

Stanje je definisano:

- listama n-torki (brReda, brKolone), koje pamte pozicije zidova
- n-torkama (brReda, brKolone), koje pamte pozicije pešaka i ciljnih/početnih polja
- celobrojn timer 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 kompjuter ili racunar

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_player_movement – proverava da li je očekivani potez pešaka validan

position_occupied – proverava da na željenoj poziciji već postoji pešak i u tom slučaju vraća prethodno slobodno polje

move_player – vraća novu poziciju pešaka, ukoliko je potez validan, suprotno vraća tekuću poziciju tog pešaka

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

h_calculate_optimized – računa vrednost za heuristiku

find_path_to_one – ako postoji put od pešaka do ciljnog polja vraća putanju kojom se stiže do ciljnog polja; ako ne postoji put vraća praznu listu

find_path – pronalazi putanju do oba ciljna polja ako postoji

generate_next_moves – generiše validne poteze za trenutnu poziciju pešaka

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.

read_move – vrši učitavanje poteza od strane korisnika

