Intelligent Mountain Rescue

Specifikáció

Egy **Search and Rescue (SAR)** alkalmazást készítünk, melynek színtere egy erdős-hegyes nemzeti park. A szimulációban a bajba jutott kirándulók segítségére mentőhelikopterek illetve szárazföldi mentőegységek sietnek. A felhasználó egy mezőre kattintással egy bajba jutott embert helyezhet le, kinek az élettartama idővel csökken. Ha eléri a nullát már nem lehet segíteni rajta. A GUIn látszódni fog, hogy hány kirándulóból hányat sikerült az ágenseknek megmenteniük. A bajba jutott kiránduló csak várja, hogy megmentsék. A helikopter illetve gyalogos ágensek közös célja együttműködéssel minden (vagy a lehető legtöbb, ha ez nem lehetséges) áldozat megmentése, vagyis megtalálása, stabilizálása és visszavétele a bázisra. Ha egy bajba jutott embert sikeresen megtalál egy mentőcsoport, vagy elfogy az életideje, akkor lekerül a GUIról. A GUI térképhez hasonló, felülnézeti, mezőkből áll. Kétféle mezőtípus létezik: az erdő és a hegy. A hegy mezőkre a helikopter ágensek nem tudnak leszállni, azonban egy időegység alatt 3 lépést tehetnek meg, szemben a gyalogos ágensekkel akik ez idő alatt csak egy mezőnyit haladhatnak. A felhasználó a bajba jutott kirándulók lehelyezésén kívül a szimuláció kezdetekor helikopteres illetve gyalogos mentőegység ágensek számát illetve a szimuláción sebességét is beállíthatja.

A rendszer terve

1. Nemes Lili: keretrendszer implementálása
2. Sally Alala: rescuer agent implementálása, helikopter rescuer lépésének implementálása
3. Jónás Dávid: controller ágens implementálása, troop rescuer lépésének implementálása

A keretrendszer dokumentációja (Nemes Lili)

A keretrendszer 3 fő részből tevődik össze: RescueFramwork, Simulator és World

A World foglalja magába a világ alkotóelemeit, mint a különböző pályaalkotó elemek (fields), a pálya (map), a játékban részt vevő entitások (users), illetve az ezek megjelenítéséhez szükséges képek (images). Alapvető entitások a sebesült (injured) és a két fajta mentőegység (troop és helicopter). A sebesült nem végez különösebb tevékenységet, értesíti a mentőegységeket a helyzetéről, majd pedig az idő elteltével csökken az életereje. Ha az életerő elfogyása előtt egy mentőegység elér hozzá, akkor megmenekül, máskülönben meghal. A mentőegységek lépését három különböző akció határozza meg: Move, Pickup, Deliver. Amennyiben egy mentőegység mozog, vagy sebesült hiányában a bázison várakozik, Move akcióval tér vissza. Amennyiben az útja végén elért egy sebesülthöz, felveszi azt (Pickup), illetve ha egy sebsülttel épségben visszaér a bázisra, ott lerakja (Deliver). Három különböző field típus alkotja a pályát: hegy, erdő és bázis. A mentőegységek a bázisokról indulnak, sebesültet tetszőlegesen lehelyezhetünk a pálya bármely pontján. A hegy és erdő közötti egyetlen különbség, hogy hegy mezőre a helikopter nem tud leszállni, így hegyen lévő sebesültet csak gyalogos mentőegység tud megmenteni (troop). A különböző fieldek koordinátával rendelkeznek, illetve lekérhetőek róluk a szomszédaik. A pálya kialakítása egy map.txt-ben van definiálva, amit módosítva tetszőlegesen változtathatjuk a pálya felépítését.
A Simulator felel a környezet felépítéséért, és a játékot megvalósító szimuláció létrehozásáért. Három osztály alkotja: Env.java, Simulator.java, TimeStepper.java. A TimeStepper egy egyszerű számlálót valósít meg, ami azért felelős, hogy a játékban eltelt időt kövesse, ami meghatározza, hogy a sebesültek milyen ütemben veszítenek az életerejükből, illetve mikor tudnak a mentőegységek mozogni (helikopter minden időpillanatban, troop csak minden 3. időegységben tud mozogni). Az Env létrehozza a környezetet, amelyet később majd a Simulator futtat. Ehhez segítségül használja a *Rendszertervezés II. laboratórium MIT4*-hez kiadott RescueFramworkön alapuló, bár alaposan átdolgozott keretrendszert a grafikus megjelenítéshez. A felületen elérhető a képekből álló pálya, mely a map.txt-ből készül el, lehetőség a szimuláció sebességének beállítására illetve a rescuer és troop ágensek számának beállítására. Vannak gombok a szimuláció elindítására, léptetésére, szüneteltetésére és megállítására is. A sebesültek a képernyőre kattintással jelennek meg, életerejük pedig lépésről lépésre fogy. A többi szereplő a start megnyomása után jelenik meg a hozzájuk tartozó állomásokon, mozgásuk jól látható a szimuláció közben. A RescueFramwork inicializálása mellett inicializálja a controller ágenst (ennek viselkedéséről később), és definiál olyan függvényeket, mint a setBids(), updateCurrentBids vagy az informRescuers(),

amelyek eszközt szolgáltatnak a controller ágens számára, hogy frissítse a hiedelmeit és percepciót. A Simulator osztály felel a játék szimulálásért: nyomon követi a sebsülteket és mentőcsapatokat, definiálja a step() metódust, ami lépteti az időt, és ennek függvényében a különböző mentőcsapatok mozgását illetve a sebsültek életének csökkenését. Kezeli emellett az olyan esetket, mint egy sebesültnek a felvétele, és a bázisra való elszállítása, egy sérült helyzetének beállítása a mentőcsapatok céljául, illetve nyomon követi, hány embert sikerült megmenteni, és hányan nem. Kezeli továbbá a grafikus felületen elérhető módosításokat, mint a szimuláció megállítása, léptetése, újraindítása és új ágensek elhelyezését.

Rescuer ágens és helikopter lépés dokumentációja (Sally Alala)

A Rescuer Agent és helikopter rescuer lépésének implementálása a következőképpen néz ki:
A Rescuer Agent ASL fájlja strukturált módon épül fel annak érdekében, hogy hatékonyan kezelje a beérkező információkat és feladatokat. Az inicializáció során, amikor a !start üzenetet kapja, az agent aktiválódik, és jelzi, hogy elindult. Ezután, ha egy newBid típusú belief érkezik, az agent reagál a !processBid akcióval, amely továbbítja ezeket az ajánlatokat a controller ágensnek a saját nevével együtt, hogy az eldönthesse, ki lesz az aktuális forduló győztese. Fontos megjegyezni, hogy az agent külön figyelmet fordít arra, hogy ne tárolja a konkrét newBid beliefeket, csupán a saját nevét, ezáltal minimalizálva a világkép túlzott zsúfoltságát. Emellett a használt beliefek és perceptek nevei az Env osztályból olvashatók ki, amelyeket az agent megfelelően kezel a hatékony működés érdekében.
A helikopter step függvényében határozzuk meg a helikopter lépésének mikéntjét. A bothCanStep nevű változót a paraméterül kapjuk, hogy jelezze, hogy mindkét mentőegység (helicopter, troop) készen áll-e a lépésre. Ha ennek az értéke false, az azt jelenti, hogy a másik mentőegység jelenleg nem tud lépni, tehát a helikopter tehet egy lépést. Ebben az esetben a helikopter lép egy lépést (egy lépés alatt 3 mezőt tud bejárni), ha még van direction amerre tudna menni.
Ha a bothCanStep értéke true, az azt jelenti, hogy mindkét mentőegység készen áll a lépésre. Ekkor ellenőrizzük, hogy van-e még hova tovább mennie a helikopternek. Ha igen, a helikopter mozog a következő mezőre az útvonal mentén.

Ha a helikopter eléri az Injured-höz tartozó mezőt, és ugyebár ez a mező a helikopter célpontja (targetLocation), akkor fel tudja venni az embert, ekkor az eltűnik a pályáról.
Ha a helikopter eléri a szállítási állomást, és már nincs további útvonala (vagyis path.isEmpty() true-val tér vissza), akkor le tudja tenni a szállított sérültet.
Ha egyik fenti feltétel sem teljesül, a helikopter alapértelmezés szerint egy lépést tesz előre.

Controller ágens és troop lépés dokumentáció (Jónás Dávid)

A keretrendszer felépítése után meg kellett valósítani a rescuer ágenst, ami magába foglalja a szárazföldi (troop) és légi (helikopter) mentőcsapatokat is. Ez az ágens kap egy newBid típusú percepciót, amelyet az Env osztály informRescuers() függvénye valósít meg. Ennek a függvénynek a segítségével minden mentőegység értesítést kap a jelenlegi sebesültekről, és minden elérhető mentőegység ajánlatot küld a controller ágensnek, hogy melyik sebesültet milyen költséggel tudja elérni. Miután ezek az ajánlatok elküldésre kerültek, következik a controller ágens, melynek feladata, hogy a beérkező ajánlatok között MinAvg algoritmust használva kiszámítsa, hogy melyik mentőegységnek a leggazdaságosabb megmentenie az adott sebesültet. Mindig eltárolja a legkisebb költségű szavazatot, és új szavazat beérkezése setén megvizsgálja, hogy jobb-e vagy rosszabb, mint a meglévő. Ezen a műveletek elvégzéséhez az Env-ben definiált függvényeket használja, mint az updateCurrentbids(), ami azt az eseményt kezeli, amikor egy sebsültet felvett egy mentőegység. Ilyenkor az adott sebesültre vonatkozó rendszerek törlődnek a szavazatok közül.
Miután az ágensek meghatározták a szavaztokat, és a controller eldöntötte, hogy melyik mentőegység induljon el egy adott sebeültért, figyelembe kell vennünk, hogy a Helicopter és Troop osztály step függvénye nem egyformán működik, mert különböző kritériumoknak kell megfelelniük. A troop csak minden harmadik időegységben tud lépést végrehajtani, erre figyel a függvény. A Helikopter nem tud leszállni hegyre, ezért ha egy sebesült a hegyen található, akkor a helikopter nem is tesz ajánlatot az adott sebsültre, mert tudja, hogy nem képes megmenteni. Ezeket a különbségeket leszámítva a két egység hasonlóan viselkedik. Alapvetően követik az útvonalukat, amit a Simulator osztály rescuedPerson() és allocateInjured() függvények szolgáltatnak számukra. Amennyiben a ótvonaluk nem üres, és nem érték el a célpontot,

akkor mozognak (Action.MOVE). Amennyiben elérték a célpontjukat, akkor felveszik a sebesültet, törlik a célpontjukat, és elindulnak a legközelebbi bázishoz (Action.PICKUP). Ha a célpontjuk üres, és az útvonaluk is üres, ez azt jelenti, hogy eljutottak a sebesülttel a bázisra, és ott lerakják (Action.DELIVER).

Bemutató videó (linkje)

https://bmeedu-my.sharepoint.com/:v:/g/personal/lili_nemes_edu_bme_hu/EZ-yVSObCxhDtUtyHUDnStsBPAWbfajiVu_ZWsck8YFRwg?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcml2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=t4KlzI

A link sajnos kattintásra nem működik, de böngészőbe másolással elérhető a file.