Robotkar programozása dámajátékra

Feladat kiírás:

A fejlesztési környezet egy kollaboratív FANUC CR-7/A/L robot Onrobot vákuumos

megfogóval, melyet a FANUC gyári szoftverével lehet külső számítógépről programozni. A

feladatban a robotnak kell egy emberrel dámajátékot játszania. A képfeldolgozási modul

azonban nem részei a fejlesztési feladatnak, de a szoftver tervezést és fejlesztést úgy kell

megoldani, hogy integrálható legyen. Így a

Humán játékos egy felületen megadja a lépését, amit a robot végrehajt. Ezután a dámajátszó

program is lép, mely program alapulhat valamilyen nyílt forráskódú megoldón, és ezt a lépést

is végrehajtja a robot. Ha kiütésre kerül a sor a robot veszi le mindkét résztvevő korongját.

illetve az esetleges dáma keletkezésekor a robot a korongot kicseréli a fix tárolóhelyen levő 3D dáma korongra. A program észleli, ha valamelyik fél nyert és kijelzi.

Feladatok:

Megfelelő modell felépítése,

A korong felvétel és lerakás tesztelése, a szükséges mozgáspályák megalkotása

A rendszerarchitektúra megtervezése

A dámajátszó szoftvermodul kialakítása (nyílt forráskódú megoldás alapján)

A felhasználói felület kifejlesztése

A robotmozgató modul fejlesztése, tesztelése

Fejlesztés git verziókövetéssel

Bemutatható demonstráció elkészítése

Kezelői felület:

A kezelői felület egy Windows Forms-al elkészített dáma tábla. Az játékos a fehér korongokat irányítja, míg a gép a feketéket. A dáma szabályainak megfelelően, ha bármelyik fél eljuttatja a korongját a tábla másik felére, akkor az a korong lecserélődik egy dámára/királyra. Ez a király már nem csak előre tud lépni, hanem hátrafele is.

A játékos a következő módon tud lépni: kiválasztja az egérrel az egyik korongját, ennek hatására az kiválasztódik, majd a táblán átszíneződnek kékre azok a mezők, amikre ez a korong átléptethető. A lehetséges mezőkbe azok is beletartoznak, amikhez a játékosnak le kéne ütnie a gép korongját. Ezek után a játékos rákattint arra a kék mezőre amelyikre lépni szeretne, ezzel végre is hajtja a lépést. Ha mégis másik koronggal szeretne lépni, mint a amelyikre eredetileg rákattintott akkor a csak ki kell választania egy másik korongját, ekkor minden mező átszíneződik megfelelő színére (érvényes lépés kékre, váltás előtti érvényes lépés visszaszíneződik eredeti színére).

Játékos lépése után a gép automatikusan lelépi a saját lépését, majd utána megint a játékos következik.

Miután veszít az egyik fél automatikusan megjelenik egy másik ablak, amin látható egy üzenet, ami azt írja ki, hogy ki nyert, illetve szerepel még rajta egy kilépés gomb, ami bezárja a programot.

A képen képernyőkép, kör, tér, dámajáték látható

Automatikusan generált leírás

A képen szimbólum, Grafika, embléma, szöveg látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, sor, szám látható

Automatikusan generált leírás

Square osztály:

A tábla egyes mezőit reprezentálják és azokról tárol információt. Tárolja, hogy milyen színű elem van a mezőn, ez lehet: Empty, Red, Black. Empty, ha nincs korong a mezőn, Red, ha fehér korong van a mezőn (mivel a dámát szokták piros-fekete bábukkal is játszani a program ezeket a színeket felhasználva íródott), Black, ha fekete korong van a mezőn. Ezen kívül még eltárolja, hogy dáma/király van-e a mezőn egy bool értékként.

Move osztály:

A lépésről tárol el adatokat. Tárolja a lépés kezdő- és végpontját, egy listát azokról a pontokról, amikről korongot üt le és a lépéshez tartozó pontértéket.

Tree osztály:

Egy fa létrehozásához és végig járásához szükséges függvények. A fa egy elemének van értéke, lehetnek gyerekei, ezeket egy listában tárolja, illetve lehet szülője.

Utils osztály:

GetOpenSquares függvény ad vissza lehetséges Move-okat. Ezek a move-ok a sima lépések és az ütések, ha egyszerre több ütés történne azt rekurzióval határozza meg.

AI\_Tree osztály:

A gép a saját lépését egy súlyozott fával határozza meg. Ennek a fának lehet állítani a max mélységét és hogy az egyes változóknak mennyi legyen a súlya, ilyen a változó: CapturePiece, CaptureKing, CaptureDouble, CaptureMulti, AtRisk, KingAtRisk.

Minden lehetséges lépésből épít egy ágat. Az egyes lépéseket végrehajtja egy virtuális táblán, majd max mélységtől függően, a lépés utáni lehetséges lépéseket is lelépi, csak most a másik játékos korongjaival. Fa készítése közben a lépésekhez a változóknak megfelelően értékeket rendel.

Amikor elkészült a fa kiértékeli az egyes ágakat és a legmagasabb ponttal rendelkező ág kezdő lépését fogja lépni.

BoardPanel osztály:

Ez az osztály felel tábla kirajzolásáért, a korongok mozgatásáért és a játékos vereségének figyeléséért. A tábla egyes mezőiről tárolt információkat egy Board nevű tömbben tárolja, amiben Square típusú elemek vannak. Ezen kívül tárolja még, hogy melyik fél köre van.

Függvényei:

Konstruktorában hozza létre az eredeti játék állást. Beállítja, hogy melyik AI-t használja majd a program.

OnPaint:

Board tömb információjának megfelelően felrajzolja a táblát és a korongokat.

OnMouseClick:

Kattintásnál megkeresi azt a pontot, ahova kattintottak, majd megnézi, hogy a megfelelő játékos köre van-e, ha nem akkor kilép a függvényből. Ha megfelelő játékos köre van éppen, akkor eldönti, hogy ez egy lépés volt, vagy csak egy korong kijelölése. Ha lépés akkor végrehajtja a lépést, ha korong kijelölés, akkor kiírja konzolra a kijelölt korong koordinátáit, illetve megkeresi a korongnak a lehetséges lépéseit. Mindkét esetben a végén újra rajzoltatja a táblát.

MoveChecker:

Ez a függvény felel a játékos és az AI korongjainak lépéséért. Bemenetként kapja a Move-ot, amit le kell lépnie. Először megnézi, hogy a Move kezdete és vége megegyezik-e, ha igen akkor kilép és a robot veszített (csak a robot tud ilyen Move-ot küldeni). Aztán megnézi, hogy a játékosnak van-e még korongja, ha nincs kilép és a játékos veszített. Ezután kiírja konzolra a leütött korongok listáját, ha vannak leütött korong. Átírja a Board megfelelő elemeit és ellenőrzi, hogy lett-e új dáma a táblán, ha lett akkor frissíti a Board megfelelő elemét, majd meghívja a robotkar azon függvényét, ami fizikailag is lelépi a lépést. Lépés végrehajtása után, még meghívja azt a függvényt, ami ellenőrzi, hogy a játékosnak van-e még lépése, ha nincs kilép és a játékos veszít. Végén újra rajzoltat és következő kört hív.

ResetSquare:

A kapott pontot visszaállítja alapállapotra.

AdvanceTurn:

Megváltoztatja kinek a köre van és ha az AI köre kezdődik akkor meghívja az AI-nak azt a függvényét, ami kiválasztja a következő lépését, majd ezt a lépést végre is hajtatja a MoveChecker-rel.

PlayerLose:

Ez a függvény felel azért, hogy a megadott ponton lévő korongnak van-e érvényes lépése. Érvényes lépésnek tekinti, ha lehet üres mezőre lépni és ha lehet ütni. Külön kezeli, ha a ponton dáma vagy sima korong van. Külön kezeli, ha a pont a tábla valamelyik szélén van vagy a tábla egyik érvényes sarkában.

isKing:

Ha a kapott koordinátákon dáma van akkor igazat ad vissza, ha nem akkor hamist.

isRed:

Ha a kapott koordinátákon fehér bábu van akkor igazat ad vissza, ha nem akkor hamist.

isBlack:

Ha a kapott koordinátákon fekete bábu van akkor igazat ad vissza, ha nem akkor hamist.

Robot osztály:

A játékos és a gép lépéseinek megvalósításáért felel a robotkar segítségével.

Tárolva van a tábla [0,0] pontja és annak a doboznak a helye, ahova a leütött korongok kerülnek. Az összes többi pont a [0,0]-ból számolódik.

nulla0 = Pose.Base(-65, 530, -300, Math.PI, 0, 0)

doboz = Pose.Base(1, 400, -50, Math.PI, 0, 0);

A játék kezdésekor a korongok már a táblán vannak. A dáma bábuk a tábla mellet helyezkednek el.

RobotMoveChecker függvény:

Bemeneten kapott Move alapján ad utasításokat a robotkarnak. Először a megtörténik a korong lépése, utána veszi le a leütött bábukat és csak utána cseréli le a korongot dámára. A lecserélt korong is az leütött korongok közé kerül.

A robotkar lépései: A korong fölé megy -> Lejjebb megy annyira, hogy érintse a korongot -> Bekapcsolja és megfogja a korongot a vákuummal -> végpont fölé megy -> Lejjebb viszi annyira, hogy érintse a táblát -> Korong elengedése

Ha nem korongot, hanem dáma bábut mozgat, akkor más a magasság változás.

FindSource függvény:

A lépés kezdőpontját számolja ki a [0,0] pontból. Két szomszédos mező középpontjának távolsága 32,5 mm.

FindDest függvény:

A lépés végpontját számolja ki a [0,0] pontból.

FindTarget függvény:

A leütött korong helyét számolja ki a [0,0] pontból.