

Terminal

A kódban megjelennek adatszerkezetek közül az egyik legfontosabb a mezo ami egy strázsált duplán láncolt listában tárol el. Ez az adatszerkezet képes tárolni:

x,y-ban a játékban megfeleltetett x,y koordinátákat,

kor_x, kor_y-ban pedig a képernyőn való x,y koordinátái,

foglalt igaz ha a mezőn van olyan egység ami blokkolhatja a rajta áthaladó egységeket

vizsgált az útkeresés fejezetben használja és megnézi hogy egy adott mező át lett-e már vizsgálva egy útkeresésnél,

vizsgálandó akár csak a vizsgált ezt is az útkeresésnél használjuk mégpedig ha ez igaz akkor ő a következő vizsgálandó elemként bekerül egy sorba,

el az az 1,2,3,4,0 számokkal meghatározza hogy él-e az adott mező és hogy a gyémánt alak melyik éle (0-nál nem él),

cal_tav meghatározza a céltól való távolságot az útkeresésben,

loc_cel_tav meghatározza a céltól való távolságot a kezdőponthoz képest,

fel,jobb,bal,also tárolja egy adott mező adott irányba levő szomszédjait,

szarmazik az útkeresésben a szomszédok közül arra mutat ahonnan előbb a cél közelebbről elérhető,

babu a mezőn levő egység memóriacíme van tárolva rajta,

elozo,kov a láncolt listához szükséges,

kie tárolja annak a játékosnak a memóriacímét aki helyezhet le egységeket az adott mezőre,

egyforma_db_piros,egyforma_db_kek,egyforma_db tárolja egy mezőn levő összes egység darabszámát, külön a kékkeket a pirosakat és az összeset.

A másik legfontosabb elem-e a játéknak az egység amiben az összes a programban használatos összes egység összes tulajdonságával,

db az összes azonos egység darabszámát tárolja

azon az egységet azonosító szám (0 esetén az üres egység)

eredeti_x,eredeti_y a bal alsó szürke egységválasztó koordinátái egységenként,

meret az eredeti egységek mérete,

ara mennyi mozgó vagy álló pontba kerül,

range milyen messziire tud ellőni az egység (-1 esetén nem képes lőni),

*x,*y az összes egységnek sorban eltárolva az x,y koordinátái,

sebzes mekkora kárt okoz az ellenségbe,

*elet az összes egységnek sorban eltárolva az élete,

kezdoelet egy egység létrehozásakor az az életmennyiség amivel születik,

A Jatekos struktúra a két játékos adatait képes eltárolni, egy azonosítóval (azon), étellel (elet), álló és mozgó pontját (allopont,mozgopont),

Ut struktúra tárolja egy egységnek bejárandó összes Mezőt lépésenként egy listában.

OsszesUt tartalmazza az összes egység összes útját listában

A KorElemi tartalmazza egy egység hatósugarába érkező mezőket egy listában,

A Vezerlesiadatok tartalmazza a vezerlo modul összes változóját, quit a kilépésért felel, vege a játék végét kell jelezze, a start a játék indítását jelzi, torolt jelzi ha a kirajzolt dolgokat le kell törölni, ujjaz jelzi hogy újra kell-e rajzolni a vászont, kor_vege jelzi egy kör végét, menu jelzi hogy a menüben vagyunk-e, a beolvas jelzi hogy be kell-e olvasnunk egy fájlból, jatekos tárolja hogy melyik játékos van soron, kor_szama tárolja a körök számát, akt_kor az gér helyzeténél levő kordinátákat jelzi, egyseg_val jelzi melyik egység az ami ki van választva a letételhez,

Palya struktúra tartalmazza a mezőket az egységeket és a játékosokat,

Modulok: Minden_inicializalasi_elem: inicializálja az összes egység, játékos, mező, vezérlő alap értékeit.

A rajzok modul felel a kirajzolásokért és a mezők feltöltéséért(uj_kor-ban), a ki_rajzol() egyben tartalmazza a legtöbb kirajzolást egyben

SDL_letrehozása létrehozza az SDL hez szükséges alap beállításokat.

Utkeresés felelős az utak és a szomszédok megkereséséért és ezek felszabadításáért

Vezerles modul az egyik legfontosabb ami összeköti az összes modult és ezek alapján és a felhasználó által generált inputokkal változtatja a Palyaban és a Vezerlesiadatokban tárolt adatokat, például képes meghatározni egy kör végét a játék végét és ugyan így levonni egy játékos pontszámát vagy életét.

Az animacio modul felel a szimulációért

A program SDL függvénykönyvtárakat használja.