

**ÓZBEKISTAN RESPUBLIKASÍ INFORMACIYALÍQ
TEXNOLOGIYALARÍ HÁM KOMMUNIKACIYALARÍN
RAWAJLANDÍRÍW MINISTRLOGI**

**MUHAMMED AL-XOREZMIY ATÍNDAĞÍ TASHKENT
INFORMACIYALÍQ TEXNOLOGIYALARÍ UNIVERSITETI
NÓKIS FILIALÍ**

Informaciyalıq texnologiyalar kafedrası

Kompyuter injiniring baǵdarı

Qorǵawǵa ruxsat etildi

Kafedra baslıǵı

f.-m.i.k. Turenıyazova A.

_____ 2019 j.
«__»_____

«Graflarda qıdırıw algoritmların Pak-man proektinde ámelge asırıw»

temasında

PITKERIW QÁNIYGELIK JUMÍSÍ

Pitkeriwshi:

Babajanova N.

Ilimiy basshı:

Baekeev T.

Nókis - 2019 j.

Mazmuni

KIRISIW	3
I-BAP. TIYKARGÍ TÚSINIKLER	5
1.1 Graflarda qıdırıw algoritmleri	5
1.2 Pakman oyını haqqında	9
II BAP. PAKMAN PROEKTI HAQQÍNDÁ ULÍWMA TÚSINIKLER	14
2.1 Pak-Man proekti haqqında maǵlıwmatlar	14
2.2 Pak-Man proektinde qıdırıw bólimi	15
2.3 Multiagentli ortalıqda qıdırıw	22
III BAP. QÍDÍRÍW ALGORITMLERIN PROGRAMMALASTÍRÍW	29
3.1 Python programmalastırıw tili	29
3.2 Qıdırıw algoritmleri realizaciyası	30
JUWMAQLAW	47
PAYDALANÍLGÁN ÁDEBIYATLAR	48
QOSÍMSHALAR	Ошибка! Закладка не определена.

KIRISIW

Jasalma intellekt túsinigi quramalı túsinik bolıp esaplanadı. Eger aqırǵı 30 jil dawamındaǵı pikirlerdi ulıwmalastırıp aytılsa onda sonday túsinik payda boladı, yaǵnıy insanlar ózine uqsap pikirley alatuǵın, háreket qıla alatuǵın, sheshim qabıl ete alatuǵın mashinanı jaratıwǵa háreket etip atır. Máqset qandayda bir ámellerde artıqsha waqıt, kúsh sarplamay sheshiwden ibarat. Aqırǵı waqıtları jasalma intellekt tarawına bolǵan qızıǵıw artıp barmaqta. Programmalıq támiynat aqlı bolıp atır, sol mene birge kúndelik qurılımalarda aqlı qurılımalarǵa aylanıp barmaqta.

Búgingi kúnde jasalma intellekt ilmiy izerklewlerdiń ishinde eń qızıǵıw artıp baratırǵan taraw bolıp esaplanadı. Bul jónelis menen kópshilik taraw iyeleri kibernetikler, psixologlar, filosoflar, matematikler, lingvistler hám injenerler shuǵıllanıp atır. Usı jasalma intellekt tarawında kópshilik sheshim tabılmaǵan máselelerge sheshim tabılıp atır. Bul jóneliske qızıǵıwshı tarawlar sanı artıwı menen, jasalma intellektiń ózi bir qansha tarawlarǵa bólinip ketdi. Olar, mashinalı oqıtıw, ekspert sistemalar, evolyucion esaplaw, kompyuter kóriwi, hawazdı tanıw hám t.b.

Jasalma intellekt tarawınıń rawajlanıwı menen, ol arqalı sheshiletuǵın máselelerdiń túride kóbeyip barmaqda. Máselen, kórinislerdi tanıw. hawazdı tanıw, mashinalı tekstti awdarlamalaw hám t.b. túrdegi máseleler. Bunday máselelerdiń tiykarındaǵı ámel bul qıdırıw bolıp esaplanadı. Qıdırıwdıń túrleri kóp bolıp, onıń keń tarqalǵan usılı bul graflarda qıdırıw bolıp esaplanadı. Bul algoritmlar kópshilik sistemalardıń tiykarında ámel qıladı. Máselen, GPS sistemasında eń optimal marshruttı anıqlaw hám soǵan uqsas máselelerde qollanıladı. Bul algoritmler jasalma intellekt tarawındaǵı ornı áhmiyetli bolǵanlıǵı sebepli, biz bul algoritmlardı realizaciya qılıwdı usı pitkeriw qánigelik jumısı ushın tema qılıp aldıq.

Usı pitkeriw qanıgelik jumısınıń máqseti bul qıdırıw algoritmlerin Pak-man proektinde realizaciya qılıwdan ibarat.

Bul máqsetke erisiw tómendegi wazıypalar belgilendi:

- Graflarda qıdırıw algoritmlerin úyrenip shıǵıw;

- Pak-man oyunın úyreniw;
- Pak-Man proektin úyreniw;
- Pak-Man proektindegi bólimlerdi úyreniu;
- Proekt bólimlerinde algoritmlerdi realizaciya qılıw;
- Alınğan nátiyjelerdi salıstırıw.

Pitkeriw qángelik jumısı materialları taraw qaysı tártibte úyrenilgen hám jumıslar ámelge asırılğan bolsa sol tártipte keltirilgen. Jumıs kirisiw, 3 bap, juwmaqlaw, qosımsha hám ádebiyatlar diziminen ibarat.

Birinshi bapta tiykarǵı túsinikler úyrenip shıǵıladı. Graflarda qıdırıw algoritmleri, olardıń túrleri, bir-birinen parqı, olardı realizaciya qılıw usılları úyrenip shıǵıladı. Sonday aq pak-man oyını, onıń tarixi, qanday oynalıwı, tártip-qaǵıydaları úyrenip shıǵıladı.

Ekinshi bapta Pak-Man proekti tolıq úyrenip shıǵıladı. Proektiń máqseti, onıń bólimleri, bólimlerdegi tapsırma, máseleleri úyrenip shıǵıladı.

Úshinshi bapta zárúr instrumental qural úyrenip shıǵıladı. Biziń jaǵdayda bul Python programmalaştırıw tili tolıq úyrenip shıǵıladı. Sonday aq, jumıstıń tiykarǵı máqseti esaplanǵan algoritmler Pak-Man proektiniń bólimlerinde realizaciya qılınadı.

Juwmaqlaw bóliminde jumıstı islew barısındaǵı erisilgen nátiyjeler keltiriledi. Qosımshada islep shıǵılğan programma kodları keltiriledi.

I-BAP. TIYKARGÍ TÚSINIKLER

1.1 Graflarda qıdırıw algoritmleri

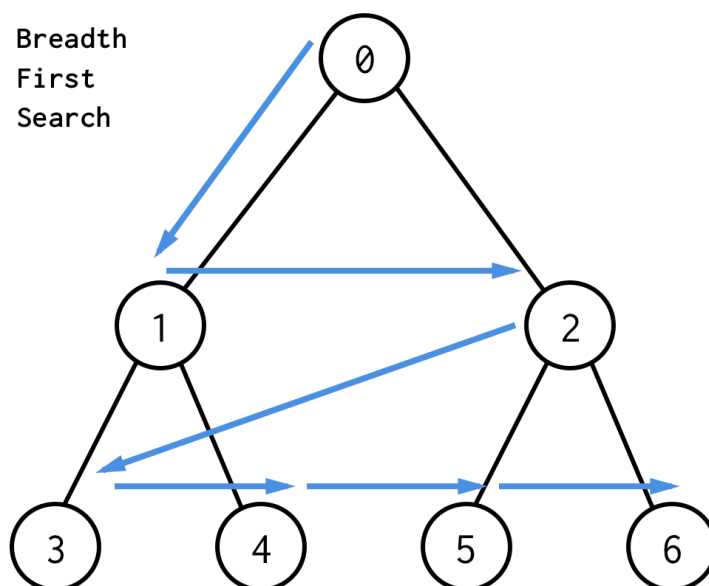
Belgili bir másele qoyılǵandan soń onıń sheshim tabıw máselesi turadı. Bul máqsetke jetisiu ushın jaǵdaylar ortalıǵında qıdırıw usılları qollanıladı. Bizin úyrenip atırǵan tarawımız sonday qıdırıw algoritmleriniń realizaciyasına baylanıslı. Bunday qıdırıw algoritmleri qıdırıw tereginde qıdırıwdı ámelge asıradı. Terekte dástlepki jaǵday belgilenedi hám funkciya járdeminde onıń keyingi uzelleri qarap shıǵıladı. Bunday qıdırıw graflarda qıdırıw dep ataladı.

Qıdırıw tereginiń tuyinleri uzeller dep ataladı. Olardan keyingi turatuǵın uzeller qabıl qılıwshı uzeller dep ataladı. Bunday terekte sheshim bul uzellerden biri bolıwı múmkin. Yamasa bir terekte bir neshe sheshim uzeller bolıwı múmkin. Qıdırıwdan máqset sol sheshim uzelin tabıw.

Uzeller maǵlıwmatlar strukturası bolıp esaplanadı. Al jaǵday bul ortalıqtıń konfiguraciyaları. Qıdırıw teregi processde shólkemlestirilip barıladı, yaǵnıy uzelleri process dawamında kóbeyip baradı. Payda bolǵan uzeller toparı, biraq eli qarap shıǵılmaǵan topar periferiya dep ataladı. Har bir periferiya uzeli keyingi qabıl qılıwshı uzellerge iye bolmaydı. Sheshimdi qıdırıw dawamında uzeller bir shetten qarap shıǵıladı. Eger sheshim tabılmasa onda uzeldiń qabıl qılıwshı uzelleri qarap shıǵıladı. Qıdırıw sol tártipte dawam ettiriledi. Bul usıl sheshimge alıp baradı, biraq kóp waqıt, kóp resurs talap etiwı múmkin. Terektiń keńeyiwı qıdırıw effektivligi túsiredi. Sol sebepli házirgi kunde bunday qıdırıwdıń optimallastırılǵan usılları islep shıǵılǵan.

Qıdırıw usılları tiykarınan eki toparǵa bólinedi. Xabarsız qıdırıw hám habarlı qıdırıw. Xabarsız qıdırıwda jaǵdaylar haqqında qosımsha maǵlıwmatlar esapqa alınbaydı. Tek ǵana másele qoyılıwında belgilengen maǵlıwmatlar esapqa alınadı. Xabarlı qıdırıwda bolsa jaǵdaylar haqqında qosımsha maǵlıwmatlar esapqa alınıp sheshim qabıl etiledi. Bunda kóp sheshimlerdiń ishind optimal esaplangan sheshim qıdırıladı. Xabarsız hám xabarlı qıdırıw usıllarınıń bir neshe túrleri bar.

Enine qıdırıw algoritmi (Breadth First Search, BFS). Bul ápiwayı usıl bolıp dástlep tiykarǵı uzal qarap shıǵıladı, keyin onıń barlıq qabil qılıwshı uzelleri qarap shıǵıladı. Keyin bulardıń qabil qılıwshı uzelleri qarap shıǵıladı. Hám sol tártipte sheshim tabılǵansha dawam etedi. Basqa kılıp aytqanda uzeller terek dárejeleri boyınsha qarap shıǵıladı (1.1-suwret).



1.1-suwret. BFS qıdırıw usılınıń islew principini

1.1-suwretde qıdırıw teregi kórsetilgen. Terek 7 uzelden ibarat. Qıdırıw baslanǵanda tiykarǵı uzal 0 qarap shıǵıladı. Keyin onıń qabil qılıwshı uzeli 1 hám 2 qarap shıǵıladı. Eger sheshim tabılmasa onda 2 dárejege ótiledi hám 3, 4, 5, 6 uzeller bir shetten qarap shıǵıladı. Bul dárejede da sheshim tabılmasa onda qıdırıw keyingi dárejege ótedi.

BFS algoritmi qıdırıw tereginde FIFO (First-in-First-out) principinde ámelge asırılıwı múmkin. Bul algoritmniń optimallıǵın anıqlaw ushın onıń qansha waqıt alıwı hám qansha yaddı iyeleytuǵınlıǵın esaplaw kerek. Maselen, terektegi hár bir uzal b qabil qılıwshı uzelden turadı. Tiykarǵı uzal b muǵdarda uzeldi jaratadı. Terekniń birinshi dárejesi b uzelden turadı. Olardıń hár biri óz aldına jáne b uzal jaratadı. Demek terekniń ekinshi dárejesinde b^2 uzal bar. Ushinchi dárejede bolsa b^3 uzal bar. Qıdırılıp atırǵan sheshim uzeli d terenlikte jaylasqan. d darejede eń aqırǵı

uzelden basqa barlıq uzeller qarap shıǵıladı dep esaplaymız. Onda $d+1$ dárejede uzeller sanı $b^{d+1}-b$ boladı. Onda barlıq uzellerdin sanı:

$$b + b^2 + b^3 + \dots + b^d + (b^{d+1} - b) = O(b^{d+1})$$

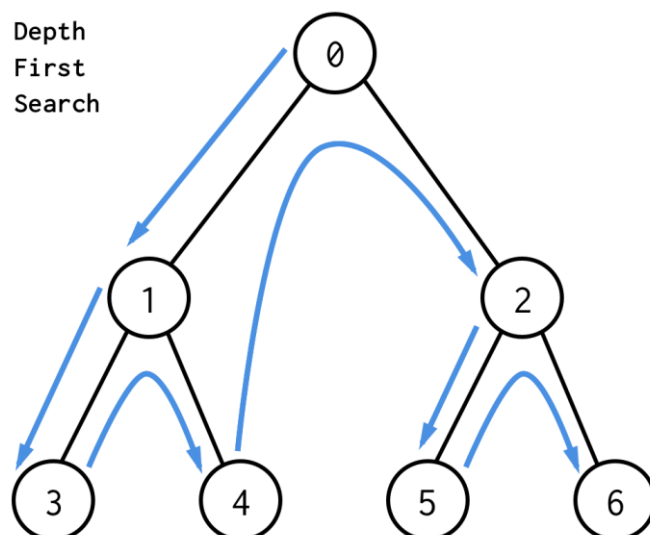
Hár bir qarap shıǵılǵan uzel yadda saqlanıwı shárt, sebebi sheshim tabılmaǵan jaǵdayda qıdırıw sol uzelleridin qabıl qılıwshı uzellerine ótiwi kerek. Sol sebepli waqıt mashqalası menen teńdey yad mashqalasında esapqa alınıw kerek.

Narx kriteriyası boyınsha qıdırıw (Uniform-Cost Search, UCS). Enine qıdırıw algoritmi optimal esaplanıwı múmkin, eger onıń barlıq jaǵdaylar narxı teńdey bolsa. Eger barlıq waqıt bunday bolmaydı. Ápiwayı qosımsha arqalı optimal esaplanatuǵın algoritm jaratıw múmkin. Narx kriteriyası boyınsha qıdırıwda eń joqarǵı uzel qarap shıǵılıwınıń ornına narxı eń jaqsı uzel qarap shıǵıladı. Eger bul algoritmda narxlar barlıǵı birdey bolsa, onda bul algoritm enine qıdırıw algoritmine aylanadı.

UCS qıdırıwda jaǵdaylar sanı emes al olardıń ulıwma narx summası esapqa alınadı. Bul algoritmda qıdırıw tereginde háreket jaǵdaydın narxı arqalı baǵdarlanadı. Sol sebepli onıń waqıt hám yad quramalıǵı b xám d arqalı esaplanbaydı.

Tereńlikke qıdırıw (Depth-First Search, DFS). Bul qıdırıw usılında eń birinshi tiykarǵı uzel qarap shıǵıladı. Keyin onıń qabıl qılıwshı uzeli, keyin onıń qabıl qılıwshı uzel hám sol tártipte qabıl qılıwshı uzel bolmaǵansha dawam ettiriledi. Keyin bir dáreje izge qaytıp qonısı uzeller sol tártipte qarap shıǵıladı.

1.2-suwretde qıdırıw teregi keltirilgen. Bunda qıdırıw tiykarǵı uzel 0 den baslanadı. Keyin onıń qabıl qılıǵıshı 1 uzel qaraladı. Eger sheshim tabılmasa qıdırıw 1 uzeldin qabıl qılıǵıshı 3 uzelve ótedi. Eger qarap shıǵılǵan uzeldin qabıl qılıwshı uzeli bolmasa onda bir dáreje izge qaytıladı. Bul jaǵdayda 3 uzelve qaytıp keledi. Demek onıń eki qabıl qılıwshı uzeli bar, 3 hám 4. 3 uzel qaraldı demek 4 uzel qaraladı. Sol tártipte qıdırıw dawam ettiriledi.



1.2-suwret. DFS algoritminiń islew principi.

Bul algoritm qıdırıw tereginde LIFO (Last-in-First-out) principinde ámelge asırılıwı múmkin. Bul principtegi vektor stek dep ataladı. Kóbinese bul algoritm rekursiya arqalı ámelge asırıladı.

Tereńligi shegaralanǵan qıdırıw. Bul qıdırıw usılı sheksiz terenlik mashqalasınıń sheshim sıpatında xizmet etedi. Bunda qıdırıw tereginiń 1 dárejesine shegara qoyıladı. Eger 1 dárejeden keyingi dárejeler bar bolsada, qıdırıw tek sol dárejege dawam ettiriledi. Bıraq bul jaǵdayda sheshim tabılmay qalıwı múmkin, eger $l < d$ bolsa.

A* qıdırıw usılı. Bul eń birinshi eń jaqsı kıdırıw usılıń bir túri bolıp esaplanadı. Onda uzeller bahalanadı. Bahalaw tómendegishe:

$$f(n) = g(n) + h(n)$$

bul jerde, $g(n)$ bul jerde házirgi uzelge shekemgi narx, $h(n)$ házirgi uzelden máqset uzelge shekemgi narx. Demek $f(n)$ bul n uzelineń ótiwshi eń jaqsı sheshimge alıp barıwshı joldı bahalaw. Máselen transpot máselesinde, qıdırıwda eń birinshi bolıp $g(n) + h(n)$ mánisi eń kishi bolǵan uzal qaraq shıǵıladı. Eger evristikalıq funkciya $h(n)$ talapa tolıq juwap beretuǵın bolsa onda A* algoritmi eń tolıq hám optimal sheshimge alıp baradı.

1.2 Pakman oyını haqqında

Pac-man – akkada janrındağı video oyın bolıp yapon kompaniyası Namco tárepinen 1980 jilda islep shıǵılaǵan. Oyın oynawda oyınshınıń wazıypası Pakmandı basqarıw, hám labirinttaǵı barlıq noqatlardı jep shıǵıwdan ibarat. Oyın waqtında oyınshıǵa qarsılıq qılıwshı arwahlarda bar. Pakman olar menen labirintta ushrasıp qalıwdan qashıwı tiyis, kerı jaǵdayda oyınshı utıladı. Oyın dawamında arwahlar pakmandı quwıp júredi. Arwahlar aldından jazılǵan jasalma intellekt arqalı basqarıladı. Oyın bir neshe darejelerden turadı. Bir oyın utılǵanda keyingi darejege ótip barıladı. Dárejele osiwi menen labirint forması ózgermeydi al tek ǵana quramalılıw ósip baradı. Arwahlardıń háreket etiwi jaqsılanıp, oyın quramalılıǵı artıp baradı. Oyında 256 dáreje belgilengen. Oyın kórinisi 1.3 suwretde kórsetilgen.

1970 jil aqırında Space Invaders oyınıń jetiskenligi nátiyjesinde video oyınlar bazarı arkada oyınların islep shıǵıwǵa kirisedi. Bul oyınlar kóbinese er ballardan ibarat auditoriyanı ózine tartadı. Namco kompaniyası jumısshısı Toru Iwatani tek ǵana er ballar ushın emes al qız ballar auditoriyaları ushında oyın islep shıǵıwǵa qarar etedi. Sol sebepli tiykarǵı tema sıpatında miwelerdi aladı. Bul islep shıǵılǵan oyın Yaponiya Puck-Man dep ataladı. Keyin oyın AQShda tarqalǵannan soń Midway kompaniyası oyında Pac-Man dep ataydı.

Oyın shıqqandan soń Yaponiyada jaqsı nátiyjeler keltiredi, biraq eń tanıqlı bola almaydı. AQShda oyın qız ballardan auditoriyanı ózine tartadı. Oyın jası hár qıylı, tarawı hár qıylı auditoriyalarda keń tarqaladı. Bul oyın AQShda tanıqlı oyınlardın birewine aylanadı.

Oyınıń tanıqlı bolıwı menen ol hár qıylı platformalarǵa ótkeriledi. Máselen sol waqıtları keń tarqalǵan Atari 2600 platforması. Pakman oyını seriya oyınlarınıń bası bolıp, keyinshelik arkada avtomatları, uy oyın sistemaları, kompyuterler hám mobil telefonlar ushın versiyaları islep shıǵıladı.

Oyın ekranın tolıq labirint iyeleydi, al onıń koridorları noqatlar menen toldırılǵan (1.3-suwret). Oyınshınıń wazıypası Pakmandı basqarıw arqalı labirinttegi barlıq noqatlardı jep shıǵıw. Oyın dawamında arwahlarǵa dush keliwden qashıw

kerek. Har bir oynınıń basında arwahlar labirint ortasında jaylasqan tórtmuyesh xanada bekitilgen boladı hám waqıttıń ótiwi menen olar qoyıp jiberiledi. Eger arwahlar Pakmanǵa tiyse onda Pakman janı ketedi hám olar óziniń baslanǵısh poziciyasına qaytarıladı, biraq erisilgen natiyej saqlap qalınadı. Barlıq noqat jep shıǵılǵandan soń keyingi dárejege ótıledi. Labirint shetinde tunelge kiriwshi eki kiriw ornı bar, olardıń birine kiriw arqalı ekinshi qarama-qarsı táreptegi kiriwden shıǵıwǵa boladı.



1.3 suwret – Pakman oynınıń kórinisi

Labirintte ulıwma 240 kishi noqatlar hám 4 úlken noqatlar belgilengen. Úlken noqatlar enerdjayzerler dep ataladı. Kishi noqatlardı jegende 10 ochko al úlken noqatlardı jegende 50 ochko beriledi. Barlıq noqatlar 2600 ochko alıp keledi. Oynınıń dástlepki dárejelerinde Pakman enerdjayzerdi jese arwahlar kók reńge boyaladı. Bul reńde olar Pakmandı quwıǵa emes al onnan qashıwǵa túsedi. Bul waqıtta Pakman arwahlardı jese boladı. Bul jaǵdayda arwahlar dástlepki ortadaǵı xanaǵa qaytarıladı. Eń birinshi jegen arwah ushın 200 ochko beriledi. Hár bir keyingi jeyilgen arhaw ushın beriletuǵın ochko eki ese ósip baradı: 400, 800, 1600 hám t.b.

Sunday qılıp Pakman bir oyında 12000 ochko jiynawı múmkin. 19-dárejeden baslap arwahlar kóq reńge aylanıwdan toqtaydı hám olardı Pakman jewdiń iláji bolmaydı.

Hár bir dárejede pástki zonada bir raundda eki márte bonuslar payda boladı, olar miyweler dep ataladı (1.3-suwret). Birinshi miywe Pakman 70 noqatdı jegende payda boladı, ekinshisi 170 noqat jeyilgende payda boladı. Bonustı jew arqalı 100 den 5000 ға shekem bolğan aralıқтаǵı ochkonı alıw múmkin. Bonus labirintda 9 sekund ǵana turadı, keyin óship ketedi. Jeyilgen miyweler ekran astında kórsetilip turadı.

21-dárejege shekem Pakman tezligi arwah tezliginen joqarı bolıp turadı. 21-dárejeden keyin arwahlar quwıw rejiminde tezligi Pakmanǵa qaraǵanda artadı. Noqatlar jegende Pakman tezligi 10% ge kemeyedi. Enerdjayzer jeyilgende Pakman tezligi artadı, al arwahlar tezligi kemeyedi.

Oyında eki oynshılı rejim bar. Bul rejimde oynshılar nawbet boyınsha oynaydı, olar ochkoları boyınsha jarıladı. Bir oynshı janı ketkende oyun ekinshi oynshıǵa ótedi.

Arwahlar úsh túrli programmalaştırılǵan rejimde háreket etedi: quwıw (ingl. chase), tarqalıw (ingl. scatter) hám qorqıw (ingl. frightened). Birinshi hám ekinshi rejimde háreket etiw ushın oyun ekranı kvadratlarǵa bólinedi. Bunday bir kvadrat 8x8 ólshemde boladı. Quwıw hám tarqalıw rejiminde arwahlar háreket etiw ushın belgili bir kvadratǵa qaray umtıladı. Tarqalıw rejiminde arwahlar shetke jaqın kvadratqa qaray háreket etedi. Qorqıw rejiminde arwahlar máqsetli noqatqa iye bolmaydı, labirintta háreket etiwı tosınnanlı ámelge asırıladı. Quwıw rejiminde hár bir arwah ushın máqsetli noqat hár qıylı bolıp esaplanadı:

- Qızıl reńdegi arwah Shedou (ingl. Shadow) atı Blinki (ingl. Blinky). Original yaponsha atı Oikake. Quwıw rejiminde bul arwahdıń máqsetli noqatı bul Pakman jaylasqa kvadrat bolıp esaplanadı. Blinki basqa arwahlardan parqı ol jeyilgen noqatlarǵa tiykarlanıp óz tezligin eki esege shekem arttıradı. Eger jeyilmegen noqatlar az bolsa, onda ol tarqalıw rejimindegi máqsetli noqatdı Pakman jaylasqan kvadratǵa ózgartiredi hám eki rejimde quwıwdı baslaydı.

- Ash qızıl reńdegi arwah Spidi (ingl. Speedy) atı Pinki. Original yaponsha atı Matibuse. Tarqalıw rejiminde onıń máqsetli noqatı etip Pakman aldındaǵı tórtinshi kletkanı belgileydi. Máselen Pakman joqarıǵa qaray háreket qılsa onda joqarıǵa qaray tórtinshi kletka.
- Kóq reńdegi arwah Bashful (ingl. Bashful) atı Inki. Original yaponsha atı Kimagure. Eń quramalı tarqalıw algoritmine iye. Ol labirintda sonday bir kesindini aladı. Kesindiniń bir sheti Blinkini, oratsı Pakman aldındaǵı ekinshi kletkanı, al ekinshi shet onıń máqsetli noqatı bolıp esaplanadı. Payda bolǵan máqsetli noqatdı aldın ala aytıw qıyın. Sol sebepli yıondaǵı eń qáwipli arwah Inki bolıp esaplanadı.
- Sarı reńdegi Poki (ingl. Pockey) atı Klayd. Original yaponsha atı Otobokey. Eger Klayd Pakmandan 8 kletka uzaqda bolsa, onda onıń máqsetli noqatı Pakmannıń ózi bolıp esaplanadı.

Arwahlar rejimlerdeń birinde háreket etedi, olar háreket etiw baǵdarı tek ǵana labirintda burılıs waqtında ózgertiwi múmkin. Biraq bir rejimnen ekinshine ótken waqıtta olar háreket etiw baǵdarı kerı baǵdarǵa ózgertiwi múmkin. Oyun basında arwahlar ortadaǵı xana shıqqan waqıtta tarqalıw rejiminde háreket etedi. Keyinshelik rejim quwıw rejimine ótkeriledi. Bir raund dauamında arwahlar tarqalıw rejiminde úsh márte hareket etiwı múmkin. Eger Pakman janı ketse bul esaplaǵısh nolge túsiriledi. Oyun dárejesine baylanıslı bul rejim 7 sekundǵa shekem waqıt aralıǵıda bolıwı múmkin. Quwıw rejimi waqıt oyun dárejelerine baylanıslı belgilenedi. 30-dárejeden baslap bul waqıt 1037 sekundǵa shekem bolıwı múmkin. Barlıq tarqalıw rejimleri isleptilip bolǵannan soń arwahlar tek quwıw rejiminde háreket etedi.

Arwahlar oyun basında tómendegi tártipte hareketdi baslaydı.

1. Blinki
2. Pinki
3. Inki
4. Klayd.

Eń birinshi úsh dárejede arwahalar Pakman belgili muǵdardaǵı noqattı jegenen soń xanadan shıǵadı. Máselen Pinki 30 noqat jeyilgende, Inki 60 noqat jeyilgende háreketti baslaydı. Tórtinshi dárejeden baslap arwahlar oyın baslanıwı menen háreketti baslaydı.

II BAP. PAKMAN PROEKTI HAQQÍNDÁ ULÍWMA TÚSINIKLER

2.1 Pak-Man proekti haqqında maǵlıwmatlar

Pac-Man proekti Berkli Kaliforniya universitetinde jasalma intellekt kursı ushın islep shıǵılǵan. Bul proektde Pak-Man oyınıń oynaw usılları qollanılǵan. Proekt video oynılar ushın jasalma intellekt islep shıǵıwǵa arnalmaǵan. Al kurs dawamında jasalma intellekt tarawın effektiv úyretiw ushın arnalǵan. Yaǵnıy ortalıqta qıdırıw, itimallıw sheshim qabil etiw hám oqıtıw baǵdarların proektte ámelge asırıw arqalıw oqıtıw. Bul koncepciya real ámeliy tarawlardıń tiykarında jatadı, máselen tabiy tildi qayta islew, kompyuter kóriwi hám robototexnika.

Proekt úsh twrli máqsetke baǵdarlanǵan. Proektde studentler ózleriniń jumısları nátiyjelerin vizual kóriniste kóriwi múmkin. Pak-Man proekti quramalı mashqalalı ortalıqtı jaratıp beredi, sebebi jasalma intellektiń real mashqalalalı quramala bolıp esaplanadı.

Pak-Man proekti bir neshe bólimlerden turadı. Olar menen óz aldına islesiw múmkin. Olar:

- Qıdırıw (Search)
- Multi-agentli ortalıqta qıdırıw (Multi-Aget Search)
- Oqıtıw (Learning)
- Arwahlardı áwlaw (Ghostbusters)
- Klassifikaciya (Classification)
- Flagdı alıw boyınsha jarıs (Pacman capture the Flag).

Pak-Man proekti Piton tilinde jazılǵan. Onıń kodı ashıq bolıp, proektiń veb saytında jaylasıtırılǵan. Qálewshiler odan paydalanıwı múmkin. Proekt menen islesiwde Piton tili boyınsha bilimler talap etiledi. Piton tili házirgi künde keń tarqalǵan til bolıp, ásirese jasalma intellekt tarawında ken qollanıladı. Jasalma intellekt programmaların jazıwda bul til bir qansha artıqmashlıqlarǵa iye.

Proekt bólimlerinde hár birinde belgilengen fayllar keltirilgen. Ondaǵı ayırım funkciyalar bos qaldırılǵan bolıp, studentlerden sol orınlarǵa funkciya jazıw talap

etiledi. Funkciya jazıp bolǵandan soń onı proektı teekserip kóriwi hám nátiyjeni vizual kóriwi múmkin. Bul fayllar dizimi tomende keltirilgen.

- **Qıdırıw:** search.py, searchAgents.py
- **Multi-agentli ortalıqta qıdırıw:** multiAgents.py
- **Oqıtıw:** valueIterationAgents.py, qlearningAgents.py, analysis.py
- **Arwahlardı awlaw:** bustersAgents.py, inference.py
- **Klassifikaciya:** perceptron.py, perceptron_pacman.py, mira.py, dataClassifier.py, answers.py
- **Flagdı alıw boyınsha jarıs:** myTeam.py

2.2 Pak-Man proektinde qıdırıw bólimi

Qıdırıw bóliminde Pakman agentine labirinttegi joldı tabıw shárti qoyıladı. Bul jol arqalı ol labirinttegi noqatdı jewi kerek. Házirgi künde keń tarqalǵan graflarda qıdırıw algoritmlerin bul proektı qollaw hám natijeni vizual kóriniste kóriw múmkin. Qıdırıw bóliminde bir neshe máseleler belgilengen. Paydalanıwshı usı máselelerdi sheshiw arqalı proektta isleydi. Proektte bir neshe túrli labirintlar belgilengen. Olardı tiyisli buyırıqlar arqalı shaqırıw múmkin.

Proekt tolıq Piton programmalaştırıw tilide jazılǵan. Aldıńǵı bólimlerde aytıp ótilgendey proektin bul bóliminde da redaktorlanıw tiyis bolǵan bir neshe fayllar belgilengen. Proekttegi eń zárúr fayllar dizimi 2.1-keste de kórsetilgen.

2.1-keste. Proekt faylları dizimi

№	Fayl atı	Tárip
Redaktorlanıwı kerek bolǵan fayllar		
1.	search.py	Barlıq qıdırıw algoritmleri jaylasqan.
2.	searchAgents.py	Barlıq qıdırıw agentleri jaylasqan.
Qosımsha maǵlıwmat ushın fayllar		
3.	pacman.py	Pakman oyının iske túsiriwshi bas fayl.

4.	game.py	Pakman oyunınıń qanday islewi haqqında maǵlıwmatlar keltirilgen. Fayl bir neshe járdemshi tiplerdi táripleydi: Agent jaylasqan ornı (AgentState), Agent, Baǵdar hám labirint.
5.	util.py	Qıdırıw algoritmlerin ámelge asırıw ushın járdemshi maǵlıwmatlar strukturaları.
6.	graphicDisplay.py	Pakman oyını ushın grafikalıq ekran.
7.	graphicUtils.py	Pakman oyını grafikası ushın járdemshi funkciyalar.
8.	textDisplay.py	Pakman oyını ushın ASCII grafikası.
9.	ghostAgents.py	Arwahlardı basqarıwshı agent.
10.	keyboardAgents.py	Pakmandı basqarıw ushın klaviatura interfeysi.
11.	layout.py	Maket faylları

Proektiń bul bóliminde menen islesiwde search.py xám searchAgent.py faylları redaktorlanadı. Basqa fayllardı redaktorlaw mumkin emes, bul proektin shala islewine alıp keliwi múmkin. Basqa fayllar maǵlıwmat sıpatında isletiliwi múmkin.

Proekt faylların kompyuterge ornatıp bolǵandan son onı iske túsiriw tiyisli buyırıqlar arqalı ámelge asırıladı. Har bir wazıpanıń ózine tiyisli buyırıqları belgilengen. Buyırıqlar buyırıqlar qatarı arqalı orınlanadı.

Proektti iske túsiriw ushın tómendegi buyırıq orınlanadı:

```
python pacman.py
```

Pakman oyını labirinttan turadı, labirint koridorlan ibarat bolıp háreket tórt tárepge qaray ámelge asırıladı. searchAgent.py faylındaǵı eń apiwayı agent bul GoWestAgent bolıp esaplanadı. Ol barlıq waqıt tek ǵana batıs tárepge qaray háreket qıladı. Bul agent tekǵana testMaze labirintında utıwı múmkin. Bunı tómendegi buyırıq arqalı orınlaw múmkin.

```
python pacman.py --layout testMaze --pacman GoWestAgent
```


Bul jerde python pacman.py arqalı oyın shaqırıladı. layout formatı arqalı labirint belgilenedi. Bul jaǵdayda testMaze labirinti belgilenen. pacman buyırǵı arqalı agent belgilenedi, al bul jaǵdayda bolsa GoWestAgent agenti shaqırılǵan. Nátiyjeni 2.2 suwretde kóriw múmkin.



2.2 – suwret. testMaze labirintında pakmannıń háreket etiwı.

Eger pakman basqa túrli labirintlerde irkilip yamasa toqtap qalatuǵın bolsa, onda oyındı CTRL+C arqalı toqtatıw múmkin. Agentler tek ǵana testMaze labirintin emes al basqa labirintlerdida utıwı talap etiledi.

Proekttegi barlıq múmkin buyırılar izbe-izligi proektin commands.txt faylında keltirilgen. Ol buyırılar tómende keltirilgen.

- python pacman.py
- python pacman.py --layout testMaze --pacman GoWestAgent
- python pacman.py --layout tinyMaze --pacman GoWestAgent
- python pacman.py -h
- python pacman.py -l tinyMaze -p SearchAgent -a fn=tinyMazeSearch
- python pacman.py -l tinyMaze -p SearchAgent
- python pacman.py -l mediumMaze -p SearchAgent
- python pacman.py -l bigMaze -z .5 -p SearchAgent
- python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
- python pacman.py -l bigMaze -p SearchAgent -a fn=bfs -z .5
- python eightpuzzle.py

- `python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs`
- `python pacman.py -l mediumDottedMaze -p StayEastSearchAgent`
- `python pacman.py -l mediumScaryMaze -p StayWestSearchAgent`
- `python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic`
- `python pacman.py -l tinyCorners -p SearchAgent -a fn=bfs,prob=CornersProblem`
- `python pacman.py -l mediumCorners -p SearchAgent -a fn=bfs,prob=CornersProblem`
- `python pacman.py -l mediumCorners -p AStarCornersAgent -z 0.5`
- `python pacman.py -l testSearch -p AStarFoodSearchAgent`
- `python pacman.py -l trickySearch -p AStarFoodSearchAgent`
- `python pacman.py -l bigSearch -p ClosestDotSearchAgent -z .5`
- `python pacman.py -l bigSearch -p ApproximateSearchAgent -z .5 -q`

№ 1 - Másele. Tereńlikke qıdırw algoritmi arqalı noqattı tabıw.

`searchAgent.py` faylında `SearchAgent` agenti tolıq realizaciya qılınǵan. Ol pakman oyunında tolıq háreket ete aladı, háreket izbe-izlikde ámelge asırıladı. `search.py` faylında qıdırw algoritmi programmalasırılmaǵan bolıp, onı jazıw birinshi wazıypa bolıp esaplanadı.

`SearchAgent` agentiniń tuwrı islep turǵanlıǵın tekseriw ushın tómendegi buyırıqtı orınlaw múmkin.

```
python pacman.py -l tinyMaze -p SearchAgent -a fn=tinyMazeSearch
```

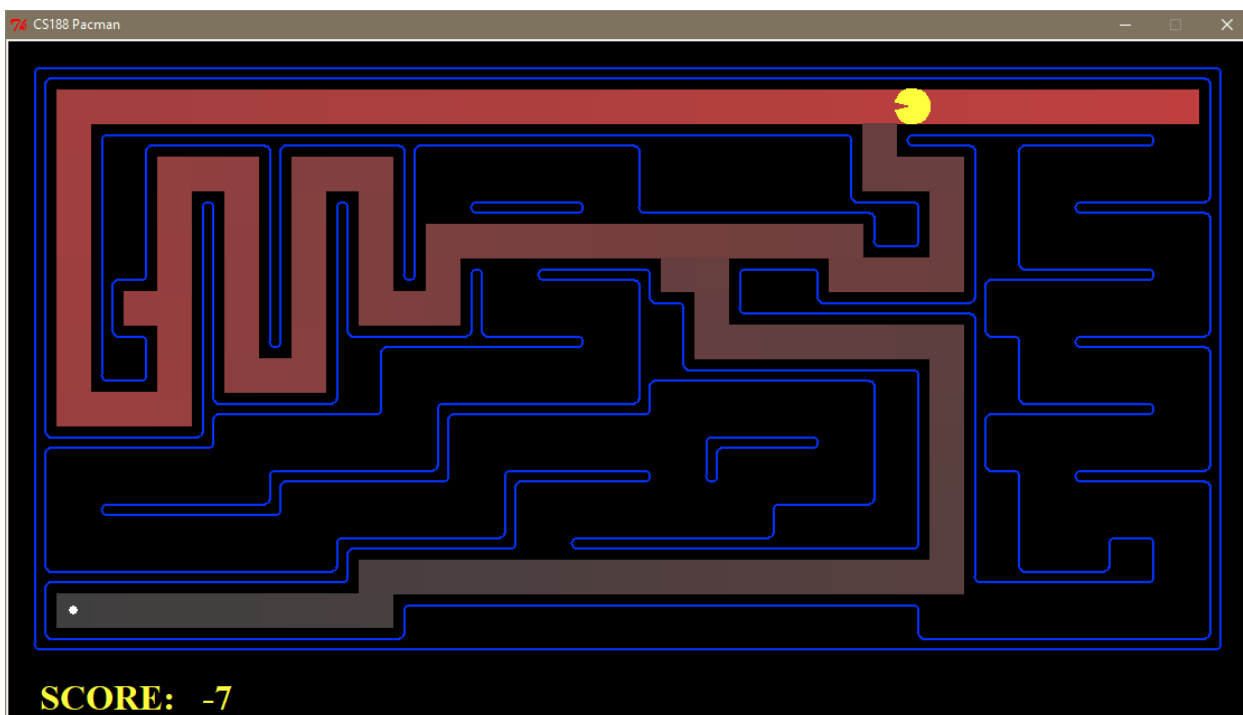
Bul jerde `tinyMazeSearch` funkciyası `SearchAgent` tárepinen orınlanıwı ushın shaqırıladı. Bul algoritm `search.py` faylında jazılǵan.

Barlıq qıdırıw funkciyaları háreketlerdi shıǵıw mánisi sıpatında qaytarıwı talap etiledi. Pakman sol mánisler tiykarında háreket etedi. Bur háreketler tórt tárepgе bolıwı múmkin. Wazıypalardaǵı qıdırıw algoritmleri bir-birine uqsas.

Tereńlikke qıdırıw algoritmi (Depth First Search, DFS) search.py faylında depthFirstSearch funkciyası arqalı orınlanadı. Bul máseleni sheshiw ushın graflarda tereńlikke qıdırıw algoritmin funkciyaǵa jazıw kerek. 1-máseleniń sheshimi tórende buyırıqlarda tolıq orınlanıwı talap etiledi.

- `python pacman.py -l tinyMaze -p SearchAgent`
- `python pacman.py -l mediumMaze -p SearchAgent`
- `python pacman.py -l bigMaze -z .5 -p SearchAgent`

Pakman doskasında qarap shıǵılǵan kletkalar hár qıylı reńde bolıwı múmkin. Jarqınlaw reńdegi kletkalar ertelew qarap shıǵılǵanlıǵın bildiredi (2.3-suwret).



2.3 – suwret. Pakman doskasındaǵı labirinttiń reńleri ózgeriwi

Tereńlikke qıdırıw algoritmin orınlaw ushın tómendegi buyırıq orınlanadı:

```
python pacman.py -l tinyMaze -p SearchAgent -a fn=dfs
```

fn=dfs bul qıdırıw funkciyası sıpatında depthFirstSearch funkciyası shaqırılıwın bildiredi.

№ 2. - Másele. Enine qıdırıw algoritmi.

Enine qıdırıw algoritmi (Breadth First Search, BFS) search.py faylındaǵı breadthFirstSearch funkciyası arqalı ámelge asırıladı. Bul máselelede da graflarda qıdırıw algoritmi bolıp esaplangan enine qıdırıw algoritmin realizaciya qılıw talap etiledi. Qıdırıwdı tekserip kóriw ushın tómendegi buyırıqlar orınlanadı:

```
python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs -z .5
```

№ 3 - Másele. Teńdey-narxlı qıdırıw algoritmi.

Funkciyanıń shıǵıwshı nátiyje narxın ózgeriw arqalı Pakmandı basqa joldan júrgiziw múmkin. Máselen qaterli jollar ushın az narxtı belgilew múmkin. Noqatlar kóp jaylasqan ortalıq ushın bolsa joqarı narxtı belgilew múmkin. Racional pakman agenti sol joldıń narxların esapqa alıp qarar qabıl etiw kerek.

Bul máselede teńdey-narxlı graflarda qıdırıw algoritmin (Uniform-Cost Search, UCS) realizaciya qılıw talap etiledi. Ol search.py faylınıń uniformCostSearch funkcisında ámelge asırıladı. Bul máseleni sheshiwde util.py faylındaǵı maǵlıwmatlar strukturasını qollaw máseleni sheshiwde járdem beredi. Máseleniń tekseriw ushın tómendegi buyırıqlar orınlanadı:

```
python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
python pacman.py -l mediumDottedMaze -p StayEastSearchAgent
python pacman.py -l mediumScaryMaze -p StayWestSearchAgent
```

№ 4 - Másele. A* qıdırıw algoritmi.

Bul máselede A* algoritmi search.py faylınıń aStarSearch funkciyasında orınlaw talap etiledi. A* algoritmi argument sıpatında evristikalıq funkciyadan paydalanadı. Evristik funkciya eki argumentti qabıl etedi, qıdırıw máselesindegi jaǵday hám máseleniń ózi. Máseleni sheshiwde Manxetten uzaqlıǵı evristikasınan paydalanıw múmkin. Onıń ushın tómendegi buyırıq orınlanadı.

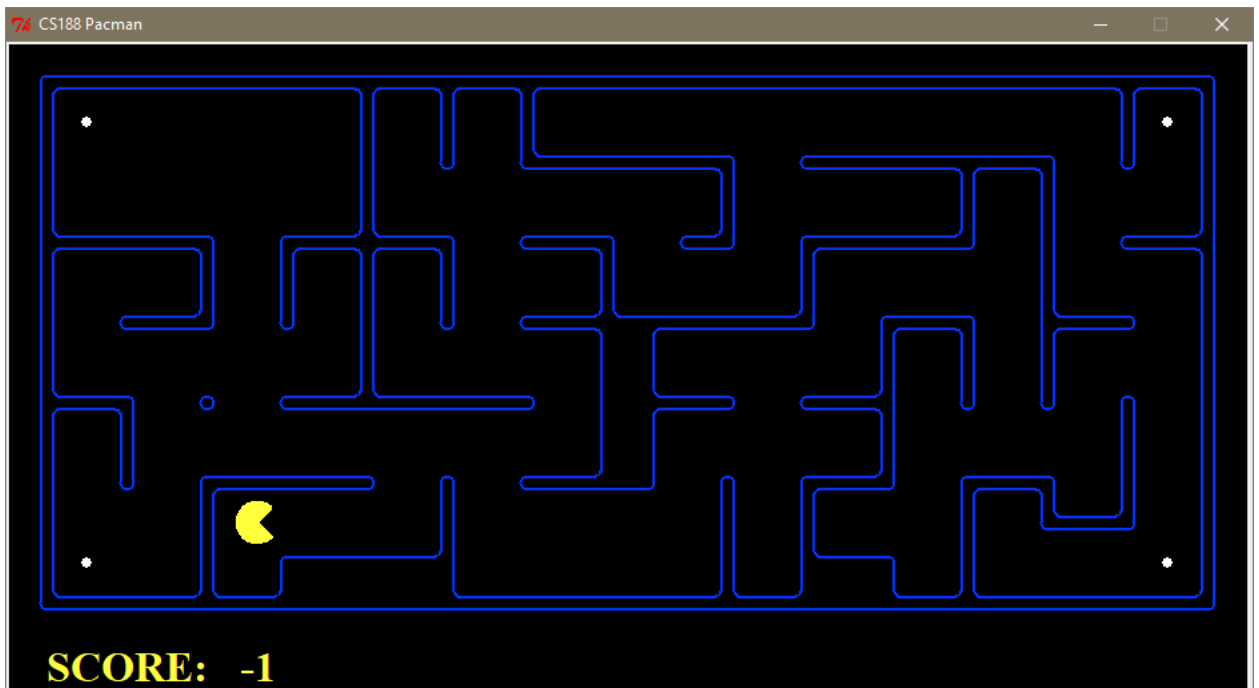
```
python pacman.py -l bigMaze -z .5 -p SearchAgent -a
fn=astar,heuristic=manhattanHeuristic
```

Bul algoritm aldınğı algoritmlerge qarağan optimal joldı tabadı.

№ 5 - Másele. Barlıq múyeshlerdi qarap shıǵıw.

A* algoritminiń haqıqıy effektivligi quramalı máselelerde qollanılǵanda kórinedi. Bul máselede aldınğı ámelge asırılǵan qıdırıw algoritmleri basqa túrli labirinta qollanıladı.

Múyeshler máselesinde labairinte tórt noqat bolıp olar labirinttıń hár muyeshine jaylastırıladı (2.4-suwret). Másele sol tórt noqatdı eń qısqa jol menen jewden ibarat. Máseleni tekseriwde islep shıǵılǵan barlıq qıdırıw algoritmlerin qollanıw hám olardıń nátiyjelerin salıstırıp kóriw múmkin.



2.4 – suwret. Muyeshler máselesi

Muyeshler máselesin sheshiw ushin searchAgents.py faylında ámelge asırıladı. Másele orınlanıwı waqtında barlıq muyeshler qarap basıp ótilii tiyis.

Tekseriw ushin tómendegi buyırıqlar orınlanadı:

```
python pacman.py -l tinyCorners -p SearchAgent -a
fn=astar,prob=CornersProblem

python pacman.py -l mediumCorners -p SearchAgent -a
fn=bfs,prob=CornersProblem
```

Máseleni sheshiwde pakmannıń dáslepki jaǵdayın hám tórt muyesh esapqa alınıwı kerek. BreadFirstSearch algoritmi orınlawda uzeller sanı 2000 ǵa shekem bolıwı múmkin. Al A* algoritmin orınlaǵanda bul muǵdar qısqaradı, yaǵnıy qıdırıw optimallasadı.

№ 6 - Máseler. Barlıq noqatlardı jew.

Bul quramalı máselerdin biri bolıp, pakman bir neshe adımlarda labirinttegi barlıq noqatlardı jep shıǵıwı kerek. Bul searchAgents.py faylında FoodSearchProblem arqalı ámelge asırıladı. Máseleniń sheshimi bul barlıq noqatlardı qarap shıǵıw kerek. Bunı ámel asırıw ushın tómendegi buyırıq orınlanadı:

```
python pacman.py -l testSearch -p AStarFoodSearchAgent
```

№ 7 - Máseler. Barlıq noqatlardı jew.

Bul máseler quramalı, bunda labirinte barlıq kletkalar noqatlar menen toldırılǵan. Barlıq kletkalardı optimal jol arqalı basıp ótiw talap etiledi. Orınlaw searchAgents.py faylındaǵı findPathToClosestDot arqalı ámelge asırıladı.

```
python pacman.py -l bigSearch -p ClosestDotSearchAgent -z .5
```

2.3 Multiagentli ortalıqda qıdırıw

Bul proektiń ekinshi bólimi bolıp bunda máseler ózgeredi. Bul proektde agent ushın algoritm jazıladı, bıralq arwahlar esapqa alınadı. Minimaks hám ekspektimaqs qıdırıw usılları qollanıladı. Proekt ushın óz aldına proekt kodları ajratılǵan, aldınǵı proekt kodlarınan parq qıladı. Proekt faylları dizimi 2.2-kestede kórsetilgen.

2.2-keste. Multiagentli ortalıqda qıdırıw proekti faylları dizimi

№	Fayl atı	Tárip
Redaktorlanıwı kerek bolǵan fayllar		
1.	multiAgents.py	Barlıq multiagentlerdi óz ishine aladı.
Qosımsha maǵlıwmat ushın fayllar		
2.	pacman.py	Pakman oyınınıń iske túsiriwshi bas fayl.

3.	game.py	Pakman oyunınıń qanday islewi haqqında maǵlıwmatlar keltirilgen. Fayl bir neshe járdemshi tiplerdi táripleydi: Agent jaylasqan ornı (AgentState), Agent, Baǵdar hám labirint.
4.	util.py	Qıdırıw algoritmlerin ámelge asırıw ushın járdemshi maǵlıwmatlar strukturaları.
5.	graphicDisplay.py	Pakman oyını ushın grafikalıq ekran.
6.	graphicUtils.py	Pakman oyını grafikası ushın járdemshi funkciyalar.
7.	textDisplay.py	Pakman oyını ushın ASCII grafikası.
8.	ghostAgents.py	Arwahlardı basqarıwshı agent.
9.	keyboardAgents.py	Pakmandı basqarıw ushın klaviatura interfeysi.
10.	layout.py	Maket faylları

Máselelerdi sheshiw waqtında multiAgents.py faylı redaktorlanadı. Basqa fayllardı redaktorlawǵa rusat etilmeydi.

Pakman proektin iske túsiriw buyırǵı tomendegishe:

```
python pacman.py
```

multiAgents.py faylındaǵı ReflexAgent agentin iske túsiriw ushın tómendegi kod jazıladı.

```
python pacman.py -p ReflexAgent
```

№ 1 - Másele. ReflexAgent agenti.

Birinshi másele sheshiwde ReflexAgent agentin jetilistiriw talap etiledi. Agent jaqsı nátiyjege erisiwi ushın arwahlardıń jaylasqan ornı, noqatlardı esapqa alıwı kerek (2.5-suwret). Agent testClassic maketin ansat hám jeńil utıwı kerek.

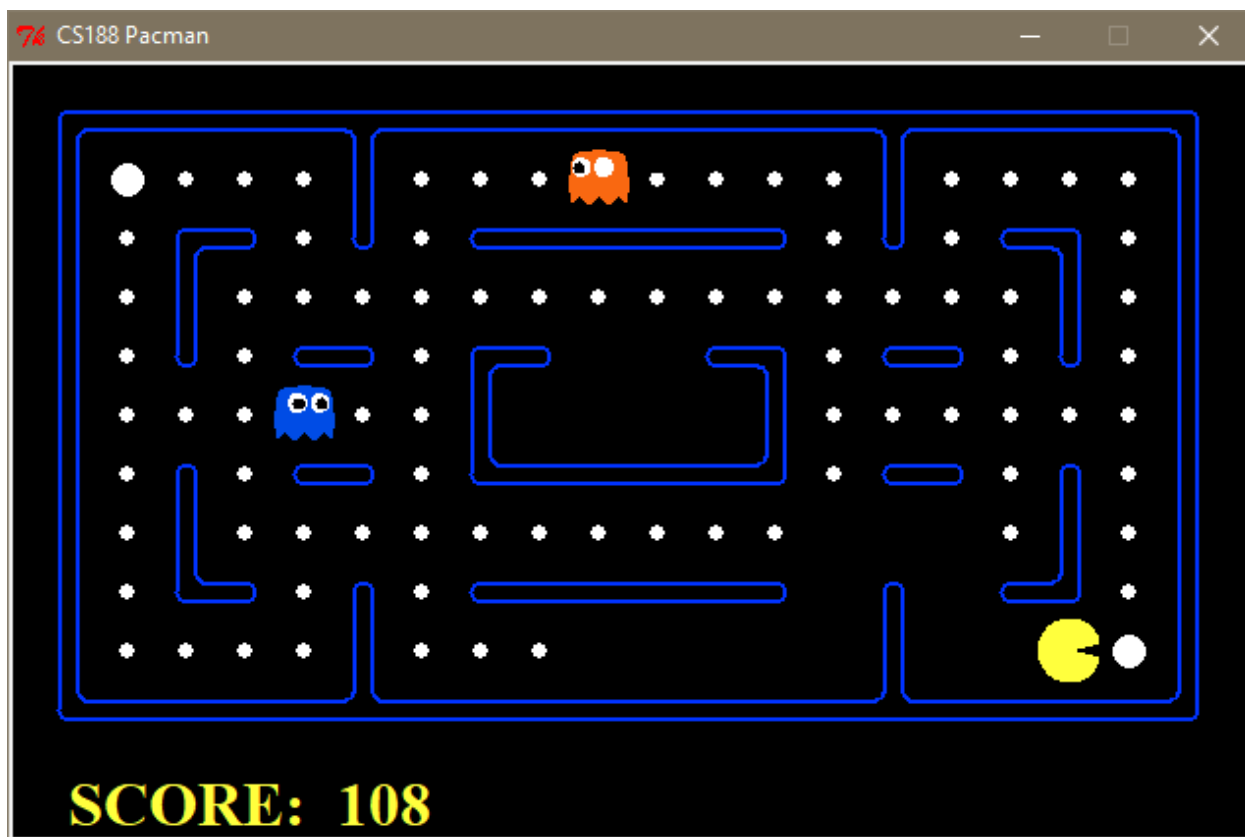
```
python pacman.py -p ReflexAgent -l testClassic
```

Agentti bir yamasa eki arwahdan ibarat labirinta tekserip kóriw ushın tomendegi buyırq isletiledi:

```
python pacman.py --frameTime 0 -p ReflexAgent -k 1
```

```
python pacman.py --frameTime 0 -p ReflexAgent -k 2
```

-k 1 formatı arqalı arwahlardıń muǵdarın belgilewge boladı. Eger jazılǵan funkciya effektiv bolmasa onda eki arwahlı doskada pakman utılıp qalıwı anıq. Sol sebepli eki rejimde da tekserip kóriw kerek. Funkciya ushın zárúr mánislerdi esapqa alıw kerek, máselen noqatqa shekemgi bolǵan aralıq.



2.5 – suwret. Multiagentlı ortalıqtaǵı háreket qılıw

№ 2 - Másele. Minimaks.

Bul máselede multiAgents.py faylındaǵı MinimaxAgent tı programmalaştırıw talap etiledi. Minimaks agent qálegen muǵdardaǵı arwahlardan ibarat ortalıqta háreket qıla alıwı tiyis. Sol sebepli quramalı algoritmdi realizaciya qılıw talap etiledi. Minimaks tereginde hár bir max dárejesi ushın bir neshe min dárejeler (hár bir arwah ushın) belgilenedi.

Jazılǵan programma oyın teregini qálegen tereńlikke shekem keńeytiriw imkaniyatına iye bolıwı talap etiledi. Minimaks teregi uzellerin bahalaw scoreEvaluationFunction arqalı ámelge asırıladi.

Bir qıdırıw dárejesi pakmannıń bir xodı hám barlıq arwahlardıń oǵan juwabın bildiredi. Sol sebepli eki dárejeli tereńliqtegi qıdırıwda pakman hám hár bir arwah eki márte qozǵaladı.

Jazılǵan minimaks agentin bahalaw ushın bir neshe testlerden ótkeriledi. Barlıq testler arqalı tekseriw ushın tómendegi kod jazıladı.

```
python autograder.py -q q2
```

Eger minimaks algoritmi durıs jazılǵan bolsada, ol ayırım testlerdi utılıwı múmkin. Bul qátelik emes. Pakmandı testten ótkeriw bahalaw funkciyası aldından jazılǵan. Bur funkciyanı ózgertiriwge jol qoyılmaydı. Bul agentler kelejektegi jaǵdaylardı bahalaydı, al rekleftor agentı bolsa tek ǵana házirgi jaǵdaydaǵı ámellerdi bahalaydı.

Minimaks agentin iske túsiriw ushın tómendegi buyırıq orınlanadı.

```
python pacman.py -p MinimaxAgent -l minimaxClassic -a depth=4
```

```
python pacman.py -p MinimaxAgent -l trappedClassic -a depth=3
```

bunda minimaxClassic labirintinde 4 tereńlikte minimaks agentı háreket etedi.

Pakman barlıq waqıt 0 shi agent bolıp, basqa agentler nomeri ósiw tártibinde belgilengen. Olardıń háreketi bolsa sol nomerlerge tiykarlanıp ámelge asırıladı. Birinshi 0-agent, keyin 1-agent hám sol tártipte dawam ettiriledi.

Minimaksstıń barlıq jaǵdayları GameStates te bolıwı yamasa getAction arqalı uzatılıwı kerek. Basqa úlken doskalarda mediumClassic xám openClassic da pakman utılıwı azayı múmkin, bıraq jaqsı nátiyje kórsetpeydi. Sebebi onıń nátiyjesiz háreketleri kóbeyedi.

№ 3 - Másele. Alfa-beta kesiliw.

Bul máseledi jańadan agentti ámelge asırıw talap etiledi. Ol minimaks algoritmin effektiv orınlaw ushın alfa-beta kesiliwdi qollawı kerek.

Alfa-beta kesiliwdi qollawdı aldınǵı máseledegi algoritmin tezlesedi. 3 tereńliktegi alfa-beta 2 tereńliktegi minimaks algoritmine qaraǵanda tez islewi

múmkín. Ideal jaǵdayda 3 tereńliktegi smallClassic labirintinde xod bir neshe sekund tezrek islewi kerek.

Máseleni sheshiwde AlphaBetaAgent agentin realizaciya qılıw talap etiledi. Onı iske túsiriw tómendegi buyırıq arqalı ámelge asırıladı.

```
python pacman.py -p AlphaBetaAgent -a depth=3 -l smallClassic
```

AlphaBetaAgent minimaks mánisi minimak MinimaxAgent agentiniń minimaks mánisi menen birdey boladı. Bıraq olardıń ámelleri parq etiwı múmkín.

Bul máseleni sheshiw ushın 2.6-suwretde keltirilgen psevdokoddan paydalanıw múmkín.

Alpha-Beta Implementation

α : MAX's best option on path to root

β : MIN's best option on path to root

```
def max-value(state,  $\alpha$ ,  $\beta$ ):  
    initialize v =  $-\infty$   
    for each successor of state:  
        v = max(v, value(successor,  $\alpha$ ,  $\beta$ ))  
        if v >  $\beta$  return v  
         $\alpha$  = max( $\alpha$ , v)  
    return v
```

```
def min-value(state,  $\alpha$ ,  $\beta$ ):  
    initialize v =  $+\infty$   
    for each successor of state:  
        v = min(v, value(successor,  $\alpha$ ,  $\beta$ ))  
        if v <  $\alpha$  return v  
         $\beta$  = min( $\beta$ , v)  
    return v
```



2.6-suwret. Alfa-beta kesiliw algoritmi psevdokodi

Sheshimdi tekseriw ushın tómendegi buyırıq qollanıladı.

```
python autograder.py -q q3
```

Pakman ayırım testlerde utılıw múmkín, bul qátelikti bildirmeydi.

№ 4 - MásELE. Ekspektimaks.

Eger qarsılas tárep optimal sheshimdi qabıl etetuǵın bolsa onda minimaks hám alfa-beta kesiliw algoritmleri arqalı jaqsı náytijelerge erisiledi. Bul máselede

ExpectimaxAgent agentin realizaciya qılıw talap etiledi. Bunda qarsilas tárep suboptimal sheshimlerdi qabıl etiwi múmkinligi esapqa alınadı.

Bul máseleniń sheshimi qıdırıw hám sheklewlerdi qanaatlandırıw arqalı qıdırıw jaǵdayına uqsas bolıp esaplanadı. Tómenдеgi buyırıq arqalı oynınıń kishi tereklerdi sheshimdi tekserip kóriw múmkin.

```
python autograder.py -q q4
```

Sheshim jaqsı nátiyjege alıp kelse onda onı Pakman oyınında qollap kóriw múmkin. Tosınnanlı arwahlar optimal minimaks sheshim bolıp esaplanbaydı, sol sebepli olardı minimaks qıdırıwı arqalı modellestiriw durıs dep esaplanbaydı. ExpectimaxAgent arwahlardıń barlıq háreketlerine óz waqıtn ketirmeydi, al ol tek ǵana agenttiń háreketlerine tiykarlanadı. Yaǵnıy agent tosınanlı usılda xoddı tanlaydı dep esapqa alınadı.

Bul másele sheshimin tekseriw ushın tómendegi buyırıq qollanıladı.

```
python pacman.py -p ExpectimaxAgent -l minimaxClassic -a depth=3
```

Alfa-beta kesiliw hám Ekspektimaks usılın tómendegi buyırıqlar arqalı salıstırıp kóriw múmkin.

```
python pacman.py -p AlphaBetaAgent -l trappedClassic -a depth=3  
-q -n 10
```

```
python pacman.py -p ExpectimaxAgent -l trappedClassic -a depth=3  
-q -n 10
```

Salıstırıw nátiyjesin sonı anıq múmkin, ekspektimaks algoritmı ayırım testlerdi utadı, al alfa-beta kesiliw usılı bolsa hámme waqıt uttıradı.

№ 5- Másele. Evolyucion funkciya.

Bul máselede betterEvaluationFunction funkciısında eń jaqsı bahalaw funkciyasın jazıw talap etiledi. Bahalaw funkciyası jaǵdaydı bahalawı kerek. Qálegen instrumentlerdi qollaw múmkin. Pakman utqan waqtında 1000 bal alıwı kerek. Testten ótkeriw buyırığı:

```
python autograder.py -q q5
```

autograder agentti smallClassic maketinde 10 márte iske túsiredi. Tómenдеgi tártipte bahalaw funkciyasın ball beriledi:

- Eger avtogreyde waqtı tausılmasdan aldın bir márte utılsa 1 ball beriledi. Bul kriteriyaǵa juwap bermeytuǵın qálegen agent 0 ball aladı;
- Minimum 5 márte utısǵa +1, barlıq 10 utısqa +2 ball;
- 500 den kem bolmaǵan ortasha ochko ushın +1, 1000 dan kem bolmaǵan ochko ushın +2 ball;
- Eger oyın 30 sekunddan az waqt talap etse +1 ball beriledi.

III BAP. QÍDÍRÍW ALGORITMLERIN PROGRAMMALASTÍRÍW

3.1 Python programmalastırıw tili

Python bul joqarı dárejeli programmalastırıw tili bolıp, ulıwma programmalastırıw ushın arnalǵan. Til programmalastırıwshılardıń ónimdarlıǵın asırıw hám koddı oqıwda qolaylılıqtı asırıw máqsetinde isle shıǵılǵan. Python yadrosı sintaksis minimallastırılǵan. Standart biblioteka keń kólemde paydalı funkciyalardı óz ishine aladı.

Python strukturalanǵan, obektke-baǵdarlanǵan, funkcional, imperativ programmalastırıwlardı qollawı múmkin. Onın tiykarǵı qásiyetleri bular dinamikalıq tiplestiriw, yadda basqarıwdı avtomatlastırıw, tolıq introspekciya, kóp aǵımlı esaplawdı qollawı, joqarı dárejeli maǵlıwmatlar strukturası. Programmanı modullerge bóliw, ham olardı jıynaw imkaniyatına iye.

Python tiliniń etalonı bul Cpython interpretatorı, bolım ol bir qansha platformalardı qollap quwatlaydı. Ol Python Software Foundation License ashıq licenziyası arqalı tarqalǵan. Onı qalegen proektlerdi sheklewlersiz qollanıw múmkin.

Python aktiv túrde jetilistirilip atırǵan til bolıp, házirgi künde onın 2, 3 versiyaları keń tarqalǵan. Biziń qarap shıǵıp atırǵa Pak-man proekti sol Python 2.7 versiyasında islep shıǵılǵan.

Python tiliniń artıqmashlıqları:

- Python interpretaciyalanıwshı programmalastırıw tili.
- Dinamikalıq tiplestiriw. Python tilinde aldınnan ózgeriwshı tiplerin jarıyalaw talap etilmeydi.
- Modullilikti qollawı. Modul jazılıp onı basqa programmalarda qollaw múmkin.
- Qatarlarda Unicode dı qollawı múmkin. Programmalastırıw waqtında kirillicadan paydalanıw múmkin.
- Obektge-baǵdarlanǵan programmalastırıwdı qollap quwataydı.

- S/S++ menen intergaciyası.
- Tusinikli, qolaylı sintaksis.
- Keń kólemdegi moduller sanı.
- Krossplatformalılıq.

3.2 Qıdırıw algoritmleri realizaciyası

Pitkeriw qanigelik jumıstın bul bólimnde adınǵı baplarda úyrenilgen qıdırıw algoritmlerin realizaciya qılıwǵa baǵdarlanadı. Aldınǵı baplarda aytıp ótilgende y Pak-man proektinde bir neshe bólimlerge bólingen. Olardıń hár biriniń ózine tiyisli máseleleri hám ozine tiyisli bolǵan proekt faylları bar.

Birinshi bólim máselelerin realizaciya qılıw.

Joqarıda aytıp ótilgende y proektler óz fayllarına iye. Bul qıdırıw bóliminde bizge kerek bolǵan fayllar bular search.py faylı bolıp, onda qıdırıw algoritmi realizaciya kodı jazıladı. Qıdırıw algoritmlarin programmalastırıw ushın maǵlıwmatlar strukturası talap etiledi. Al bul maǵlıwmatlar strukturaları util.py faylında jaylasqan. util.py faylında maǵlıwmatlar strukturaları realizaciya qılınǵan. Tiykarınan úsh túrli maǵlıwmatlar strukturası paydalanıladı. Olar:

- Stek (stack);
- Náwbet (queue);
- Prioritetli náwbet (priority queue).

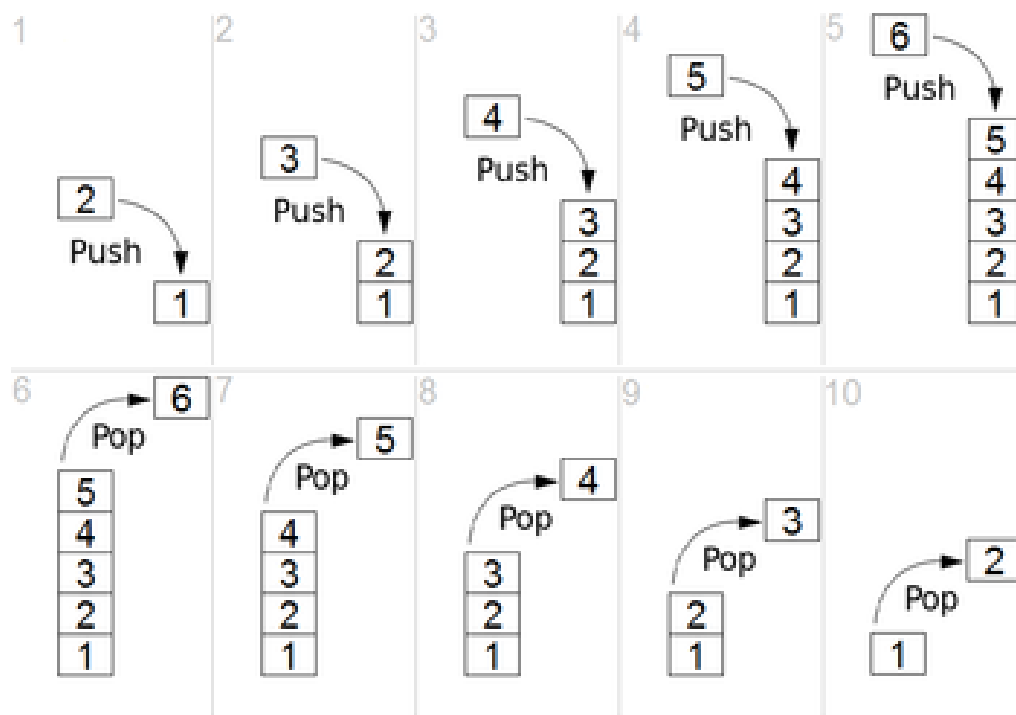
Zárur maǵlıwmatlar strukturası

Stek (stack). Bul abstrakt maǵlıwmatdan strukturası blıp, óz ishine elementlerdi aladı. Elementler LIFO (Last In First Out) principinde shólkemlestiriledi. Yaǵnıy aqırǵı kelgen birinshi shıǵadı. Kóbinese bul strukturanı birinin ustine qoyılǵan tarelkalarǵa uqsatadı. Astındaǵı tarelkanı alıw ushın ustinde turgan tarelkalardı alıp shıǵıw kerek. Bul jerde da sonday princip isleydi.

1946 jılı Alan Turing stek túsinigin kiritedi. Al 1957 jılı Klaus Samelson i Fridrix L. Bauer Turing ideyasına patent aladı.

Ayrım programmalaştırıw tillerinde (Lisp, Python) stek dep qálegen dizimdi aytıwǵa boladı. Sebebi olar ushın pop xám push operaciyaları belgilengen. S++ tilinde bolsa standart biblioteka realizaciya qılınǵan struktura hám metodlarına iye.

Stekde isleu principini 3.1-suwrette kórsetilgen.



3.1-suwret. Stektiń islew principini

Stek ádette bir tárepleme dizim kórinisinde realizaciya boladı. Bundan tısqari ol bir ólshemli massivte sortlangan adreslerge iye bolıwı múmkin. Bunday shólkemlestiriw qolaylı bolıwı múmkin, eger maǵlıwmat elementi anıq bir kólemge iye bolsa. Bul stek elementinde keyingi elementge bolǵan kórsetkishti saqlawǵa hájet qalmaydı.

Pak-man proektında maǵlıwmatlar strukturaları menen islesiw ushın imkaniyat jaratıp berilgen, yaǵnıy stek proektiniń ózinde realizaciya bolǵan. Onıń kodları util.py faylında jazıladı.

```
class Stack:
```

```

"A container with a last-in-first-out (LIFO) queuing
policy."

def __init__(self):
    self.list = []

def push(self, item):
    "Push 'item' onto the stack"
    self.list.append(item)

def pop(self):
    "Pop the most recently pushed item from the stack"
    return self.list.pop()

def isEmpty(self):
    "Returns true if the stack is empty"
    return len(self.list) == 0

```

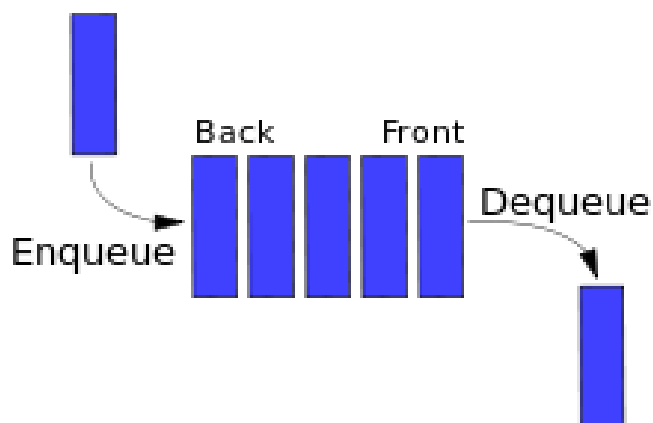
Bul jerde stek menen islesiw ushın tórt turli operaciya kiritilgen, olar:

- `_init_`;
 - `push()`;
 - `pop()`;
 - `isEmpty()`.
- Birinshi operaciya bul konteynerdi vektor kórinisinde ashıw ushın qollanılğan.
 - `push()` operaciyası stekge elementti qosıw ushın isletiledi.
 - `pop()` operaciyası stekdan eń aqırǵı qosılǵan elementti alıw ushın isletiledi.
 - `isEmpty()` bul stekdiń bos yamasa bos emesligi anıqlaw qollanıladı.

Steklar programmalaştırıwda keń qollanıladı, ásirese terekler hám graflarda isleu ushın, yaǵnıy qıdırıw ushın qollanıladı. Bizlerdiń máselede bul stekdan paydalanıw qolay usıl bolıp tabıladı.

Náwbet (queue). Abstrakt maǵlıwmatlar tipi bolıp, elementlerge ruxsat alıw FIFO principinde ámelge asırıladı. FIFO bul First-In, First-Out yaǵnıy birinshi

kelgen, birinshi shıgadı principinde isleydi. Elementti qosıu ádette enqueue dep ataladı. Bunda element náwbettin eń aqırına qosıladı (3.2-suwret).



3.2-suwret. FIFO principini

Al elementti náwbetten alıw náwbettiń eń basınan baslap ámelgen asırıladı.

Náwbetti realizaciya qılıwdıń bir neshe usıllar bar. Bulardan biri bul massiv arqalı realizaciya qılıw. Bunda bir ólshemli massiv q alınadı, hám start, end ózgeriwshileri alınadı. Start ózgeriwshisi náwbettiń basın, end ózgeriwshisi náwbettin aqırın bildiredi. Náwbetke element qosılǵanda q[end], jańa element massivke jazıladı hám end ózgeriwshisi mánisi birge kemeytiriledi (end--). Náwbetten element alıńǵanda q[start] onda start mánisi birge kemeyedi. Sonday qılıp n shi element barlıq waqıt bos boladı.

Ekinshi usıl bul dinamikalıq yad menen islesiw arqalı. Náybet sızıqlı dizim kórinisinde boladı. Al element qosıw hám alıw dizimniń eki shetinen ámelge asırıladı. Bunda usıldıń jaqsı tárepi, bunda dizim uzınlıǵı tek ǵána yad menen sheklenedi.

Joqarıda aytıp ótilgendey Pak-man prektinde da náwbet realizaciya qılıńǵan.

```
class Queue:
    def __init__(self):
        self.list = []

    def push(self, item):
        self.list.insert(0, item)
```

```

def pop(self):
    return self.list.pop()

def isEmpty(self):
    return len(self.list) == 0

```

Bunda queue klassı jarıyalanǵan bolıp, onday bir neshe operaciyalar berilgen. Bul operaciyalar stekdagı operaciyalargá uqsas, biraq islew principi menen parq qıladı.

Prioritetli náwbet (Priority Queue) – bul abstrakt maǵlıwmat tipi bolıp, eki shártli túrde operaciyanı óz ishine aladı. Element qosıw hám maksimumdı (minimumdı) alıw. Náwbettegi hár bir element ushın prioritetlerdi belgilew múmkin. Prioritet mánisi sanlar. Joqarı prioritetli element pást prioritetli elementten aldın turadı.

Tiykargı metodlar bula:

- Insert(gilt, mánisi) – náwbetke gilt hám mánisti qosadı.
- Extract_minimum() – náwbettegi eń kishi prioritetli, yaǵnıy gilt mánisi kishi bolǵan elementtiń mánisi hám giltin náwbetten shıǵaradı.
- Extract_maximum() – aldınǵı metoddıń keri kórinisi bolıp, eń joqarı mánisli prioritetke iye elementti náwbetten shıǵaradı.

Ámelde prioritetli náwbetler menen islesiwde qosımsha operaciyalardan paydalanıladı.

- minimal elementti náwbetten óshirmey shıǵarıp alıw;
- qálegen elementtiń prioritetin ózgartiriw;
- qálegen elementti óshiriw;
- eki náwbetti bir náwbetke biriktiriw.

Indekslengen prioritetli náwbetlerde elementlerge olardıń indeksleri boyınsha murájaat qılıw múmkin. Bunday náwbetler izbe-iz sortlarǵan náwbetlerdi biriktiriw ushın qollanıladı.

Bundan tısqarı eki tárepleme ruxsatlı prioritetli náwbetler (doubе-ended priority queue, DEPQ) qollanılıp kelmekte. Bunday náwbetler bir waqıtta minimal hám maksimal elementlerine ruxsat alıw operaciyalarına iye boladı.

Pak-man proektinde bunday náwbet Priority Queue klassında realizaciya qılınǵan.

```
class PriorityQueue:
    def __init__(self):
        self.heap = []
        self.count = 0

    def push(self, item, priority):
        entry = (priority, self.count, item)
        heapq.heappush(self.heap, entry)
        self.count += 1

    def pop(self):
        (_, _, item) = heapq.heappop(self.heap)
        return item

    def isEmpty(self):
        return len(self.heap) == 0

    def update(self, item, priority):
        for index, (p, c, i) in enumerate(self.heap):
            if i == item:
                if p <= priority:
                    break
                del self.heap[index]
```

```

        self.heap.append((priority, c, item))

        heapq.heapify(self.heap)

        break

    else:

        self.push(item, priority)

```

Bul klassta bir neshe operaciýalar jaryalangan. Element qosıw ushın ádetdegi push() operaciyası. Biraq ol aldınğı operaciyalardan parq etedi, bunda tek gana elementtiń ózi emes al onıń prioritetinde qosıw kerek boladı.

Update() operacisınıń islew principini:

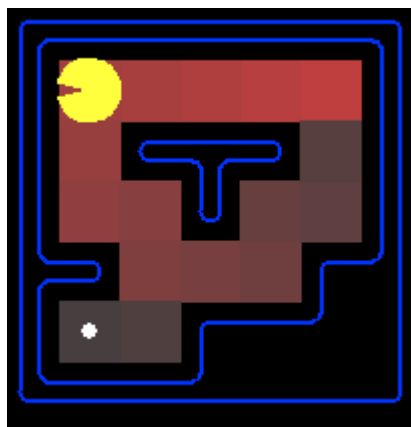
- Eger nábette element joqarı prioritet mene bar bolsa, onda onıń prioriteti jańalanadı hám nábette qaytadan qurıladı.
- Eger nábette element pást prioritetli bolsa, onda hesh nárese orınlanbaydı.
- Eger element ulıwma joq bolsa, push() operaciyası orınlanadı.

Birinshi proekt máseleleri.

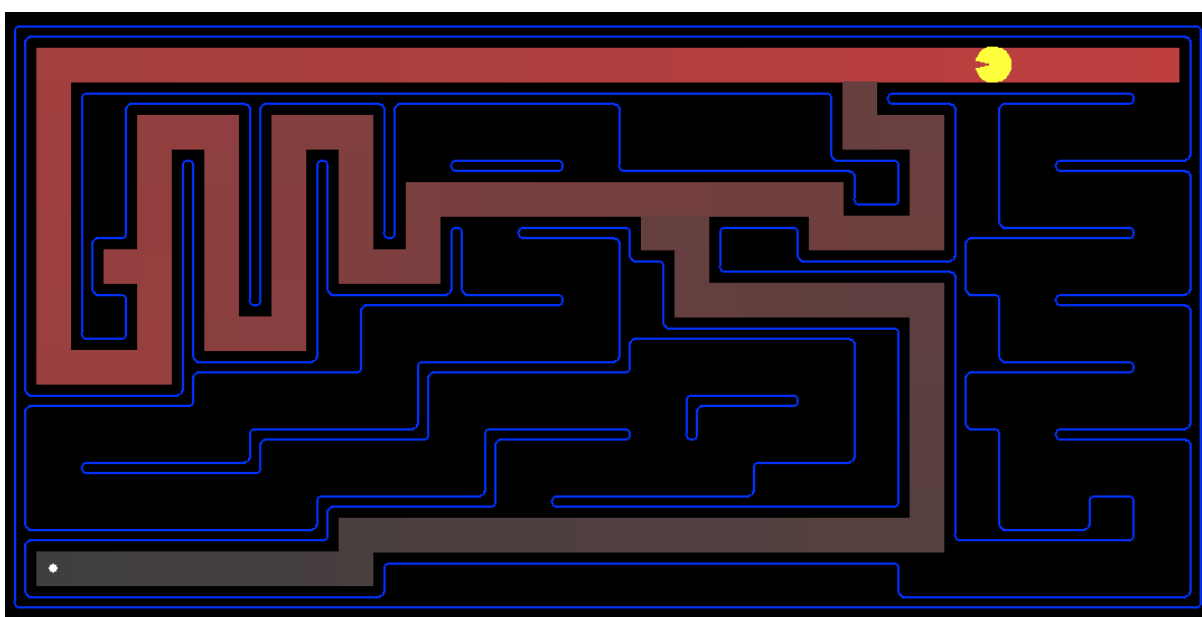
Demek biz aldınğı baplarda Pak-Man proektindegi bólimler, olardagı tapsırmalar yaǵnıy máselelerdi úyrenip shıqtıq. Máseleniń qoyılıw, shártlere hám t.b. Usı bólim sol máselelerdi optimal jol menen sheshiwge arnaladı.

Birinshi bólim bul qıdırıw algoritmların realizaciya qılıwǵa arnalǵan. Máseleniń qoyılıwın wyrenilgende ayırım maǵlıwmatlar anıqlastırıldı. Pakman háreket etetuǵın labirintlerdiń tiykarǵı úsh túri bar. Olar:

- tinyMaze (3.3-suwret);
- mediumMaze (3.4-suwret);
- bigMaze (3.5-suwret).



3.3 – suwret. tinyMaze labirinti.

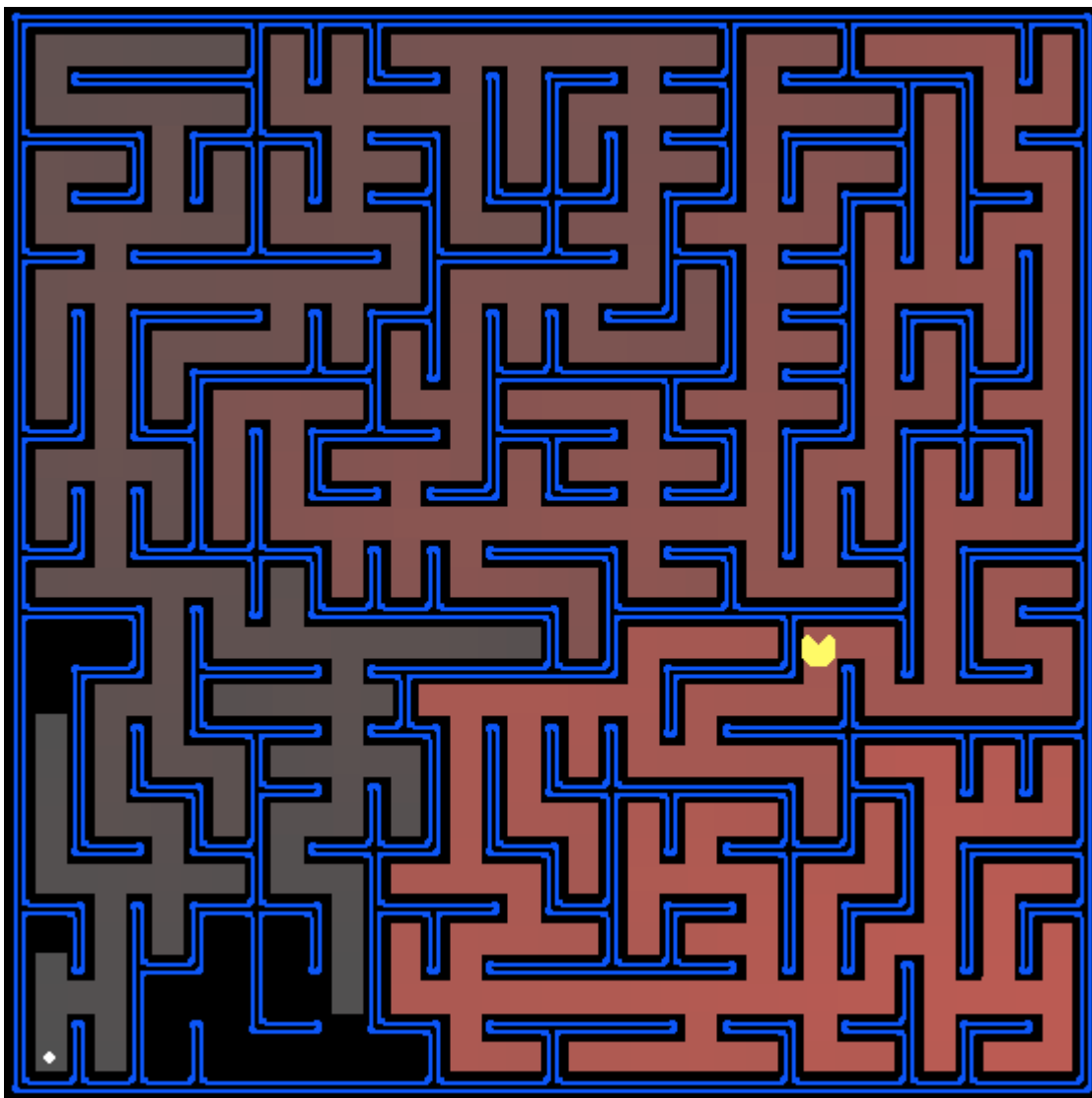


3.4 – suwret. tinyMaze labirinti.

Shárt boyınsha realizaciya kılınğan algoritmlar usı úsh labirinttada islewi talap etiledi. Bundayn tısqarı hár bir funkciyanıń shıǵıwshı mánisi bul dizim kórinisinde bolıwı kerek. Yaǵnıy labirintta juriw jolları dizimi.

Bunıń ushın test funkciya esaplanǵan tinyMazeSearch funkciyasın qarap shıǵamız.

```
def tinyMazeSearch(problem):
    from game import Directions
    s = Directions.SOUTH
    w = Directions.WEST
    return [s,s,w,s,w,w,s,w]
```



3.5 – suwret. tinyMaze labirinti.

bul funksiya tek ǵana tinyMaze labirintında xáreket etiw ushin arnalǵan. Bul jerde tórt tarepten ekiwine SOUTH hám WEST yaǵnıy shepke hám tómenge háreket qılıw ushin ózgeriwshi belgilep shıǵıldı. Sebebi tinyMaze labirintında tek ǵana tómen hám shepke háreket etiw múmkin (3.3-suwret).

```
s = Directions.SOUTH
```

bul jerde Direction baǵdarlar diziminen South baǵdarı ushin ózgeriwshi belgilenedi. Tap sonday ekinshi baǵda ushında ózgeriwshi belgilenedi. Endi funksiyanıń shıǵıwshı mánisinín islep shıǵıw kerek. Bunıń tek ǵana sol baǵdarlar izbe-izligin shıǵarıw jeterli bolıp esaplanadı.

```
return [s, s, w, s, w, w, s, w]
```

bul jerde pakmannıń háreket izbe-izligi kórsetilgen.

Yaǵnıy:

1. shepke qaray bir kletkaǵa háreket,
2. shepke qaray bir kletkaǵa háreket,
3. tómende qaray bir kletkaǵa háreket,
4. shepke qaray bir kletkaǵa háreket,
5. hám t.b.

Basqa máselelerde da tap sonday shıǵıwshı mánis talap etiledi. Bıraq shıǵıwshı izbe-izlikti algoritmnıń ózi islep shıǵıwı kerek.

1-másele. Bul máselede tereńliq boyınsha qıdırıw (DFS) algoritmin tolıq realizaciya qılıw talap etilgen. Bunıń ushın search.py faylında depthFirstSearch() funkciyasın jazıw soraladı. Al funkciya jazıp bolıńandan soń onı iske túsiriw ushın tiyisli buyırıqlar orınlanadı. Joqarıdaǵı úsh labirint ushın óz aldına úsh túrli buyırıq jazıw kerek.

```
python pacman.py -l tinyMaze -p SearchAgent -a fn=dfs
```

```
python pacman.py -l mediumMaze -p SearchAgent -a fn=dfs
```

```
python pacman.py -l bigMaze -p SearchAgent -a fn=dfs
```

Bul buyırıq orınlanǵanda 3.3-suwrettegidey labirintda pakman háreket etip baslaydı. Onıń xáreketi noqatǵa shekem dawam etedi. Noqatqa barılǵanda nátiyjeler jarıyalanadı. Tereńlikke qıdırıw algoritmin Python tilinde islep shıǵıldı:

```
def depthFirstSearch(problem):  
    start = problem.getStartState()  
    c = problem.getStartState()  
    exploredState = []  
    exploredState.append(start)  
    states = util.Stack()  
    stateTuple = (start, [])
```

```

states.push(stateTuple)

while not states.isEmpty() and not problem.isGoalState(c):
    state, actions = states.pop()
    exploredState.append(state)
    successor = problem.getSuccessors(state)
    for i in successor:
        coordinates = i[0]
        if not coordinates in exploredState:
            c = i[0]
            direction = i[1]
            states.push((coordinates, actions +
[direction]))
    return actions + [direction]
util.raiseNotDefined()

```

Bul funkciya úyrenilgen algoritmlardıń islew principleri, alıńǵan bilimler tiykarında islep shıǵıldı.

Funkciyanı jazıwda stek maǵlıwmatlar strukturasınıan paydalanıldı, sebebi stektin islew principini Tereńlikke qıdırıw algoritminiń islew princine tuwrı keledi.

Bul jerde:

- exploredState[] – bul qarap shıǵılǵan kletkalar xaqında maǵlıwmattı ózinde saqlaydı;
- States – bul bas stek;
- Start – bul baslanǵısh kletka;
- State – jaǵday, koordinata

Dáslepki sazlawlar ornatiǵandan soń ciki baslanadı:

```
while not states.isEmpty() and not problem.isGoalState(c):
```

bunda ciki maqsatli kletkaga jetkenshe yamasa stekda element bar bolsa dawam eteberedi.

Maqsatli kletka jetkenlikte tekseriw ushin


```
problem.isGoalState(c)
```

qollaniladı.

Labirintta qıdırıwda eń zárúr bolǵan bul bir kletkadan júriw múmkin bolǵan basqa kletkalar dizimi. Bul maǵlıwmattı alıw ushın tómendegi funkciya qollaniladı.

```
successor = problem.getSuccessors(state)
```

bul jerde successor vektor bolıp, onda barlıq múmkin bolǵan jollar jaylasqan. Bul dizim menen islesiw ushın cikldan paydalanamız:

```
for i in successor:
```

Dizimde neshe element bolsa cikl sonsha márte isleydi. Yaǵnıy dizimdeki barlıq elementler qarap shıǵıladı.

Eger qarap atırılǵan kletka exploredState[] diziminde bolmasa bul kletka qarap shıǵıladı. Eger dizimde bar bolsa onda taslap ketiledi.

Basıp ótilgen jollar actions ózgeriwshinde jiynalıp barıladı. Funkciya qıdırıwdı toqtatǵanda usı izbe-izliq funkciyanıń shıǵıwshı mánisin etip shıǵarıladı.

3.5-suwretde kórinip turǵanıday pakman basıp ótken jol reńi har qıylı. Eger kletka qansha kóp qarap shıǵılsa onıń reni sonsha toq boladı.

Hár bir másele ni isletip qorgende konsolda onıń nátiyjesi jariyalanadı. Sol nátiyjeler arqalı algoritmlardı yamasa sheshimlerde salıstırıw múmkin. 1 máseledeki Tereńlikke qıdırıw algoritminiń realizaciyası nátiyjesi 3.1-kestede kórsetilgen.

3.1-keste. DFS algoritminiń úsh labirintta kórsetken nátiyjesi

№	Labirint	Uzeller	Ball
1.	tinyMaze	15	500
2.	mediummaze	146	380
3.	bigMaze	390	300

2-másele. Bul másele aldınǵı máselege uqsas bolıp, bunda tereńlikke qıdırıw algoritmi ornına enine qıdırıw (BFS) algoritmin realizaciya qılıw kerek. Bul máselede da islep shıǵılǵan algoritm úsh labirintdada jaqsı nátiyje kórsetiwi talap etiledi. Algoritmdı tekserip kóriw buyırıqları tómende keltirilgen.

```
python pacman.py -l tinyMaze -p SearchAgent -a fn=bfs
python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
python pacman.py -l bigMaze -p SearchAgent -a fn=bfs
```

Úyrenip shıǵılǵan materiallar, bilimler tiykarında enine qıdırıw algoritmın (BFS) realizaciya qılındı. Bunıń ushın search.py faylında breadthFirstSearch() funkciyası jazıp shıǵıldı:

```
def breadthFirstSearch(problem):
    start = problem.getStartState()
    exploredState = []
    exploredState.append(start)
    states = util.Queue()
    stateTuple = (start, [])
    states.push(stateTuple)
    while not states.isEmpty():
        state, action = states.pop()
        if problem.isGoalState(state):
            return action
        successor = problem.getSuccessors(state)
        for i in successor:
            coordinates = i[0]
            if not coordinates in exploredState:
                direction = i[1]
                exploredState.append(coordinates)
                states.push((coordinates, action + [direction]))
    return action
util.raiseNotDefined()
```

bul funkciyadı eni boyışa qıdırıwdı ámelge asırıw ushın náwbet (queue) maǵlıwmatlar strukturasınıan paydalanıldı. Funkciya tómendegi vektor hám ózgeriwshilerden turadı.

- exploredState[] – qarap shıǵılǵan kletkalar;
- states – náwbet;
- action – házirge shekem basıp ótilgen jol;
- successor – bul bir kletkadan juriw múmkin bolǵan kletkalar dizimi
- direction – labirinttaǵı házirgi baǵdar.

Biziń islep shıqqan BFS algoritminiń úsh labirintda kórsetken nátiyjelei 3.2-kestede kórsetilgen.

3.2-keste. BFS algoritminiń úsh labirintta kórsetken nátiyjesi

№	Labirint	Uzeller	Ball
1.	tinyMaze	15	502
2.	mediummaze	269	442
3.	bigMaze	620	300

3-másele. Bul máselede Uniform-Cost Search algoritmin realizaciya qılıw talap etilgen. Bul algoritmda prioritetli náwbet maǵlıwmatlar strukturasınıan paydalanıldı. Algoritm uniformCostSearch() funkciyasında realizaciya qılındı.

```
def uniformCostSearch(problem):
    start = problem.getStartState()
    exploredState = []
    states = util.PriorityQueue()
    states.push((start, [], 0))
    while not states.isEmpty():
        state, actions = states.pop()
        if problem.isGoalState(state):
            return actions
        if state not in exploredState:
```

```

successors = problem.getSuccessors(state)

for succ in successors:
    coordinates = succ[0]
    if coordinates not in exploredState:
        directions = succ[1]
        newCost = actions + [directions]
        states.push((coordinates, actions +
[directions]), problem.getCostOfActions(newCost))
    exploredState.append(state)

return actions

util.raiseNotDefined()

```

bul funkciyada da aldınǵı funkciyalardaǵıǵa uqsas ózgeriwshiler paydalanıldı. Funkciyanı tekseriw ushın tómendegishe buyırq jazıladı.

```

python pacman.py -l tinyMaze -p SearchAgent -a fn=ucs
python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
python pacman.py -l bigMaze -p SearchAgent -a fn=ucs

```

Islep shıqqan UCS algoritminiń úsh labirintda kórsetken nátiyjelei 3.3-kestege kórsetilgen.

3.3-keste. UCS algoritminiń úsh labirintta kórsetken nátiyjesi

№	Labirint	Uzeller	Ball
1.	tinyMaze	15	502
2.	mediummaze	269	442
3.	bigMaze	620	300

4-másele. Bul máselede A* qıdırıw usılı járdeminde labirintda noqatǵa optimal jol menen jetip barıw kerek. Bunıń ushın search.py faylında aStarSearc() funkciyasın jazıw kerek. Hám funkciyanı tekseriw ushın aldınǵı buyırıqlarǵa ózgeris kiritiw kerek.

fn=astar

islep shıǵılǵan algoritm tómente kórsetilgen.

```
def aStarSearch(problem, heuristic=nullHeuristic):
    start = problem.getStartState()
    exploredState = []
    states = util.PriorityQueue()
    states.push((start, []), nullHeuristic(start, problem))
    nCost = 0
    while not states.isEmpty():
        state, actions = states.pop()
        if problem.isGoalState(state):
            return actions
        if state not in exploredState:
            successors = problem.getSuccessors(state)
            for succ in successors:
                coordinates = succ[0]
                if coordinates not in exploredState:
                    directions = succ[1]
                    nActions = actions + [directions]
                    nCost = problem.getCostOfActions(nActions) +
                    heuristic(coordinates, problem)
                    states.push((coordinates, actions +
                    [directions]), nCost)
            exploredState.append(state)
    return actions
    util.raiseNotDefined()
```

Bul funkciyanı jazıwda prioritetli náwbetten paydalanıldı. Al qıdırıw ushın bir neshe ózgeriwshiler hám vektorlar qollanıldı. Úsh labirintta alınǵan nátiyjeler 3.4-keste de keltirilgen.

3.4-keste. A* algoritminiń úsh labirintta kórsetken nátiyjesi

№	Labirint	Uzeller	Ball
1.	tinyMaze	15	502
2.	mediummaze	269	442
3.	bigMaze	620	300

Ámelge asırılğan tórt algoritmnıń kórsetken nátiyjelerine sonı aytıwımız múmkin (3.5-keste). DFS xám BFS algoritmleri labirinttıń jaǵdayına qarap hár qıylı nátiyjeler kórsetiwi múmkin. Olardıń biri jaqsı biri jaman nátiyje kórsetdi dep aytalmaymız.

3.5-keste. Algoritmardıń nátiyjelerin salıstırıw

№	Labirint	DFS		BFS		UCS		A*	
		Uzeller	Ball	Uzeller	Ball	Uzeller	Ball	Uzeller	Ball
1.	tinyMaze	15	500	15	502	15	502	15	502
2.	mediummaze	146	380	269	442	269	442	269	442
3.	bigMaze	390	300	620	300	620	300	620	300

Al UCS hám A* algoritmaların salıstıratuǵın bolsaq olarda parq sezilmeydi. Sebebi eki algoritmda da birde maǵlıwmatlar strukturası prioritetli nábwet qollanıldı.

Ulıwma tórt algoritmdı salıstırıp aytatın bolsaq, onda eń jaqsı natıyjeni A* algoritmi kórsetedi dep ayta alamız.

JUWMAQLAW

Piteriw qanigelik jumısınıń máqseti bul graflarda qıdırıw algoritmların Pak-man proektinde realizaciya qılıwdan ibarat edi. Bul máqsetke jetisiw ushın bir neshe wazıypalar belgilendi.

Jumıstı islew barısında tómendegi wazıypalar orınladı:

- Graflarda qıdırıw algoritmlerin úyrenip shıǵıldı;
- Pak-man oyını, onıń tártip qaǵıydaları, oyın dárejeleri olardıń parqları tolıq úyrenip shıǵıldı;
- Pak-Man proekti, máqseti, onıń bólimleri, bólimlerdegi máseleleri, olardı sheshiw ushın kórsetpeler, tolıq úyrenip shıǵıldı;
- Proekt bólimlerindegi máselelerde graflarda qıdırıw algoritmları realizaciya qılındı;
- Realizaciya qılınǵan algoritmlardıń nátiyjeleri salıstırıldı hám juwmaqlandı..

Pitkeriw qanigelik jumısın orınlaw waqtında ilimiy basshı hám ilimiy másláhatshi jaqından járdem berdi.

Jumısta graflarda qıdırıw algoritmları esaplanǵan tereńlikke qıdırıw algoritmi, enine qıdırıw algoritmi, A* qıdırıw algoritmleri hám basqada algoritmler úyrenip shıǵıldı hám pak-man proektinde realizaciya qılındı.

Islep shıǵılǵan programma bul jasalma intellekt tarawına kiriwdegi eń birinshi adımlardan biri boldı. Qıdırıw algoritmların tolıq úyrenip, olardı realizaciya qılıwda tájriybege iye bolǵan halatda jasalma intellekt tarawına kirisiwde jeńillik boladı.

Pitkeriw qanigelik jumısınıń wazıypaları barlıǵı tolıq orınladı, jumıstıń máqsetine erisildi.

PAYDALANILGAN ÁDEBIYATLAR