Boriceanu Ioana-Roxana, grupa 331AB

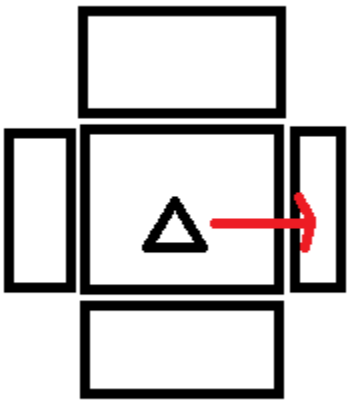
Pentru rezolvarea cerinței am implementat un automat cu 10 stări, iar trecerea de la o stare la alta se face pe frontul crescător al ceasului în partea secvențială. O să încerc în continuare să explic ce se întâmplă în fiecare stare și cum am gândit automatul.

Pentru început am definit cele 10 stări cu ajutorul unor macro-uri.

Avem starea 0: INIT în care le-am dat variabilelor row și col valorile coordonatelor poziției inițiale, l-am inițializat pe done cu 0 și pe maze_oe=1 ca să pot verifica în următoarea stare ce valoare se află în celula în care mă aflu și apoi am făcut trecerea către S1.

Stările S1, S3, S5 și S7 funcționează după același principiu, diferă desigur semnificația fiecăreia. S1 am considerat-o ca fiind starea în care „omulețul" meu vine din Estul labirintului(dreapta), S3 este starea în care „omulețul" vine de jos/Sud, S5 starea în care „omulețul" vine din Nord, iar S7 cea în care „omulețul" vine din Vest.

Jumătate din tema asta a fost gândită în Paint, deci o să atașez și o poză micuță ca să ajute explicația.



Considerăm că omulețul merge către Nord, deci de aflăm în starea S5, iar peretele pe care trebuie să îl urmărească se află în Est. El o să facă următorul lucru, pentru început verifică dacă se află cumva pe una din marginile labirintului adică row==63 || row==0 || col==63 || col==0 și, de asemenea, dacă în celula respectivă avem valoarea 0 sau 2, deci maze_in!=1. În cazul în care acestă condiție este

îndeplinită el o să știe că a ajuns la destinație și trebuie să se oprească, deci la următorul front crescător de ceas îi dă semnalului done valoarea 1 și trece din starea actuală în starea STOP, de unde nu mai trece în nicio altă stare, marchează celula cu 2 prin maze_we=1 și îl face și pe maze_oe=0 pentru că nu mai avem de făcut nicio citire. Dacă încă nu am ajuns la destinație atunci trece la următorul if unde verifică dacă în poziția pe care ne aflăm se găsește valoarea 0 sau 1. Dacă valoarea este 0, o marcăm cu 2 și trecem într-o stare suplimentară S6.

Stările suplimentare S2, S4, S6 și S8 le-am adăugat deoarece dacă egalam simultan maze_oe și maze_we cu 1, el îmi punea 2 și în poziții pe care aveam 1 nu 0 și apoi dacă trebuia să verific poziția respectivă în alt ciclu de ceas, el nu mai găsea „peretele” acolo, ci o celulă marcată cu 0 și își schimba întregul drum, în câteva cazuri făcând o buclă între 4 celule. De asemenea, la începutul celui de-al doilea always l-am inițializat pe maze_we cu 0 pentru că la un moment dat mi-am dat seama că, din păcate, nu-mi citește gândurile, maze_we rămâne 1 după ce îl modific și îmi pune 2 pe toate pozițiile prin care trece, cauzând aceleași probleme menționate mai sus.

În starea suplimentară S6, îl facem pe maze_oe=1 și verificăm dacă peretele din dreapta noastră(celula din dreapta ⇔col=col+1), în cazul nostru peretele din Est este 0 sau 1 și facem trecerea către starea în care omulețul merge către Est, în cazul nostru S1, iar în S1 începe să facă aceleași verificări pe care le-a făcut în S5. Dacă în celula respectivă găsim valoarea 0, atunci o marcăm cu 2 și verificăm peretele din dreapta, altfel ne întoarcem în celula din care am venit(col=col-1) fără să schimbăm nimic în celula respectivă și făcându-l pe maze_oe=1 ca să putem verifica în următoarea stare care este valoarea din celula în care ne mutăm.

Practic, omulețul urmărește mereu peretele drept, așa cum specifică enunțul temei, el verifică mereu valoarea din celula în care se află, dacă e 1 se întoarce de unde a venit, dacă e 0 atunci verifică valoarea care se află în celula din dreapta, dacă e 1 atunci se întoarce în căsuța din care a plecat și verifică următorul perete până când găsește o celulă pe care are voie să se mute, adică o celulă marcată cu 0 sau 2.

Știu că nu e cea mai eficientă metodă pentru că durează foarte, foarte mult să parcurgă întregul labirint în felul ăsta, dar e singura care mi-a ieșit. :D