

Rešavanje Slide Puzzle Igre

Uvod

Slide puzzle igra je klasična igra zagonetke koja zahteva od igrača da pomeraju delove u okviru ograničenog prostora kako bi postigli određeni ciljni raspored. Osnovna verzija ove igre obuhvata 3x3 tablu sa 8 pločica numerisanih od 1 do 8 i jednim praznim mestom koje omogućava pomeranje pločica. Igrači pomeraju pločice horizontalno ili vertikalno, ciljajući da postave pločice u određeni redosled koristeći što manje poteza. Ovaj projekat ima za cilj implementaciju ove igre u više faza, počevši od osnovne 3x3 puzle, a zatim proširujući igru na proizvoljne dimenzije i dodatne funkcionalnosti, istražujući algoritme za generisanje solvable stanja i algoritme za rešavanje puzle.

Funkcionalni zahtevi

Prva faza

1. Implementirati osnovnu logiku igre za 3x3 slide puzzle, uključujući prikaz table, pomeranje pločica i proveru da li je puzla rešena.
2. Omogućiti inicijalizaciju igre sa random stanjem table koje je solvable. To znači da se od početnog stanja može doći do ciljnog stanja pomeranjem pločica unutar pravila igre.
3. Implementirati korisnički interfejs (UI) koji omogućava korisniku da pomeri pločice klikom na njih ili korišćenjem tastature.

Druga faza

1. Proširiti igru tako da podržava table proizvoljnih dimenzija ($n \times n$), gde je n veće od 3. Korisnik treba da može da izabere dimenziju table pre početka igre.
2. Generisati solvable inicijalno stanje za proizvoljne dimenzije table. Implementirati algoritam koji osigurava da je svako generisano stanje rešivo.
3. Dodati funkcionalnost koja omogućava korisniku da sačuva trenutno stanje igre i nastavi kasnije.

Treća faza

1. Implementirati algoritam za automatsko rešavanje slide puzzle igre. Ovo uključuje razvoj ili primenu postojećih algoritama za rešavanje puzli, kao što su A* ili IDA*.
2. Omogućiti korisniku da odabere opciju "Reši" koja pokreće algoritam za rešavanje i prikazuje korake do rešenja.
3. Dodati opciju za podesavanje težine puzle pri inicijalizaciji, što može uključivati broj poteza potrebnih da se dođe od početnog do ciljnog stanja.

Nefunkcionalni zahtevi

1. Interfejs treba da bude intuitivan i jednostavan za korišćenje, sa jasnim instrukcijama za korisnike.
2. Performanse aplikacije treba da budu optimizovane za brzo generisanje solvable stanja i efikasno rešavanje puzle, čak i za veće dimenzije table.

3. Kod treba da bude strukturiran i dokumentovan, omogućavajući lako dodavanje novih funkcionalnosti ili poboljšanja postojećih.
4. Aplikacija treba da bude robustna, sa odgovarajućim tretmanom grešaka i izuzetaka, osiguravajući stabilno iskustvo za korisnika.