

# Implementacija igre Jamb u Haskellu

Marko Pekica  
Fakultet informatike u Puli

2. rujna 2025.

## 1 Uvod

Jamb je popularna društvena igra s kockicama, gdje se bodovi upisuju u listić pod različite kategorije. Cilj igre je ostvariti što veći broj bodova kroz 12 kategorija raspoređenih u četiri stupca (*dolje*, *gore*, *gore/dolje*, *najava*).

Projekt prikazuje implementaciju igre Jamb u programskom jeziku **Haskell**, s podrškom za:

- ručno igranje u terminalu,
- jednostavne AI protivnike (*random* i *greedy*),
- automatsko bodovanje i provjeru pravila,
- prikaz rezultata i poredak igrača.

## 2 Pravila igre

- Igra se s 5 kockica.
- Igrač ima najviše tri bacanja po potezu (prvo bacanje - sve kocke, druga dva - reroll kocki po želji, može sve, može samo neke)
- Nakon bacanja igrač mora upisati rezultat u jedno od slobodnih polja listića ili prekrižiti polje ako nema kocke za upis niti u jedno polje.
- Listić (pojednostavljena implementacija) ima tri stupca:
  - **Dolje** – polja se moraju puniti odozgo prema dolje.
  - **Gore** – polja se moraju puniti odozdo prema gore.
  - **DoljeGore - slobodni stupac** – polja se mogu puniti proizvoljno.
- Bodovanje:
  - Zbrajanje jedinica, dvojki, trojki, četvorki, petica, šestica.
  - **Max/Min**: razlika  $Max - Min$ , pomnožena s brojem jedinica u stupcu.
  - **Straight**: mala (1–5, 35 bodova), velika (2–6, 45 bodova).

- **Full house**:  $30 + \text{zbroj kockica}$ .
- **Poker**:  $40 + \text{zbroj kockica}$ .
- **Yamb (pet istih)**:  $50 + \text{zbroj kockica}$ .
- Ako je zbroj kategorija 1–6 u stupcu  $\geq 60$ , dodaje se **bonus od 30 bodova**.

### 3 Arhitektura programa

Program je podijeljen u module:

- `Types.hs` – osnovni tipovi podataka: `GameState`, `Player`, `ScoreCard`, `Category`, `Column`.
- `Engine.hs` – logika igre: bacanje kockica, promjena stanja, validacija poteza.
- `Score.hs` – pravila bodovanja, provjera ispravnosti upisa, izračun završnog rezultata.
- `Terminal.hs` – korisničko sučelje (unos poteza, ispis listića i kockica).
- `Strategies.hs` – AI protivnici (*randomAI*, *greedyAI*).
- `Hints.hs` – pomoćni savjeti s najboljim opcijama.
- `Main.hs` – glavna petlja igre.

### 4 AI protivnici

Implementirana su dva osnovna AI pristupa:

- **Random AI** – nasumično odabire poteze.
- **Greedy AI** – evaluira sve slobodne poteze i bira onaj koji trenutno donosi najviše bodova.

Eksperimentalno, greedy AI u svakom testu pobjeđuje random AI sa znatno većim rezultatom.

### 5 Rezultati

Primjer outputa iz igre:

```
-----
Na potezu: Rand
AI potez: Roll
Kockice: [5,1,2,5,4]
Hint (top 3 moves):
  Ones / Up = 1
  Yamb / Down = 0
```

-----  
Na potezu: Rand  
AI potez: Reroll [1,2,3]  
Kockice: [5,1,6,1,4]  
Hint (top 3 moves):  
    Ones / Up = 2  
    Yamb / Down = 0

-----  
Na potezu: Rand  
AI potez: Reroll [1,3,4]  
Kockice: [5,2,6,3,4]  
...

-----  
Na potezu: Rand  
AI potez: Write Ones / Up  
Upisano u Ones / Up

-----  
Na potezu: Greedy  
AI potez: Roll  
Kockice: [2,3,4,6,2]  
Hint (top 3 moves):  
    Twos / Up = 4

-----  
Na potezu: Greedy  
AI potez: Reroll [1,2,3]  
Kockice: [2,1,3,1,2]  
...

-----  
Na potezu: Greedy  
AI potez: Reroll [1,2,3]  
Kockice: [2,4,5,6,2]  
...

-----  
Na potezu: Greedy  
AI potez: Write Twos / Up  
Upisano u Twos / Up

-----  
Na potezu: Rand  
AI potez: Roll  
Kockice: [5,2,4,6,4]  
...

```

-----
Na potezu: Rand
AI potez: Write Yamb / Down
Upisano u Yamb / Down

```

```

-----
Na potezu: Greedy
AI potez: Roll
Kockice: [1,1,2,1,5]
...

```

```

-----
Na potezu: Greedy
AI potez: Write Ones / Up
Upisano u Ones / Up

```

```

=== KRAJ IGRE ===
Rezultati:

```

```

Igrač: Rand

```

	down	up	free
ones	[ 2]	[ 0]	[ 3]
twos	[ 2]	[ 0]	[ 0]
threes	[ 0]	[ 0]	[ 3]
fours	[12]	[ 4]	[ 4]
fives	[ 5]	[ 0]	[ 5]
sixes	[ 6]	[ 0]	[ 0]
max	[15]	[11]	[21]
min	[17]	[25]	[17]
straight	[ 0]	[ 0]	[ 0]
full	[ 0]	[ 0]	[ 0]
poker	[ 0]	[ 0]	[ 0]
yamb	[ 0]	[ 0]	[ 0]

Ukupno: 54

```

Igrač: Greedy

```

	down	up	free
ones	[ 3]	[ 3]	[ 2]
twos	[ 6]	[ 4]	[ 4]
threes	[ 6]	[ 9]	[ 9]
fours	[ 4]	[16]	[12]
fives	[15]	[20]	[15]
sixes	[ 6]	[12]	[18]
max	[16]	[13]	[18]

min	[20]	[20]	[20]
straight	[ 0]	[ 0]	[45]
full	[ 0]	[ 0]	[38]
poker	[ 0]	[ 0]	[ 0]
yamb	[ 0]	[ 0]	[ 0]

Ukupno: 270

## 6 Zaključak

Projekt pokazuje kako se Jamb može modelirati u funkcionalnom jeziku Haskell kroz modularan dizajn i čvrsta pravila tipova. Dobivena je funkcionalna verzija igre s podrškom za čovjeka i AI protivnike, te proširenim bodovanjem i završetkom igre.

Moguća poboljšanja:

- Naprednije AI strategije (heuristike, očekivani dobitak).
- Vizualno bogatije sučelje.
- Analiza rezultata kroz više simulacija.

## 7 Reference

- Pravila igre Jamb: <https://en.wikipedia.org/wiki/Yahtzee>
- Haskell dokumentacija: <https://haskell.org>
- Learn You a Haskell by Miran Lipovača: <https://learnyouahaskell.github.io/>
- Predavanja, FIPU – Funkcionalno programiranje