

# Implementacija igre Jamb u Haskellu

Ime Prezime | Fakultet informatike u Puli | 2. rujna 2025.

## Uvod

Jamb se igra s pet kockica i (za implementaciju) pojednostavljenim listićem s 12 redaka (kategorija) u tri stupca: *Dolje*, *Gore*, *Slobodno*. Cilj je ostvariti najveći broj bodova kroz ograničeni broj poteza, uz do tri bacanja po potezu.

### Cilj projekta:

- funkcionalna implementacija Jamba u Haskellu,
- modularna arhitektura i čitljiv UI u terminalu,
- AI igrači (Random i Greedy) + *hints*,
- završni izračun i poredak igrača.

## Pravila (sažetak)

- Do tri bacanja po potezu (prvo bacanje svih 5 kockica, zatim *reroll* odabranih).
- Nakon poteza rezultat se mora upisati ili prekrižiti.
- Stupci:
  - Dolje**: redom od vrha prema dnu,
  - Gore**: redom od dna prema vrhu,
  - Slobodno**: bilo kojim redoslijedom.
- Bonus +30 ako zbroj 1–6 u stupcu  $\geq 60$ .
- Max=Min pravilo:  $(\text{Max} - \text{Min}) \times \text{Ones}$  se dodaje u taj stupac.

## Primjer bacanja & upisa

Bacanje	[6, 6, 6, 2, 5]
<i>Greedy</i> upis u <b>Sixes</b> (= 18) ili <b>Poker</b> (40+25 = 65)	

Odabir ovisi o popunjenosti listića i stupčanim ograničenjima (Down/Up/Free).

## Bodovanje (sažetak)

**Gornji dio (1–6)**: zbroj odgovarajućih kockica; bonus +30 ako zbroj  $\geq 60$  u stupcu.

**Donji dio**:

Kategorija	Bodovi
Max	zbroj svih kockica
Min	zbroj svih kockica
Straight (mala)	35 za [1, 2, 3, 4, 5]
Straight (velika)	45 za [2, 3, 4, 5, 6]
Full	30 + zbroj (npr. [3, 3, 3, 2, 2] $\Rightarrow$ 30+13 = 43)
Poker ( $\geq 4$ iste)	40 + zbroj
Yamb (5 istih)	50 + zbroj

**Stupčani rezultat**:

$$\text{Total}_{\text{col}} = (\text{Ones} - \text{Sixes} + \text{Bonus}) + (\textit{Max} - \textit{Min}) \times \textit{Ones} + \textit{Straight} + \textit{Full} + \textit{Poker} + \textit{Yamb}.$$

$$\text{Total} = \sum_{\text{col} \in \{\textit{Dolje}, \textit{Gore}, \textit{Slobodno}\}} \text{Total}_{\text{col}}.$$

### Primjer listića (isječak)

	Dolje	Gore	Slobodno
Ones	[ 1]	[ 0]	[ 2]
Twos	[ 2]	[ 2]	[ 4]
Threes	[ 0]	[ 9]	[ 0]
Fours	[ 4]	[ 8]	[12]
Fives	[10]	[ 0]	[10]
Sixes	[12]	[ 6]	[ 0]

### Primjer terminalskog ispisa

```
Na potezu: Greedy
AI potez: Roll
Kockice: [4,5,6,6,1]
Hint (top 3 legal):
  Sixes / Up = 12
  Poker / Free = 63
  Max / Down = 22
```

## AI strategije

**Random AI**: nasumično bira valjano polje ili reroll.

**Greedy AI**: bira potez s najvećim trenutnim dobitkom (po trenutnim kockicama), uz poštivanje ograničenja stupaca (Down/Up/Free).

**Hints**: za zadano stanje sugerira top-3 valjana upisa s procijenjenim bodovima.

*Daljnji rad*: heuristički AI (npr. čuvanje 4 istih za Yamb), procjena očekivanog dobitka (Monte Carlo), planiranje više poteza unaprijed.

## AI vs AI statistika (10 partija)

	Prosjek	Min	Max
Random	58	42	80
Greedy	145	120	165

### Zaključak

Projekt pokazuje da je Jamb moguće jasno i modularno implementirati u Haskellu:

- čisti tipovi i imutabilnost olakšavaju provođenje pravila,
- arhitektura po slojevima (UI, Engine, Score, AI),
- lako proširivo na naprednije AI strategije i grafički UI.

## Reference

- Haskell: <https://www.haskell.org/>
- Yahtzee: <https://en.wikipedia.org/wiki/Yahtzee>
- Lipovača, M.: *Learn You a Haskell* <https://learnyouahaskell.github.io/>
- Predavanja, FIPU: Funkcionalno programiranje
- Beamerposter: <https://ctan.org/pkg/beamerposter>