

Vještačka inteligencija za igru Snake

Ahmed Zekić

Prirodno-matematički fakultet Sarajevo

Teorijska kompjuterska nauka

Sarajevo, Bosna i Hercegovina

ahmedzekic10@gmail.com

Abstract—U ovom radu su predstavljena tri algoritma pomoću kojih se može razviti vještačka inteligencija za zmiju u igri Snake. Igra se sastoji od ploče po kojoj se kreće zmija i jedne jabuke na toj ploči. Ako zmija pojede jabuku, tijelo zmije se povećava, a nova jabuka se stvara na drugom mjestu. Igra je gotova ako zmija udari u sebe ili izađe izvan granice ploče.

Ključne riječi—zmija, vještačka inteligencija, BFS, DFS, Hamiltonov ciklus

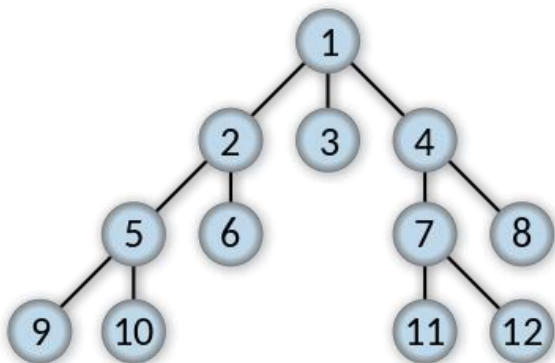
I. UVOD

Na početku igre, na nasumičnom mjestu na ploči se nalazi jabuka, a na sredini ploče se nalazi zmija. Dužina zmije je jedan, to jeste na ploči je samo glava zmije. Svaki put kada glava zmija dođe na mjesto gdje je jabuka (zmija je pojela jabuku), generiše se nova jabuka na slobodnom mjestu na ploči (mjesto gdje je zmija su zazuzeta), a tijelo zmije se povećava za jedan. Zmija se može kretati gore, dole, lijevo i desno. Bitno je napomenuti da nije dozvoljeno kretanje zmije unazad, to jeste ako je zmija išla ulijevo, ne može u sljedećem potezu ići udesno. Cilj igre je da tijelo zmije bude što veće, pri čemu je maksimalna dužina tijela veličina ploče. Ako zmija dostigne maksimalnu dužinu, odigrana je „savršena igra“. Igra završava ako glava zmije udari u svoje tijelo ili ako udari u granice ploče. S obzirom da se ploča može posmatrati kao neusmjereni, bestežinski graf, algoritmi za pretragu grafa mogu poslužiti kao osnova za razvijanje vještačke inteligencije za zmiju.

II. ALGORITMI ZA VJEŠTAČKU INTELIGENCIJU ZMIJE

A. Breadth-first search (BFS)

BFS je algoritam za prolazak ili pretragu kroz graf. Algoritam počinje od nekog čvora u grafu i prolazi kroz sve njegove susjedne čvorove na trenutnoj dubini, prije prelaska na čvorove veće dubine.

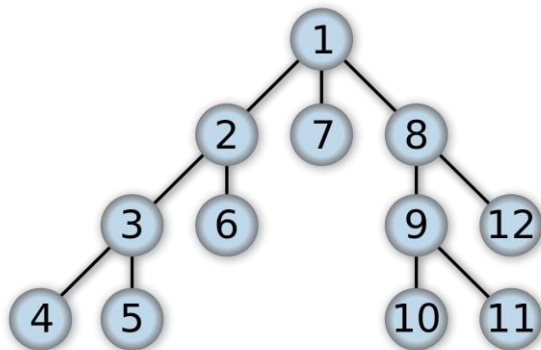


Slika 1: BFS obilazak grafa

Na slici 1 je prikazan primjer obilaska grafa uz pomoć BFS algoritma. Bitno je napomenuti da BFS daje najkraći put od početnog čvora do cilja.

B. Depth-first search (DFS)

DFS je algoritam za prolazak ili pretragu kroz graf. Algoritam počinje od nekog čvora u grafu i prolazi što je dublje moguće kroz svaku granu prije nego pređe na drugu.



Slika 2: DFS obilazak grafa

Na slici 2 je prikazan primjer obilaska grafa uz pomoć DFS algoritma.

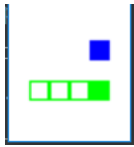
C. Hamiltonov ciklus

U teoriji grafova, Hamiltonov put je put u usmjerenom ili neusmjerenom grafu koji prođe kroz svaki čvor grafa tačno jednom. Hamiltonov ciklus je Hamiltonov put koji je ciklus. Pronalazak ovog ciklusa je NP-kompletni problem.

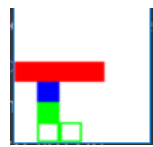
III. IMPLEMENTACIJA ALGORITAMA

Ploča na kojoj se igra Snake može se posmatrati kao neusmjereni, bestežinski graf. Tada je čvor grafa polje te ploče. Njegovi susjedi su gornje, donje, lijevo i desno polje od njega (ako postoje). U čvorove se neće ubrajati ona polja koja su zauzeta. To su polja na kojima se nalazi zmija ili neki drugi objekat koji predstavlja prepreku. Za testiranje algoritama je uzeta ploča dimenzije 6 x 6 koja je prikazana na slici 3. Ista ta ploča, ali sa dodatnom preprekom je prikazana na slici 4.

Plavom bojom je obojena jabuka, zelenom zmija, a crvenom prepreka.



Slika 3: Ploča 6x6



Slika 4: Ploča 6x6 sa preprekom

A. BFS

Algoritam je implementiran na osnovu sljedećeg pseudokoda:

```

1  procedure BFS( $G, start\_v$ ):
2      let  $S$  be a queue
3       $S.enqueue(start\_v)$ 
4      while  $S$  is not empty
5           $v = S.dequeue()$ 
6          if  $v$  is the goal:
7              return  $v$ 
8          for all edges from  $v$  to  $w$  in
 $G.adjacentEdges(v)$  do
9              if  $w$  is not labeled as
discovered:
10                 label  $w$  as discovered
11                  $w.parent = v$ 
12                  $S.enqueue(w)$ 

```

Pokretanjem 10 puta algoritma na običnoj ploči dobiju se sljedeći prosječni rezultati:

- Zmija pojede 13, 2 jabuke po igri (max je 36)
- Odigra 55,8 poteza po igri
- Odigra 4,22 poteza prije nego pojede jabuku

Isti algoritam pokrenut 10 puta na ploči sa preprekom daje sljedeće rezultate:

- Zmije pojede 7,2 jabuke po igri (max je 32)
- Odigra 33,3 poteza po igri
- Odigra 4,625 poteza prije nego pojede jabuku

Igra završava onda kada zmija zatvori sama sebe, to jeste onda kada BFS ne može naći jabuku na ploči.

B. DFS

Algoritam je implementiran na osnovu sljedećeg pseudokoda:

```

1  procedure DFS-iterative( $G, v$ ):
2      let  $S$  be a stack
3       $S.push(v)$ 
4      while  $S$  is not empty
5           $v = S.pop()$ 
6          if  $v$  is not labeled as
discovered:
7              label  $v$  as discovered
8              for all edges from  $v$  to  $w$ 
in  $G.adjacentEdges(v)$  do
9                   $S.push(w)$ 

```

Pokretanjem 10 puta algoritma na običnoj ploči dobiju se sljedeći prosječni rezultati:

- Zmija pojede 11, 7 jabuke po igri (max je 36)
- Odigra 77 poteza po igri
- Odigra 6,58 poteza prije nego pojede jabuku

Isti algoritam pokrenut 10 puta na ploči sa preprekom daje sljedeće rezultate:

- Zmije pojede 7,4 jabuke po igri (max je 36)
- Odigra 42,9 poteza po igri
- Odigra 4,79 poteza prije nego pojede jabuku

Igra završava onda kada zmija zatvori sama sebe, to jeste onda kada DFS ne može naći jabuku na ploči.

C. Hamiltonov put

Implementacija ovog algoritma se bitno razlikuje od BFS-a i DFS-a zato što cilj nije jabuka, već kraj zmije. Ako zmija stalno ide do svog repa najdužim mogućim putem, to garantuje da se zmija nikad neće zatvoriti i samim tim će se odigrati „savršena“ igra. Pozicija jabuke uopšte nije bitna – zmija će je pojesti na putu do repa.

Najduži put se nalazi tako što kao početni put uzmemo najkraći put, to jeste BFS od glave do repa zmije. Nakon toga, proširujemo taj put tako što za svaki par uzastopnih čvorova, pokušava se ubaciti ≤ 2 slobodna čvora tako da se sačuva prvobitni poredak puta kao što je prikazano na slici 5. Postupak se ponavlja dok se ne dobije najduži put.

A	B	G	D
E	F	G	H
I	J	K	L
M	N	O	P

BFGCD, BF \rightarrow BAEFGCD

A	B	G	D
E	F	G	H
I	J	K	L
M	N	O	P

Slika 5: Povećavanje puta

Pokretanjem 10 puta algoritma na običnoj ploči dobiju se sljedeći prosječni rezultati:

- Zmija pojede 36 jabuke po igri (max je 36)
- Odigra 364,1 poteza po igri
- Odigra 10,11 poteza prije nego pojede jabuku

Isti algoritam pokrenut 10 puta na ploči sa preprekom daje sljedeće rezultate:

- Zmije pojede 32 jabuke po igri (max je 32)
- Odigra 332,4 poteza po igri
- Odigra 10,38 poteza prije nego pojede jabuku

IV. ZAKLJUČAK

BFS algoritam se pokazao nešto boljim nego DFS zato što traži najkraći put do jabuke. Zbog svoje mogućnosti da odigra „savršenu“ igru najbolji je algoritam koji traži Hamiltonov put iako pomoću njega zmiji treba više poteza da pojede jabuku.

V. REFERENCE

- [1] https://en.wikipedia.org/wiki/Breadth-first_search
- [2] https://en.wikipedia.org/wiki/Depth-first_search
- [3] <https://towardsdatascience.com/slitherin-solving-the-classic-game-of-snake-with-ai-part-1-domain-specific-solvers-d1f5a5ccd635>