Moonlander

Felhasználói dokumentáció:

A Moonlander egy játék, melynek célja a sikeres holdraszállás végrehajtása egy űrkomppal. Az űrkompot lehet forgatni, valamint ki-be kapcsolni a motorját. Az irányításhoz szükséges információkat meg lehet tekinteni a játék indításakor a “Controls” menüpontban.

A sikeres holdraszállásnak feltételei vannak:

- Az űrkompnak az adott (fehérrel jelölt) leszálló pályán kell landolnia.
- Az űrkomp sebességének függőleges irányú komponense nem lehet nagyobb 30-nál, míg a vízszintes komponense kisebb kell, hogy legyen 5-nél.
- Az űrkomp hajtóművét nem szabad bekapcsolni túl alacsonyan (ha ég a lámpa), mert akkor leég a hajtómű és sikertelen lesz a landolás.
- Az űrkomp szögelfordulásának 3-nak vagy annál kisebbnek kell lennie.
- Az űrkompnak rendelkeznie kell még, üzemanyaggal, mert ha elfogy a komp lezuhan.

A játék rögzíti a játékos nevét, a landoláshoz szükséges időt, valamint a megmaradt üzemanyagot. Ezek az adatok alapján készít egy dicsőség listát, ami az idő alapján összeállított sorrendben (amennyiben az idő megegyezik az üzemanyagszint alapján különbözteti meg ki kerül feljebb) kiírja az első 20 játékos nevét, idejét és megmaradt üzemanyag százalékát. Ezt meg lehet tekinteni a „Scoreboard” menüpontban. Itt a jobb felső sarokban található gombbal kinullázhatóak az adatok.

Sikertelen leszálláskor „Game Over” szöveg jelenik meg és az indok, ami miatt sikertelen volt a leszállás, sikeres leszálláskor a „Victory” felirat íródik ki a képernyőre. A játék ezután felkínálja a lehetőséget, hogy újra induljon egy következő játék, vagy kilépjen. A menüből az „Esc” gomb lenyomásával bármikor ki lehet lépni.

Programozói dokumentáció:

A program .c típusú fájljai:

- main.c → A játék fő loopja és az inicializálások találhatóak itt.
- alapfv.c is → Olyan függvényeket tartalmaz, amelyeket, más modulok használnak.
- menu.c ebben a → A menü működéséhez szükséges függvények vannak modulban.
- map.c → A pálya generálásához szükséges függvények.
- colision.c → A komp és a felszín ütközését figyelő függvények összessége.
- end.c → A játék végét szabályozó függvények találhatóak itt.
- filekez.c → A filekezelést végző függvények.

A program .h típusú fájljai:

- alapfv.h → alapfv.c függvényeinek deklarációi.
- menu.h → menu.c függvényeinek deklarációi.
- map.h → map.c függvényeinek deklarációi.
- colision.h → colision.c függvényeinek deklarációi.
- end.h → end.c függvényeinek deklarációi.
- filekez.h → filekez.c függvényeinek deklarációi.
- structs.h → A játékhoz szükséges struktúrák.

Struct-ok:

adat: [char * Név, int perc, int mp10, int mp1, int Fuel] → Ez az adatstruktúra a fájlba mentést és a fájlból kiolvasást segíti.

Ido: [int perc, int mp10, int mp1] → Az idő méréséhez szükséges integer-ek.

Sebesseg: [double x, double y] → A sebességnek két komponense.

Komp: [char* Name, double x, double y, Sebesseg v, bool motor, bool jobbra, bool balra, int szog, double a, double Fuel, ido Time, bool lampa_eg] → A létrehozott komp adatai.

Függvények:

alapfv.c:

void urkomp_innit	(Komp* k): Egy létrehozott Komp-nak az értékeit változtatja meg.
Uint32 timer	(Uint32 ms, void* param): Egy SDL_Timer-hez szükséges időzítő függvény.
Uint32 mp	(Uint32 ms, void* param): Egy SDL_Timer-hez szükséges időzítő függvény.
void szamol	(Komp* k): A kompnak az aktuális gyorsulását, sebességét és helyzetét számolja ki, majd ezeket az adatokat beleírja a k változóiba.
void ido_szamlal	(Komp* k): Az idő mérését elősegítő függvény.
void sdl_innit	(char const *felirat, int szeles, int magas, SDL_Window**pwindow, SDL_Renderer **prenderer): A SDL ablak létrehozását végzi.
void rajzol_terkep	(SDL_Renderer *renderer, Sint16 *x, Sint16 *y, int melyik): Felrajzolja a „Hold” felszínét (egy sokszög alapján), valamint egy fehér vonalat (leszállópálya).
void rajzol_komp	(SDL_Renderer *renderer, Komp *k, SDL_Texture *kep): Kirajzolja a kapott képet a komp koordinátaira.
void lampa_rajzol	(SDL_Renderer *renderer, SDL_Texture *lampa): Kirajzolja a kapott képet egy rögzített helyre.
void animacio	(SDL_Renderer *renderer, Komp k, SDL_Texture *anim, int i): A komp helyzetéhez képest kirajzolja a kapott képet 3 részre osztva.
void ir	(SDL_Renderer *renderer, SDL_Rect hova, TTF_Font *font, char *szoveg, SDL_Color szin): Kiír egy szöveget a megadott betűstílusban, egy kapott helyre, egy kapott színnel.

void adatok_kiir (SDL_Renderer *renderer, Komp k, TTF_Font *font, Sint16 *py, int melyik): Kiírja a komp adatait a jobb felső sarokba és a játékos nevét a bal felsőbe.

adat* rendez_seged (adat* data, int i): Rendez függvénynek egy segédfüggvénye, felcseréli egy tömb két egymás melletti értékeit.

adat* rendez (adat* data, int sor_sz): A kapott adatokat sorrendbe rendezi az idő alapján (ha az idő megegyezik akkor „Fuel” alapján).

menu.c:

void szoveg_jo (SDL_Renderer *renderer, TTF_Font *font, SDL_Rect n, char* szoveg, SDL_Color szoveg_sz): Kiírja a kapott szöveget egy adott téglalap belsejébe.

void hover_jo (SDL_Renderer *renderer, TTF_Font *font, int a, int b, SDL_Rect n, char* szoveg, SDL_Color kitolt, SDL_Color szoveg_sz, SDL_Color szel): Kiszínezi a kapott téglalapot egy kapott színnel, és beleírja a kapott szöveget, ha a téglalapon van a felhasználó egere. Ha nem visszaszínezi feketére a téglalap belsejét és visszaírja bele a szöveget.

bool press (SDL_Renderer *renderer, SDL_Event *ev, SDL_Rect n): Megvizsgálja, hogy a bal klikk lenyomása az adott SDL_Rect-en belül történt-e. Ennek alapján return-öl.

bool menu (SDL_Renderer *renderer, TTF_Font *font): A menü vezérlésének a függvénye.

void utmutato (SDL_Renderer *renderer, TTF_Font *font): A „Controls” fül vezérléséhez szükséges függvény.

void press_reset (SDL_Renderer *renderer, TTF_Font *font, SDL_Event *ev, SDL_Rect n, adat* data, int sor_sz, bool *scoreboard): A ranglista adatainak törlését végzi.

void ranglista_seged (SDL_Renderer *renderer, TTF_Font *font, int i, char* str, char* szam_str, int m): A ranglista_kiir segédfüggvénye, kiírja a kapott string-et egy szám mögé.

void ranglista_kiir	(SDL_Renderer *renderer, TTF_Font *font, adat* adatok, int sor_sz): A ranglista függvény segédfüggvénye, kiírja a kapott adatokat.
void ranglista	(SDL_Renderer *renderer, TTF_Font *font): A „Scoreboard” ablak működését szabályozó függvény.

map.c:

void loop	(Sint16 *pontokx, Sint16 *pontoky, int i): Random generál x és y koordinátákat és a kapott tömbökbe menti.
void random	(SDL_Renderer *renderer, Sint16 *x, Sint16 *y, int melyik): A random pályagenerálást végzi ez a függvény.

coilision.c:

Uint32 Get_Pixel	(SDL_Surface *palya, int x, int y): Visszaadja egy pixel koordinátait Uint32-es formátumban.
void felterkepez	(int *szurke, SDL_Surface *palya): Feltérképezi a „Hold” felszínét a pixelek színét vizsgálva.
void testfor	(Komp k, int *szurke, SDL_Surface *palya, bool *colide): Megvizsgálja, hogy a komp belement-e a felszínben az alapján, hogy megváltozott-e a felszín színe.
void adott_magassag	(Komp *k, int *szurke, bool *lampa): Azt vizsgálja, hogy a játékos használja-e a motort a felszínhez túl közel.

end.c:

bool vege	(SDL_Renderer *renderer, TTF_Font *font, Komp k, Sint16 *px, int melyik): Megvizsgálja, hogy mért lett vége a játéknak és kiírja, hogy „Victory” vagy, hogy „Game Over”.

void lampa_motor	(SDL_Renderer *renderer, TTF_Font *font): Ha túl alacsonyan volt a motor használva, kiírja, hogy game over és a vereség indokét.
void press_again	(SDL_Renderer *renderer, TTF_Font *font, SDL_Event *ev, SDL_Rect n, bool* again, bool *exit): play_again menünek a segéd függvénye, megvizsgálja, hogy melyik SDL_Rect-en belül lett lenyomva a bal klikk.
void play_again	(SDL_Renderer *renderer, TTF_Font *font, bool* megint, bool *leave): Feldob egy kétopciós menüt, és megvárja míg a felhasználó választ.

filekez.c:

int hany_sor	(): Megvizsgálja, hogy az Adatox.txt hány sorból áll.
adat* betolt	(int *sor_sz): Az Adatok.txt-ből kiolvassa az adatokat és egy adat tömbben tárolja, ami a függvény visszatérési értéke.
void file_kezeles	(FILE *fp, Komp k): Beleírja az Adatok.txt-be a komp adatait.
char* nev_bevitel	(SDL_Renderer *renderer, TTF_Font *font): A játék indulása előtti névbevitelhez szükséges függvény, addig vár míg a felhasználó meg nem nyomja az entert.
bool input_text	(char *dest, size_t hossz, SDL_Rect teglalap, SDL_Color hatter, SDL_Color szoveg, TTF_Font *font, SDL_Renderer *renderer): InfoC-n található függvény, ami szöveg beolvasást szolgálja.

A header file-okban az előbb említett .c fájlok függvényeinek a deklarációi találhatóak.

A structs.h-ban vannak a játékhoz bevezetett struktúrák.