Programozói dokumentáció

Felépítés

Főbb file-ok

a) GameCycle - a különböző scene-eket és azok közötti átváltást kezeli
a1) SceneMenu - megjeleníti és kezeli a főmenüt
a2) SceneGame - megjeleníti és kezeli a gameplayt
b) DinTomb - a kert állapotát mentő dinamikus tömböt kezeli (foglalás, feltöltés, átméretezés, felszabadítás)
c) ManagementGame - a játékállásokat kezeli (mentés, betöltés, sceneváltás)
d) AssetDraw - grafikus megjelenítést kezeli
e) InputGame - egér, billentyűzet vezérléseket kezeli

Többi file-ok

- Ablak
  ablak megjelenítéséért, teljesképernyős váltásért és gombok helyének meghatározásáért felelős
- Animation
  játékos animációjáért felelős, következő animációs állapotot állítja be
- Collisions
  Azt ellenőrzi, hogy a melyik gombot nyomtuk meg és, hogy valami egy adott határon belül van-e.

- KertErtekad
  kerten belül (tömbbnek) növeli az értékét ezzel a növényeket növeszti
- SzovegKiir
  adott szöveget, adott helyen, adott igazítással megjelenít

Könyvtárak

A grafikus megjelenítéshez az SDL könyvtárait használja a program:

- <SDL2/SDL.h> alap SDL függvények használatához
- <SDL2/SDL_ttf.h> szovegek grafikus megjelenítéséhez
- <SDL2/SDL2_gfxPrimitives.h> alap kirajzolásokhoz (pl:boxRGBA)
  Ezen kívül a C beépített könyvtárait használja:
- <stdio.h> kiírások, file-kezelésekhez
- <stdlib.> dinamikus memóriakezeléshez
- <string.h> stringek egyszerűbb kezeléséhez
- <stdbool.h> bool változó használatához

Főbb adatszerkezetek

A kert állapotát egy dinamikusan foglalt 2D tömben mentem.

```
typedef struct DinTomb
{
    int **adat;//novenyek allapota
    int meretSor;//hany darab sor van
```

```
    int meretOszlop;//hany darab oszlop van
} DinTomb;
```

Az **adat megfelelő sora és oszlopában lévő érték határozza meg, hogy éppen milyen állapotú növényt kell kirajzolni és hogy az teljesen megnőtt-e. Ezt az értéket növelem az idő múlásával, hogy a növények nőjenek.
A kertnek a méretét lehet növelni és csökkenteni melyet a méretSor vagy méretOszlop megváltoztatásával és az adat tömb dinamikus újrafoglalásával és a régi foglalás felszabadításával tehetünk meg a dintomb_atmeretez függvényel.
Az itt foglalt memóriáta program befejeztével fel kell szabadítani amiért a memoryCleanup függvény felelős.

Függvények

Játékkezelés

```
void gameCycle();
```

Funkció: létrehozza az alapvető változókat, a megjelenítő felületet. Továbbá kezeli, hogy éppen melyik Sceneben kell lennünk.
Paraméterek melyet a Sceneknek ad és elötte inicializál:
Paraméterként kapja:
-ablak megjelenítshez szükséges változókat tartalmazó struct
-gameVar játék állapotával kapcsolatos változók
-save_file melyik mentett állást töltse be a program

```
void MenuScene(WindowParam* ablak, gameVariables* gameVar,
char* save_file)
```

Funkció: megjeleníti és kezeli a főmenüt (gombokat kezeli). Segítségével új játékot hozhatunk létre, vagy már meglévőket tölthetünk be.

```
void GameScene(WindowParam* ablak, gameVariables* gameVar,
char *save_file)
```

Funkció: megjeleníti és kezeli a gameplayt (kert dinamikus kezelését, inputokat, gombokat kezeli) inicializálja és változtatja a játékhoz szükséges változókat.

Adatkezelés

```
bool dintomb_atmeretez(DinTomb *ketDinTomb, int
ujmeretSor, int ujmeretOszlop)
```

Funkció: atmeretez egy megadott dinamikus 2D tombot, megadott sor és osszlop nagyságúra
Paraméterek: dinamikus foglalt tömb struktúrája, új oszlop és sorok mennyisége
Visszatérési érték: sikerült-e az átméretezés
Megjegyzés: a lefoglalt területet a hívónak használat után fel kell szabadítania

```
void memoryCleanup(DinTomb *ketDinTomb,SDL_Texture
*novenyek,SDL_Texture *playerText)
```

Funkció: felszabadítja a dinamikus tömb és az egyéb textúrák álltal foglalt memóriát
Paraméterek: dinamikus foglalt 2D tömb, novenyek és karakter kirajzolásához szükséges textúra

```
void gamekeyDown(SDL_Event event, DinTomb *kertTartalma,
player* p1, int *isSetting,char* save_file);
```

Funkció: kezeli a billentyűzet álltal adott inputokat
Paraméterk: billentyűzet állapotát tartalmazó SDL esemény, a játékban lévő kert dinamikus tömbje, a játékos állapotát tároló struktúra címe, a jelenlegi beállítások, a mentési fájl neve.

```
bool loadGame(int *kertSor, int *kertOszlop, player *p1,
char *save_file)
```

Funkció: Betölti a játékállást a paraméterként kapott változókba a megadott file-ból
Paraméterek: kert méretét meghatározó paraméterek, játékos pénze, játékos pozícióját, sebességét tartalmazó struktúra, melyik mentett állást töltse be a program
Visszatérési érték: sikeres volt-e a betöltés