run()) přidej výsledek na konec seznamu
14        print('Winner is:', g.winner.doc)
15        wins = compute_stats(winners) analyzuj celkové výsledky
16        print(p1.__doc__, 'won %d times' % wins[0])
17        print(p2.__doc__, 'won %d times' % wins[1])
```