

A* pathfinding per fillestaret

Pershndetje te gjithëve.

Ja ku vjen nje artikull interesant mbi Pathfinding.

C'eshte pathfinding?

Eshte nje teknike shume efikase per gjetjen e rruges me te shkurter ndermjet dy pikave plani, duke evituar c'do pengese pergjate saj.

Imagjinojme nje plan koordinativ te thjeshte, dhe ne te, dy pika referimi A dhe B. Sic dihet nga te gjithë ju, rruga me e shkurter per te shkuar nga nje pike A ne nje pike B te planit eshte largesia midis ketyre pikave (largesia ne kuptimin matematik te fjales).

Por, ne realitet mund te kemi te bejme me terrene tek te cilat, duke i pershtatur ne nje plan koordinativ, do te kishim qe: per te shkuar nga pika e nisjes A ne piken e deshiruar B, rruga e ndjekur rezulton te jete me doemos jo ne vije te drejte, per shkak te pengesave eventuale.

Pikerisht per te gjetur, nga rruget e ndryshme qe mund te ndiqnim per te shkuar nga A ne B, ate me te shkurter, na ndihmon teknika e Pathfinding. Kjo e fundit bazohet mbi nje algoritem te quajtur A*(A-star) . C'do aplikim korrekt i ketij algoritmi jep gjithmone nje rezultat:

- a) gjen nje rruge ndermjet dy pikave (rasti me i mire, rruga e gjetur eshte me te vertete me e shkurtra).
- b) nuk gjen asnje rruge ndermjet dy pikave (rasti me i keq , pikat jane fizikisht te paarritshme ndermjet tyre).

Permendim qe taknika e pathfinding eshte e perdorur ne shume lojra per kompjuter, ne vecanti per shume lojra qe egzekutohen nen Linux OS.

Artikulli : original "A* Pathfinding for Beginners", i shkruar nga Patrick Lester mund te gjendet ne <http://www.policyalmanac.org/games/aStarTutorial.htm>

A*(shqiptohet A-star) mund te jete i komplikuar per fillestaret. Ne web gjenden shume artikuj mbi pathfinding, pjesa me e madhe e te cileve i kushtohen atyre qe i kane marre bazat qe me pare. Ky perkundrazi, i kushtohet fillestareve te vertete ne kete koncept.

I rendesishem fakti qe, ky artikull nuk i referohet ndonje programi specifik apo ndonje gjuhe programimi te vecante. Per pasoje, mund te pershtatet per programe dhe rrethana te ndryshme dhe mbi te gjitha mund te implementohet ne shume gjuhe programimi sic jane C/C++, JAVA etj...

*Le te fillojme me pershkrimin e Pathfinding dhe A**

Intro : Zona e kerkimit

Le te marrim persiper te shkojme nga pika A ne piken B si ne figure me poshte, ku sic mund te vihet re, ne mes te dy pikave gjendet nje pengese (mur): mbi, poshte ose nepermjet te ciles nuk jemi ne gjendje te kalojme. Ne figure pika A, B dhe muri jane te ngjyrosur respektivisht me ngjyre: jeshile, te kuqe dhe blu.



[Figure. 1]

Sic shihet, e kemi ndare zonen e kerkimit ne nje “rrjet kutish”, per analogji me planin koordinativ. Ndarja ne kutiza e zones na lejon te reduktojme terrenin ne nje vektor bidimensional (matrice). Pikerisht thjeshtimi i zones si me siper, eshte hapi i pare ne pathfinding. Keshtu, c’do pjese apo njesi e zones se kerkimit (c’do element i vektorit), identifikohet nga nje kuti e cila mund te kete vetem dy lloj gjendjesh: e pershkueshme ose e paparshkueshme. Muri, per shembull, eshte i papershkueshem: e kemi paraqitur nepermjet tre kutizave me ngjyre blu. Kutizat me ngjyre te zeze perbejne te gjitha terrenin e pershkueshem. Njehere qe gjendet rruga nga A ne B, mjafton te levizim neper qendren e kutizes A, me pas ne qendren e kutizes se ardhshme e keshtu me rradhe derisa te gjendemi ne kutizen B. Qendrat e ketyre kutizave quhen “nyje”. Ky term i fundit eshte gjenerik per sa i perket njesive te ndarjes se zones se pathfinding. Keshtu, ne raste te tjera mund te na duhet ta ndajme zonen jo ne kutiza (katrore) por p.sh ne trekendesha, apo gjashkendesha apo forma gjeometrike te tjera... Ne pergjithesi nyja nuk eshte e thene te mendohet ne qender te njesise te ndarjes, ka raste qe mund te mendohet edhe ne borde. Ne po marrim ne konsiderate vetem rastin me te thjeshte: kutiza, dhe nyje ne qender te kutizes.

Fillojme kerkimin

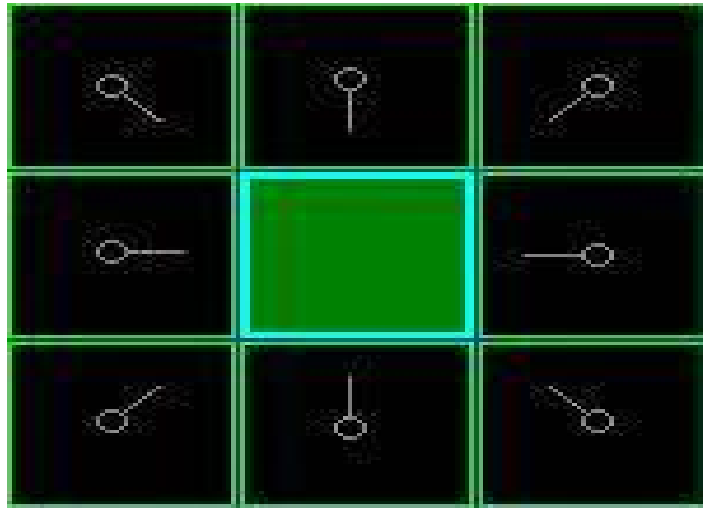
Pas thjeshtimit te zones se kerkimit ne nje numer nyjesh te manipulueshme fare lehte, hapi tjetër eshte gjetja e rruges me te shkurter. Kete e bejme duke u nisur nga pika A dhe duke konsideruar kutizat adjacente (kutizat qe linden menjehere prej nje kutize te dhene) derisa te arrijme ne B.

Nisim kerkimin duke vepruar si me poshte:

1. Ne fillim vendosemi ne piken e nisjes A dhe e shtojme kete te fundit ne nje “liste te hapur” nyjesh qe do te konsiderohen. Sa per te fiksuar idete, lista e hapur eshte dicka e ngjashme me listen e shpenzimeve. Per momentin kemi vetem nje objekt ne te, por shume shpejt do te kemi shume te tjere. Ne te gjenden kutizat te cilat mund te jene pjese e rruges me te shkurter ose jo. Pra keto kutiza s’jane gje tjetër vecse, ato qe duhet te shqyrtojme.
2. Konsiderojme te gjitha kutizat e arritshme, adjacente (qe dalin prej) te pikes se nisjes, duke injoruar ato qe jane te papershkueshme (pjese se pengesave te ndryshme). Shtojme dhe keto ne listen e hapur. Per secilen prej tyre memorizojme kutizen A si “kutiza parent” dmth si kutiza nene. Kjo kutize parent eshte e rendesishme per te percaktuar rrugen me te shkurter. Se si, do ta shpjegojme me poshte.

3. Spostojme piken e nisjes A ne nga lista e hapur ne nje “liste te mbyllur” e cila permban ato kutiza te cilat nuk kemi nevoje (tani per tani) ti marrim ne shqyrtim.

Pasi keni bere si me lart, duhet te keni dicka te ngjajshme me figuren me poshte, ne te cilen: kutiza me ngjyre leshile te erret tregon kutizen e nisjes. Bordet e saj , sic vihet re, jane ngjyrosur me ngjyre blu per te treguar qe kete kutize e kemi shtuar ne listen e mbyllur. Te gjitha kutizat direkt adjacente (bija) te A-se, jane tashme ne listen e hapur te kutizave qe jane ne shqyrtim, dhe ne figure jane treguar me borde te ngjyrosur me leshile te hapur. Secila ka nje indikator ngjyre gri, qe tregon kutizen nene, qe ne kete rast eshte kutiza e nisjes A.



[Figure. 2]

Pas pak, do te zgjedhim vetem nje prej kutizave bija te listes se hapur dhe pak a shume do te veprojmë si me poshte. Cilen kutize do te zgjedhim? Ate me koston F me te vogel.

Percaktimi i rruges

Cilesi per te percaktuar kutizen e duhur per te parcaktuar me pas te tere rrugen, eshte ekuacioni i meposhtem:

$$F = G + H$$

ku :

- G eshte kostoja e levizjes nga pika e nisjes A ne nje pike te dhene te planit, nepermjet rruges se gjetur per te arritur tek pika e dhene.
- H eshte vlera e perafert e koston per te levizur nga pika e dhene deri tek pika pika e destinuar, pika B. Themi e perafert sepse, derisa nuk kemi rrugen per te shkuar, nuk e dime me saktesi sa hapa duhet te bejme dhe neper c'fare kutizash duhet te kalojme. Ne kete rast konsiderohen edhe kutizat e papershkueshme.

Nje menyre per te llogaritur vleren e H eshte e meposhtemja:

$$H = 10 * [abs(currentX - targetX) + abs(currentY - targetY)]$$

ku:

- ° abs është funksioni i vleres absolute. Kujtojmë që vlera absolute e një numri real a , është $\{-a, \text{ për } a < 0 \text{ dhe } a, \text{ për } a \geq 0\} = |a| \geq 0$, për çdo a nga \mathbb{R}
- ° current është pika e dhënë
- ° target pika e destinacionit
- ° X dhe Y respektivisht abshica dhe ordinata e një pike plani.

Variante të tjera të llogaritjes së vlerës H: <http://www.policyalmanac.org/games/heuristics.htm>

Rruga jone përfitohet duke përsëritur zgjedhjen e kutizës me kosto F më të vogël, nga lista e hapur. Ky proces do të shtjellohet pak më poshtë. Por tani le të shohim më vëmendje si e llogarisim ekuacionin.

Si thamë më lart, G është kostoja e levizjes nga pika e nisjes në një pike të dhënë të planit duke ndjekur rrugën e gjetur për të përfituar këtë pike. Në këtë shembull c' do levizje horizontale apo vertikale ka një kosto 10 dhe c' do levizje në diagonale ka kosto 14. Përdorim këta koeficientë sepse kostoja për të levizur në diagonale është rrethja katërore e 2-shit, dmth 1.414 herë koston e levizjes në horizontal apo vertikal. Për të evituar të trajtojmë me numra racional, shumëzohet me 10 dhe 14, për të paraqitur kostot respektive. Tashmë kostot janë të shprehura me numra të plotë.

Për një kutizë të dhënë, përcaktimi i kostos G është numri i hapave të bërë për të arritur në këtë pike që nga pika e fillimit, duke shtuar 10 nëq levizja është bërë në drejtim ortogonal (jo diagonal) nga kutiza nënë, dhe 14 nëq levizja është bërë në diagonal.

H mund të llogaritet sic thamë dhe më lart, në mënyrë të ndryshme. Metoda që kemi përdorur quhet “metoda Manhattan”, sipas së cilës llogarisim numrin total të kutizave levizjeve në horizontal dhe në vertikal për të arritur nga pika e dhënë në pikën e destinacionit, duke injoruar levizjen në diagonale dhe duke injoruar c' do pengesë përgjatë rrugës. Këtu numri është shumëzohet me 10.

F llogaritet si shuma e G dhe H . Rezultatet e hapit të parë janë ilustruar në figurën më poshtë. Vlerat F , G , dhe H janë të shkruara për c' do kutizë, brenda saj. Sic shihet, në kutizën menjëherë në të djathtë të pikës së nisjes, F është shkruar lart në të majtë; G në të majtë poshtë, dhe F në të djathtë poshtë.



[Figure. 3]

Le te shohim disa prej ketyre kutizave. Ne ato ku vlera $G = 10$, eshte e tille sepse duhet vetem nje hap per te shkuar ne drejtim horizontal apo vertikal ne ate kutize nga pika e nisjes A. Kutizat e tjera kane $G = 14$. Edhe per keto kutiza mafton nje hap i vetem, por ne diagonal.

Vlerat H jane llogaritur duke vleresuar largesine Manhatan nga kutiza e kuqe, duke levizur vetem ne ortogonal dhe duke injoruar murin ne mes. Duke perdorur kete metode, kutiza menjehere ne te djathte te pikes se pikes jeshile (pika e nisjes) eshte 3 kutiza larg pikes se kuqe. H vlen 30. Kutizat siper dhe poshte kesaj kutize te fundit kane $H = 40$, nje kuti ne vertikale dhe tre ne horizontale.

Per c' do kutize $F = G+H$.

Vazhdojme kerkimin

Ne kerkim te rruges me te shkurter, duhet lehte fare, te zgjedhim nga kutizat e listes se hapur ate me kosto F me te vogel. Me pas me kete kutize kryejme operacionet e meposhtme:

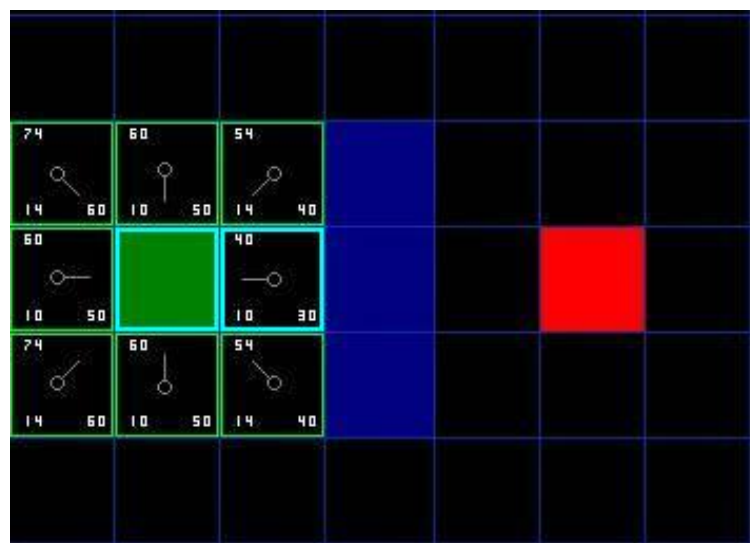
4. E spostojme ate nga lista e hapur ne listen e mbyllur.
5. Shqyrtojme kutizat adjacente. Injorojme ato qe tashme ndodhen ne listen e mbyllur ose te paparshkueshme, shtojme ne listen e hapur ato qe nuk jane ne kete liste. Kutizen e seleksionuar e bejme "parent" te kutizave te reja.
6. Ne qofte se nje kutize adjacente ne shqyrtim, ishte qe me pare e ne listen e hapur, verifikojme nese kjo rruge per ta perfituar kete kutize eshte me shkurtra(me e mira). Me fjale te tjera kontrollojme nese vlera G per kete kutize eshte me e vogel po te perdorim kete rruge per te mberritur tek kutiza vete.

Nqs jo, mos nuk bejme asgje.

Ne te kundert , nqs kostoja G e rruges se re eshte me e vogla, ndryshojme kutizen "parent" te kutizes adjacente se kutizes se seleksionuar (nuk eshte e komplikuar sic mund te duket!).

Ne figure ndryshojme drejtimin e indikatorit te "parent" per kutizen ne fljale. Me se fundi llogarisim perseri vlerat F dhe G per kete kutize.

Ok, le te shohim si funksionon. Nga 9 kutizat qe kishim ne fillim, tani kemi vetem 8, pasi kutizen e nisjes A e spostuam ne listen e mbyllur. Nga keto te mbeturat, kutize me F me te vogel eshte ajo menjehere ne te djathte te asaj te nisjes, me $F = 40$. Seleksionojme kete kutize si kutiza pasardhese. Bordet ja ngjyrosim me blu.



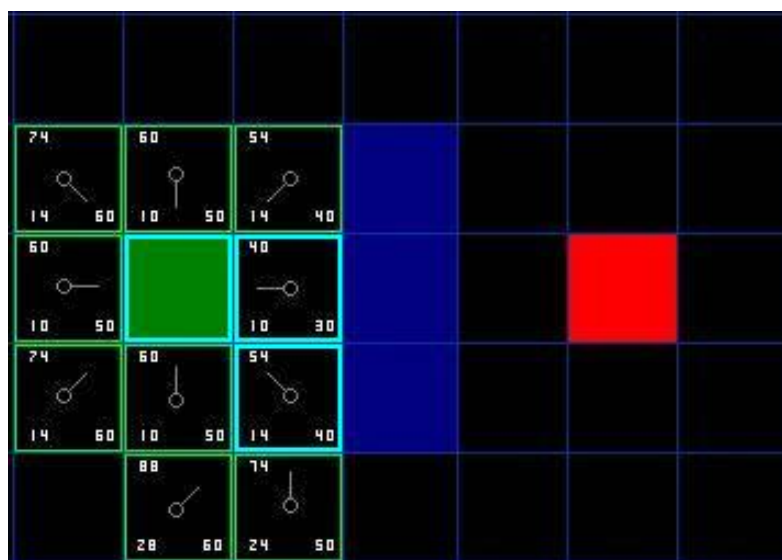
[Figure. 4]

Pike se pari e spostojme kete kutize ne listen e mbyllur (ku gjendet edhe A). Tashme konsiderojme kutizat adjacente te saj. Ato ne te djathe i injorujme sepse jane pjese e murit. Injorujme gjithashtu ate direkt ne te majte sepse ndodhet ne listen mbyllur. Kutizat e tjera gjendeshin qe me pare ne listen a hapur. Duhet te verifikojme tani nqs: per te shkuar nga pika A ne secilen nga kutizat e listes se hapur ka ndonje rruge me te vogel se ajo e bere duke perdorur kutizene e seleksionuar per te shkuar tek te njetat kutiza.

Le te shohim kutizen menjehere siper kutizes se seleksionuar. Vlera G aktuale e saj eshte 14. Nqs perdorim rrugen qe kalon nepermjet kutizes se seleksionuar per te shkuar tek kutiza ne konsiderate kemi $G' = 20 > 14 = G$. Prandaj per te shkuar tek kutiza e ne fjale eshte me mire te kalojme direkt nga pika e nisjes meqenese, ne me te mirin nga rastet e tjere, do te na duhej te benim dy hapa ortoganale... dmth do te kishim $G = 20$.

E perserisim procesin per te 4-ta kutizat qe gjenden ne listen e hapur. Pasi zgjedhim ate me te miren e shtojme ne listen mbyllur dhe konsiderojme kutizat adjacente te saj .

Ke zgjedhem ? Nga 2 kutizat me $F = 54$, zgjedim ate me poshte, thjesht per pune shpejtesie meqe e kishim shtuar me vone ne listen e hapur se kutizen me lart. Tashme kutiza e seleksionuar eshte kutiza e zgjedhur. Mos harrojme: ato qe ndodhen ne listen e mbyllur jane me borde blu.



[Figure. 5]

Nga te gjitha kutizat adjacente me kutizen e fundit te seleksionuar, ate direkt ne te djathte e injorujme pasi eshte pjese e murit. Injorujme gjithashtu kutizen qe qendron menjehere poshte saj. Kjo gje, per faktin se nga kutiza e seleksionuar nuk mund te shkohet menjehere tek kutiza ne fjale pa u ndeshur me paretet e murit. Na duhet me pare te shkojme poshte me nje hap, dhe me pas ne te djathte, sic mund te shihni nga figura ketu siper. Mbeten 5 kutiza.

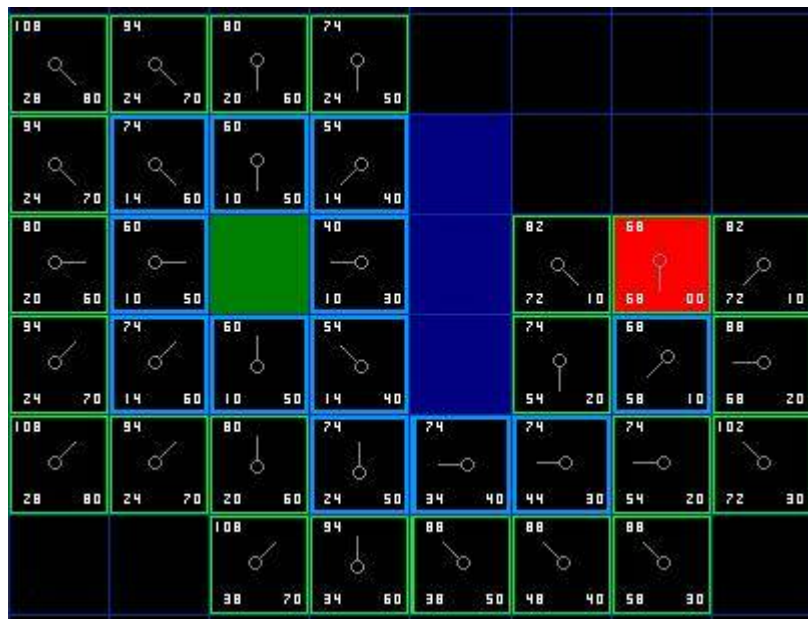
Nga keto, dy kutizat me poshte nuk gjenden ne listen e hapur, per pasoje i shtojme dhe ato.

Nga tre kutizat e tjera, dy gjenden ne listen e mbyllur... i injorujme.

Per kutizen tjetere (te mbeturen), shohim per nje G me te vogel...Resultati negativ.

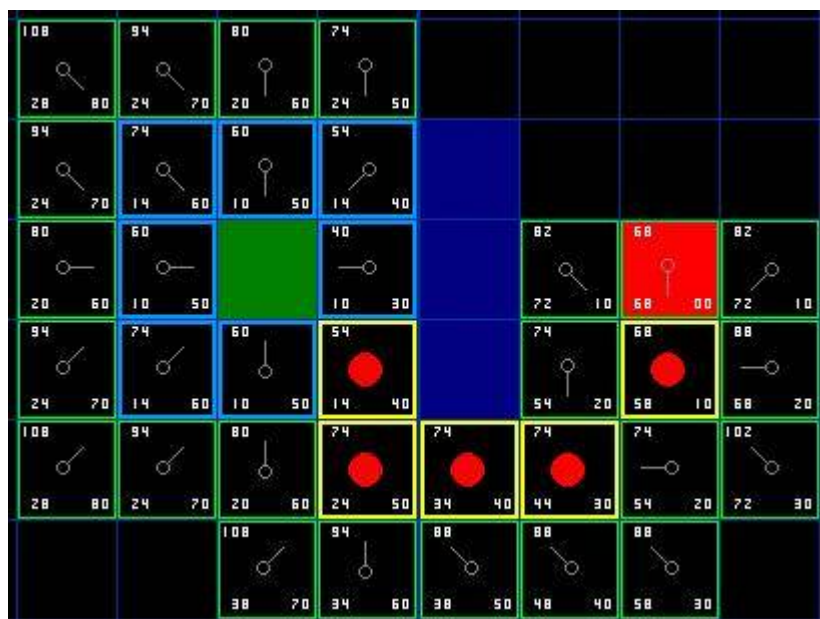
Ajo me F me te vogel, eshte kutiza direkt poshte kutizes se seleksionuar.

Perserisim kete proces derisa te kemi shtuar ne listen e hapur kutizen e destinacionit B.
 Deri tani kemi nje shtate si ne figure:



[Figure. 6]

Si te percaktojme rrugen me te shkurter? Thjesht duke u nisur nga kutiza e destinacionit B dhe duke ndjekur rrugen e treguar nga indikatorit parent i c’do kutize, derisa te arrijme tek kutiza e nisjes A. Duke kryer pikerisht kete rruge te gjetur, nga A ne B , po harxhojme me pak energji se ne c’do rruge tjeter te mundeshme nga A per ne B.
 Situata perfundimtare eshte paraqitur ne figuren me poshte.

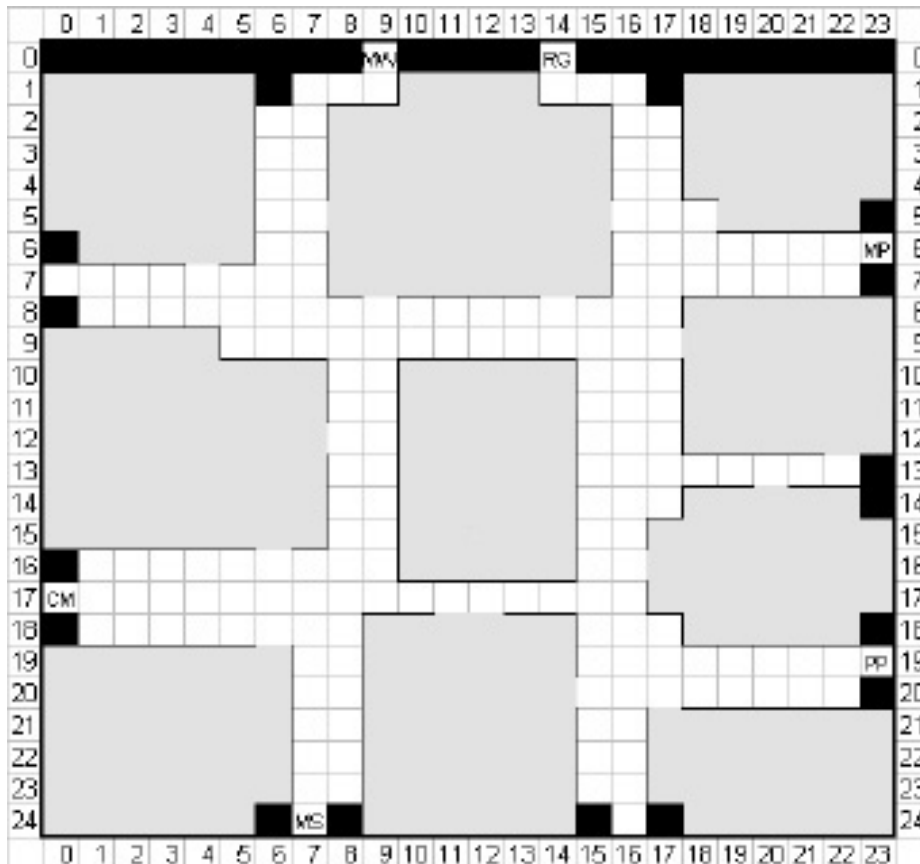


[Figure. 7]

Permbledhje e algoritmit A*

- 1) Shto kutizen e nisjes ne listen e hapur.
- 2) Perserit c'ka me poshte:
 - a) Shiko per kutizen me koston F me te vogel ne listen e hapur. I referohemi asaj si kutiza "current".
 - b) E spostojme tek lista e mbyllur.
 - c) Per secilen nga 8 kutizat adjacente te saj...
 - 1) Nqs eshte e papershkueshme ose nqs eshte ne listen e mbyllur , injoroje. Ne te kundert bej si me poshte...
 - a) Nqs nuk eshte ne listen e hapur, shtoje. Bej kutizen "current"... , "parent" te kesaj kutize. Memorizo koston G, H, dhe F te kutizes.
 - b) Nqs ndodhej qe me pare ne listen e hapur, shiko nese rruga e tanishme per te patur kete kutiza eshte me e mira, duke perdorur koston G. G-ja me e vogel tregon se rruga qe i korrispondon kesaj G-je, eshte me e shkurtra. Kur kjo ndodh, ndrysho "parent" te kesaj kutize ne kurrent.
 - d) Ndalo kur:
 - 1) Shton ne listen e hapur kutizen e destinacionit. Ne kete rast rruga u gjet dhe eshte me e mira (me e shkurtra).ose:
 - 2) Nuk gjen dot kutizen e destinacionit dhe lista e hapur eshte bosh. Ne kete rast rruga nuk u gjet (pikat jane te paarrishme fizikisht ndermjet tyre).
- 3) Memorizojme rrugen e gjetur, duke u nisur nga kutiza e destinacionit dhe duke ndjekur me rradhe kutizat "parent" derisa te mberrijme tek kutiza e nisjes.

Ne arkivin .zip po sjell nje aplikim te A* ne Java, qe mu desh te implementoja disa kohe me pare per nje projekt mbi nje loje te quajtur Cluedo (ose Clue). Per te mos u futur shume ne detaje te lojes (qe nuk na interesojne) medojme te kemi figuren me poshte:



Jane te pershkueshme te gjitha dhe vetem, kutizat e ngjyrosura me ngjyre te bardhe. C’do ambient tjetër, përfshirë zonat me ngjyre gri (ne Cluedo jane dhoma) dhe kutizat e ngjyrosura me te zeze , eshte i papershkueshem. Levizja behet vetem ne ortogonal.

Objektivi : percaktimi i rruges me te shkurter ndermjet dy kutizave c’fare do te tabelës.

Sic mund te vihet re per te shkuar p.sh nga kutiza (7, 24) ne kutizen (16, 24) mund te shkohet duke ndjekur rruge te ndryshme por vetem njera eshte me e shkurtra nga gjithe te tjerat, pikersht ajo qe jepet nga aplikimi (jo grafik, por tekstual) i algoritmit A* (classa AStar e arkivit .zip). Per te egzekutuar programin, kodi i te cilit ndodhet ne arkivin .zip, do te keni nevojë per nje JVM, te cilen mund ta shkarkoni lirisht nga <http://java.sun.com> dhe me pas, ta instaloni ne makinën tuaj.

Nen windows: Nqs keni instaluar versionin aktual te Java Virtual Machine, pasi keni dekompresuar archivin Astar.zip ne desktop, egzekutoni file-in batch brenda karteles Astar. Nqs perkundrazi, keni nje versioni te vjeter te JVM mjafton te modifikoni ne kete file batch, path-in e compilerit javac (dhe java) me ate tuajen.

Nen linux: Pasi keni dekompresuar ne /home/user/ archivin Astar.zip, nga shell si user, me tre komanda te thjeshta:

```
...#cd Astar
...#javac ./*.java
...#java AStar
```

COReE fton te gjithë ne: www.alblinux.net dhe <http://www.albalinux.org>
Per c' do verejtje apo sugjerim jeni te mirepritur ne thecoree@yahoo.it

COReE 29 /09/2004