

Stanje je definisano:

- listama n-torki (brReda, brKolone), koje pamte pozicije zidova
- n-torkama (brReda, brKolone), koje pamte pozicije pešaka i ciljnih/početnih polja
- celobrojnim promenljivama koje pamte broj preostalih zidova igrača
- n-torkom (broj redova i kolona), koja pamti dimenzije table
- bool promenljivom koja nam govori da li je na potezu računar ili čovek

main.py

main – pokreće aplikaciju i omogućava započinjanje nove partije nakon završetka prethodne

blockade.py

blockade – inicijalizuje stanje i obezbeđuje odigravanje poteza do završetka partije, a nakon završetka prikazuje završne poruke

moves.py

is_game_end – proverava da li je partija završena

is_wall_place_valid – proverava da li je mesto na kome se očekuje postavljanje zida validno

is_pawn_move_valid – proverava da li je očekivani potez pešaka validan

move_pawn – vraća novu poziciju pešaka, ukoliko je potez validan, suprotnom vraća tekuću poziciju tog pešaka

place_wall – postavlja zid na željenu poziciju i vraća pozitivan odgovor, ukoliko je dostupna, a u suprotnom vraća negativan odgovor

update_wall_connection_points – ažurira graf u kome je svaki zid predstavljen sa 3 tačke

add_neighbor_connection_point – pomoćna funkcija koje se poziva u update_wall_connection_points; dodaje nove susede u graf

generate_border_connection_points – inicijalizuje graf tako da su u njemu samo tačke ivica table

is_wall_connected_with_two_or_more_walls – proverava da li je zid ima kontakt sa drugim zidovima(uključujući i okvire tabele) u najmanje 2 tačke

view.py

show_table – vrši iscrtavanje tabele na osnovu tekućeg stanja

Ostale pomoćne funkcije služe za generisanje tabele u obliku niza Unicode karaktera , neophodnog za funkciju `show_table`, ispis u konzoli i učitavanje početnih parametara.

`path_finding.py`

`h_calculate_raw` – određuje cenu puta između dve pozicije na tabli

`find_path` – pronalazi putanju od pešaka do jednog odredišta i drugog

`a_star` – pronalazi putanju od pešaka do jednog odredišta

`generate_next_moves` – generiše sve moguće pozicije za trenutno izabranog pešaka

`do_walls_make_polygon` – ispituje da li je generisano zatvaranje na tabli

`min_max.py`

`min_max` – traži najbolji potez koji računar može da odigra

`next_states` – vraća sva stanja u koje se može preći iz zadatog stanja (funkcija promene stanja)

`is_state_good` – vrši odsecanje novogenerisanih stanja na osnovu heuristike da li je to stanje potrebno obraditi

`distance` – izračunava udaljenost između dve pozicije na tabli

`evaluate_state` – vrši procenu koliko konkretno stanje vodi određenog igrača ka pobedi

`max_value(min_value)` – vraća najoptimalnije stanje za `max(min)` igrača

`generate_walls_positions` – generiše sve zidove koje je moguće postaviti

`generate_pawns_positions` – generiše sve moguće pozicije za oba pešaka konkretnog igrača