FELHASZNÁLÓI DOKUMENTÁCIÓ

JÁTÉKMENET ÉS A FŐBB TUDNIVALÓK

A játékot megnyitva egy kezdőképernyő fogja fogadni a játékos. A kezdőképernyőn a játék logója és egy kis felirat található. Ahogyan a felirat is jelzi a játékosnak, ahhoz, hogy elinduljon a menet ez „enter" billentyű gombot le kell nyomni.

Amint a játék elindult, a figura és a helyzetjelző kijelző megjelenik a képernyőn. A helyzetjelző kijelzőn az alábbi adatokat tudjuk leolvasni: Y irányú gyorsulás (Velocity Y), X irányú gyorsulás (Velocity X), Magasság (Attitude), Üzemanyag (Fuel). A kijelzőn továbbá van egy leszállást jelző lámpa is, ami zöldre vált, ha az űrhajónk a leszálló pad felett van körülbelül 100 egységnyi távolságra. Ez a jelzőlámpa tulajdonképpen felhívja a játékos figyelmét, hogy most már nagyon óvatosan kell az űrhajót irányítani.

Az úrhajó nagyobb ütközés, gyors leszállás, oldalirányba sodródva leszállás, elforogva, becsapódás esetén eltűnik és egy „game over" panel fog felugrani. Ekkor a játékos kiválaszthatja, hogy újból szeretné-e kezdeni a játékot (New Game), vagy kilép (Quit). Viszont, ha sikerült megfelelően lerakni az űrkompot akkor egy „You Win" panel fog felugrani a következő opciókkal.: Tabella (Leaderboard), Új játék (New Game), Mentés (Save), Kilépés (Quit). Amennyiben a játékos a mentés opciót választotta új játék már nem kezdhető és nyilvánvalóan többször ugyanazt az eredményt nem mentheti el ezért a New Game és Save opciók nem lesznek elérhetőek. A tabella folyton frissül, ezért mentés után a játékos megnézheti, hogy felkerült-e a Top 10 listába. A tabellán a 10 legjobb játékos neve, pontszáma és időeredménye látható, pontszám és idő szerint rendezve.

A játékmenet közben elérhető még egy funkció, a menü. A játék menüjét az „Esc" billentyű lenyomásával lehet előhozni. Innen is elérhető a tabella, új játékot is lehet kezdeni, illetve folytatni a játékot. A játék folytatására két opciónk van az „Esc" billentyű újbóli lenyomásával, vagy a „Resume" gomb megnyomása. A játékból teljesen kilépni a „Quit" opcióval lehet.

JÁTÉK IRÁNYÍTÁSA, BILLENTYŰZET KIOSZTÁS

Az úrhajót a következőkkel lehet irányítani:

- Jobbra nyíl – *Elforog jobbra*
- Balra nyíl – *Elforog balra*
- Szóköz (első lenyomás) – *Bekapcsol a hajtómű*
- Szóköz (második lenyomás) – *Kikapcsol a hajtómű*

Menü irányítása:

- Esc billentyű (első lenyomás) – *Előhozza a menüt*
- Esc billentyű (második lenyomás) – *Eltünteti a menüt*
- Egér – *A menü gombjaira kattintani kell*

PROGRAMOZÓI DOKUMENTÁCIÓ

A PROGRAM FELÉPÍTÉSE, FŐBB PONTJAI

A main program a függvények, a setup függvény kivételével, egy while loopba vannak belerakva. A programban használt függvények modulokra lettek bontva feladatuk szerint. A programban található főbb függvények és feladataik:

- Initialize_window – *Itt készítjük el az ablakunkat*
- Setup - *Ebben a függvényben adunk a változóknak kezdő értékeket, illetve itt történik a pálya legenerálas is*
- Processe_input - *Ebben a függvényben figyeljük/mentjük az interakciókat a perifériákról*
- Update - *A játék menetének lényegi részé itt történik. A változásokat ebben a függvényben kezeljük, itt állítjuk be a megfelelő FPS-t, és a mozgások kirajzolásához delta_time-ot is itt számoljuk ki. Az ütközéseket, a karakter hold közötti távot is itt számoljuk ki és a karakternek itt adunk sebességet.*
- Render – *Rajzoló függvény, amit a játékos a képernyőn lát ez a függvény rajzoltatja ki. Ez a függvény foglalkozik továbbá azzal is, hogy mi és mikor legyen kirajzolva.*
- Save – *Elmenti a játékos nevét, idejét és pontját (ha a játékos elmenti a rekordját)*
- Destroy_window - *Kilépéshez és megfelelő programleállításhoz szükséges műveleteket végez. Dinamikus tömbök, láncolt lista felszabadítása stb.*

A grafikai megjelenítéshez egy platformfüggetlen multimédiás függvény könyvtárat, az SDL-t használtam (fordításhoz szükséges). A következő SDL-es könyvtárakat használja a program: SDL2_gfxPrimitives, SDL_image, SDL_ttf és az alap SDL könyvtárat.

MODULOK ÉS FÜGGVÉNYEIK

A program több modulra lett felbontva (tíz .c és tizenegy .h fájlra). Ezek a modulok a következők: *destroy, initialize, input, lista, render, save, setup, surfaces, text, updates, structs, lista.* Ezekről a modulokról bővebben.:

Initialize modul

Ebben készítjük el az ablakot a megfelelő paraméterek mentén. Az ablakot fix magasságú és fix szélességűnek adjuk meg, és az ablak típusa pedig borderless ennek fő oka az, hogy rákényszerítsük a felhasználót, hogy használja a menüt, ha már van, illetve a későbbi szövegbeviteleknél sem kell annyi esetet kezelnünk. A renderer rajzolónkat is itt inicializáljuk.

Structs modul

Ebben a modulban hozzuk létre azokat a struktúrákat, amiknek az értékeivel későbbiekben dolgozni fogunk. Ezeket a struktúrákat és azok változóit későbbiekben egy táblázat formájában részletesebben le lesznek írva, rövid magyarázattal. Főbb struktúrák: *Time, poly, ints, bools, doubles, fuel_tomb, koordinata, karakter, bgr, display, ListaElem.*

Setup modul

A setup modul egyetlen egy függvényt tartalmaz, aminek a feladata a következő: a struktúrák adatainak kezdőérték adás, pálya pontjainak legenerálása. A pálya pontjainak legenerálása a következő keppen néz ki:

A pont_tomb típusú tömb x és y részeiben tároljuk el az egyeneseket összekötő pontokat (Az egyeneseket majd a rendererbe rajzoltatjuk ki). A poligon tömb a pálya vonala alatti rész kitöltéséhez kellő tömb, ami szinten pontokat tartalmaz, ami alapján ki fogja a renderben tölteni a teret. A pálya le generálását random számmal oldjuk meg. Ha páros a randomszámunk akkor felfele megy, ha páratlan lefele fog menni. A randomszámunk maximum érteknek 24 vehet fel, minimum pedig 0-at. Mindig az előző pont y koordinátájához adjuk, vagy vonjuk ki a random szam érteket. A pontok között fix 30 pontos távolság van, hogy fixen végig érjen a vonal a képernyő szeléig. Fontos, hogy a pálya pontjai nem lehetnek 100 pontnál melyebben és 300 pontnál magasabban. FONTOS: A képernyő teteje a y=0 pontban van, a képernyő legalja a 600 pontban van. Ezek a pontok fixek.

Lista modul

A lista modulon belül öt függvény található. Mind az öt függvény fontos szerepet játszik a tabellához kellő, láncolt lista létrehozásához, alakításához. Az első függvény a *lista_felszabadita*s. Ez a függvény gondoskodik arról, hogy a láncolt listánk megfelelően fel legen szabadítva. A *lista_elejere_beszur* függvényünkkel tudunk a listához új bejegyzést hozzáadni. Ha szóközt tartalmaz a felhasználónév akkor kicseréli '_' jelre. A *fajlbol_beolvas* a mentes.txt fájlból olvassa ki az adatokat és tárolja el a listában. A *fajlba_iras* függvény írja bele a mentes.txt fájlba a listát és a *rendezes* függvény pedig pontszám, majd idő szerint sorba rendezi a listánkat, ezt majd a tabella kirajzoltatásánál használjuk.

Input modul

Ebben a modulban csak egy függvény található, ami pedig a külső perifériákról beszedett információ alapján tárolja/változtatja az adatokat.

Updates modul

Ebben a modulban több függvény is található: *tav, ujmeret_fuel_tomb, figura, idozito, update.* Ebben a modulban történik meg minden változás vezérlése *(update függvényen keresztül)* és mentése a struktúrákba. A *tav* függvényünk feladata a következő: ki kell, hogy számolja a karakter és a hold közötti távolságot. A távolság kiszámításának lényegi gondolatmenetét az egyenesek adták (a setup modulból legenerált pontokat összekötő egyenesek). Ha megvan két pont, abból már fel lehet írni az egyenes egyenletét (két pont lesz az, ami között van a karakter). Ha pedig megvan az egyenes egyenlete akkor már csak meg kell találni azt a pontot, amihez a legközelebb van a karakter. Ehhez fel fogunk venni

két segéd pontot (a függvényen belül ki van kommentelve a két pont kirajzolása). Legyen ez a két pont a Q és az E. A Q pont y koordinátája megegyezik a karakter y koordinátájával, az x koordinátája pedig az egyenesen van rajta (amit a két ponttal adtunk meg), ezt az egyenes egyenletével es egy kis egyenlet rendezessel számoljuk ki. Az E pont x koordinátája a karakter x koordinátájával egyezik meg az y koordinátáját pedig az egyenesen van rajta. (fenti módszerrel határozzuk meg azt is). *FONTOS: ha túlságosan "meredek" az egyenes ezek a számítások nem fognak valós végeredményt adni.*

Az *ujmeret_fuel_tomb* függvény méretezi újra a hajtómű részecskéinek középpontját tartalmazó tömbjét. Akkor méretezi újra, ha a hajtómű bevan kapcsolva, és minden egyes másodpercben egyel növeli a hosszát. Az ujraméretezést az update függvény dönti el.

A figura függvényben kezeljük a karakter elforgatását, hajtómű kezdőállapotának koordinátáit, és a karakter kinézetét.

Időzítő függvény tárolja el a Time típusú változóba az indítás óta eltelt időt perc másodperc formájában. Fontos, hogy mivel a másodpercet delta_time-al kell megszoroznunk a másodperc értéke a struktúrán belül double típusúnak kell lennie.

Render modul

Ebben a modulban egy függvény található és ebben a függvényben döntjük el, hogy mik rajzolódjanak ki és mik nem. Tulajdonkeppen összefogja az összes rajzoló függvényt es adott paraméterek alapján jelenteti meg őket.

Surfaces modul

A surfaces modulban a „felugró ablakok” paneleinek kirajzolása történik. Minden függvényhez tartozik egy adott függvény a text modulból, amik pedig majd a szöveget íratják rá a „téglalapokra”. Panelt alatt gondolok például.: "You are the winner" ablak fekete nem áttetsző részére, amin a felirat rajta van.

Text modul

Ebben a modulban szövegeket rajzoltatunk ki. Két féle SDL-es text függvényt használtam, TTF_RenderUTF8_Blended és a TTF_RenderUTF8_Blended_Wrapped. Az utóbbira a sortörések miatt volt szükség a tabella feliratihoz.

Save modul

A save modulban történik a felhasználó által bevitt rekord eltárolása a *mentes*.txt fájlban és a listában. Itt meghív pár lista modulos függvényt is a program, hiszen láncoltlistába tároljuk a tabella adatait.

Destroy modul

A *destroy* modul felelős a megfelelő kilépésért. Itt szabadítjuk fel a részecske meghajtó dinamikus tömböt és a láncolt listát, töröljük az ablakot és a grafikus renderert.

ADATSZERKEZET

Az adatszerkezet felépítésének gondolatmenete mögött az a főbb szempont volt, hogy minden változót, amit csak tudok, tároljam el struktúrában, még azokat a változókat is amelyekre specifikusan nem akasztható semmilyen címke. Így született meg a gyűjtő struktúra ötletem. A gyűjtőstruktúrák elemeinek típusai megegyeznek. Ilyen struktúra például.: *bools, ints.*

Struktúrák típusai és mire valóak:

- Time – Időtárolás
- Poly – Pálya pontjai
- Ints – Intek tárolása
- Bools – Bool-ok tárolása
- Doubles – Double-s tárolása
- Fuel_tomb – Meghajtó részecskéinek középpontjának koordinátái tárolására
- Koordinata – Koordináták x y tárolására alkalmasak
- Karakter – Bábu adatainak tárolása
- Bgr – Háttérkép adatainak tárolása
- Display – Kijelző adatainak tárolása
- ListaElem – Láncoltlista tárolása

Adat szerkezet		
Struktúra	Változó	Magyarázat
Time	int perc	a percet tárolja el
	double másodperc	a másodpercet tárolja el
Poly	Sint16 polyx	a poligon x koordinatai tarolja el
	Sint16 polyy	a poligon y koordinatait tarolja el
Ints	int katt_x	kattintas x koordinatajat tarolja
	int katt_y	kattintas y koordinatajat tarolja
	int eger_x	eger helyzet x koordinatajat tarolja el
	int eger_y	eger helyzet y koordinatajat tarolja el
	int temp	a reszecske tomb elozo hosszat tarolja
	int hosz	a reszecske tomb jelenlegi hosszat tarolja
	int x	a hajtomu langjanak x kezdoallapotat tarolja
	int y	a hajtomu langjanak y kezdoallapotat tarolja
	int le_x1	a leszallo pad elso x koordinatajat tarolja
	int le_y1	a leszallo pad elso y koordinatajat tarolja
	int le_x2	a leszallo pad masodik x koordinatajat tarolja
	int le_y2	a leszallo pad masodik y koordinatajat tarolja
	int leszallo	egy index erteket tarol a pont tombhoz aminek erteket felul fog
	int ponszam	a jatekos pontszamat tarolja
Bools	bool space_up	igaz, ha a nincs lenyomva a space
	bool left	igaz, ha le van nyomva a balra nyil
	bool right	igaz, ha le van nyomva a jobbra nyil
	bool up	igaz ha egyszer lenyomtak a "space"-t
	bool freez	akkor lesz igaz ha lenyomtak az escapet (megallitja az timert, a hatter
	bool menut	akkor igaz, ha az escapet lenyomtak megjelenik vele a menu
	bool menu_mutat	akkor igaz, ha az escapet lenyomtak, ujboli lenyomasnal hamis erteket kap
	bool presstostart	a kezdokepernyonel hasznaljuk, az enter lenyomasaval igaz lesz
	bool game_o	akkor lesz igaz, ha kimegyunk a kepernyorol a karakterrel, vagy belecsapodunk a holba
	bool tabella_mut	akkor igaz, ha a menubol kivalasztottuk a "Leaderboard" opciot
	bool winner mutat	akkor lesz igaz, ha megfelelo feltetelek mellett jol leraktuk a holdkompot
	bool game_is_running	igaz, ha sikerult az ablak inicializalasa, hamis ha a menuben kivalasztjuk a "Quit" opciot
	bool bonus	akkor lesz igaz, ha a megfelelo feltetelek mellett mas feltételeknek is meglelet (pl kivallo landolas)
	bool beiras	enter lett nyomva beirasokor
	bool mar_beirt	mar irt be a felhasznalo, egyszer irhat csak be
	bool nev_beirva	hogy ne vegtelenszer legyen beirva (loop)ezert kell egy valtozo ami meggatolja
	bool mentes	akkor lesz igaz, ha a menu opciobol kivalasztjuk a "Save" opciot, ekkor elougrik meggy panel a mentes
Doubles	double karakter_egyenes_tav	a karakter es a fold kozotti erteket tarolja
	double delta_time	valos ido erteket tarolja
	double fuel	az uzemanyagot tarolja
Fuel_tomb	double x	a reszecske meghajto buborek kor kozeppontjanak x koordinatajat tarolja
	double y	a reszecske meghajto buborek kor kozeppontjanak y koordinatajat tarolja
	double opicaty	a reszecske meghajto buborek kor attetszoseget tarolja
Koordinata	double x	koordinata tipusu valtozok x erteke
	double y	koordinata tipusu valtozok y erteke
Karakter	double x	karakter x koordinatajat tarolja
	double y	karakter y koordinatajat tarolja
	double width	karakter szelesseget tarolja
	double height	karakter magassagat tarolja
	double y_velocity	karakter y iranyu gyorsulasat(sebesseget) tarolja
	double x_velocity	a karakter x iranyu gyorsulas (sebesseget) tarolja
	double rotation	karakter elfordulasi szoget radianban tarolja
Bgr	double x	hatter x koordinataja
	double y	hatter y koordinataja
	double width	hatter szelessege
	double height	hatter magassaga
Display	double x	jobb fenti kijelzo x koordinataja
	double y	jobb fenti kijelzo y koordinataja
	double width	kijelzo szelessege
	double height	kijelzo magassaga
ListaElem	char felhasznalo	felhasznalo neve
	int pontszam	felhasznalo pontszama
	Time ideje	ideje-> perc és másodperc