

A program egy parancssor alapú körökre osztott kalandjáték, tematikailag autóversenyzés. A játék összes mechanikája függvényekre van bontva, melyeket egy GameState változó alapján hív meg a main függvény.

Fájl lista betűszerinti sorrendben:

Forrásfájlok:

- Driving_System.c – A játékos és a gép irányításának függvényei
- econio.c – Szöveges megjelenítéshez tartozó függvények
- File_System.c – Fájlkezelést vezérlő függvények
- List_and_Entities.c – Dinamikus adatszerkezet és entitások generálása
- main.c – A játék fő függvénye
- Menus.c – A menü rendszer függvényei
- Play_Functions.c – A játékmenethez tartozó fő vezérlőfüggvények

Fejlécfájlok:

- Driving_System.h
- econio.h
- File_System.h
- GameState.h – A játék fő vezérlő állapota
- List_and_Entities.h
- Map.h – A játék fő pálya szerkezete
- Menus.h
- Play_functions.h

Lényegesebb függvények funkciói, struktúrák:

Menus.c

void Menu(int *Load_State)

A játék kezdő menüjét jeleníti meg. A Load_State változóval adja meg, hogy a felhasználó szeretne-e betölteni egy mentést.

void Intermission()

Játékmenet közben a pályák közötti átvezetésre szolgál. Lehetőséget nyújt a programból való kilépésre.

void Prologue()

Játék kezdetén a történet prológusát jeleníti meg.

void Story(int Level)

A játékmenet folyamán a játékos által elért szintnek (Level) megfelelően tovább viszi a történetszálat.

void PlayerCrash()

Ha a játékos a játékmenet folyamán súlyos hibát vét, ez a függvény vezet vissza a játék elejére.

```
typedef struct Item
{
    char *Item_Level;
    char *Item_Tuner;
    int Item_Tuner_Index;
    int Item_Level_Int;

    struct Item *next;
} Item;
```

A tárgylista által használatos tárgyak struktúrája. Tartalmazzák a tárgy szintjét a szövegesen megjelenítendő formában és decimálisan is, továbbá a megnevezés listában való indexét és nevét.

```
typedef struct Machine
{
    char *Tuner;
    int TunerPoint;
    char *Car;
    int CarPoint;
    char *Name;
    int HP;
    int Top;
    int A;
    int B;
    Item *Inventory;
} Machine;
```

A játékos és az ellenfél karaktereinek struktúrája. A játék által figyelembe vett összes tulajdonságukat tartalmazza.

int random_range(int min, int max)

Véletlenszám generálásához alkalmazott függvény, min és max között.

Machine* P_Creation()

A játékos „karakterének” generálására alkalmazott függvény.

Machine* O_Creation(int HP, int Top, int A, int B, int Level)

A játékos ellenfelének generálása, a játékos „karakter” tulajdonságaihoz mérten.

Item* Part_Acquired(int Level)

Tárgylistához tartozó függvény. Láncolt listát hoz létre, melynek elemei a játékos által birtokolható tárgyak. A tárgyak a játékos által elért szint (Level) függvényében kerülnek generálásra.

Item* Pushback(Item* Inventory, int Level)

A megkezdett tárgylistához fűz hozzá még egy szint függvényében generált elemet. Paraméterként megkapja a tárgylista első elemének mutatóját.

void Inventory_Menu(Item* Inventory, Machine *Player)

A játékos tárgylistájának megjelenítésére szolgáló függvény. Paraméterként megkapja a játékos „karakterének” mutatóját, mivel a tárgyak befojásolják a „karakter” tulajdonságait, megváltoztatja azokat.

int Inventory_Free(Item* Inventory)

A tárgylista (láncolt lista) elemeinek felszabadítására szolgáló függvény. Paraméterként a tárgylista kezdő elemének mutatóját kapja meg.

File_System.c

File_System.h

int Load(Machine *Player, int *Level, int *P_DST, int *Pride_Points, int *Win_Count)

Egy szövegfájlból beolvassa egy elmentett játék állását. Paraméterként megkapja a játékos „karakterének” mutatóját, továbbá az elért szintet, a jelenlegi pozíciót, a játékos pontszámát és győzelmeinek számát, melyeket a függvény beállít a szövegfájlban található állapotra.

int Save(Machine *Player, int Level, int P_DST, int Pride_Points, int Win_Count)

Elmenti a játék jelenlegi állapotát egy szövegfájlba.

Map.h

typedef struct Turns

```
{  
    int T_DST;  
    int Limit;  
} Turns;
```

A Turns struktura raktározza egy szövegből (Turns.txt) beolvasott pályának kanyarjait. A kanyaroknak megvan adva a nehézsége, és ezekhez tartozik egy sebességhatár.

void Road_Text(int P_DST)

A játékos pozíciójához mérten megadja szöveggént a legközelebbi kanyar távolságát és nehézségét.

Driving_System.c

Driving_System.h

int Driving(Machine *Player, Machine *Opponent, int P_DST, int O_DST, int P_SPD, bool *Focus)

A játékos „karakterének” irányítására szolgáló függvény. Az irányítás a játékmenetből adódóan sok változót figyelembe vesz. Ezek közé tartozik a játékos „karakterének” tulajdonságai, az ellenfél tulajdonságai, a játékos és az ellenfelének jelenlegi pozíciói, a játékos jelenlegi sebessége és a Focus képesség állapota.

int AI_Driving(Machine *Opponent, int O_SPD, int ADV)

A gép által vezérelt ellenfélnek az irányítása. A gép cselekményét befolyásolja a jelenlegi sebessége és a játékostól való relatív távolsága.

Play_Functions.c

Play_Functions.h

int Start(int *Win, int *P_SPD, int *P_DST, int *O_SPD, int *O_DST, int *P_SP, int *O_SP, int *ADV, bool *Focus, int *Crash, int *Win_Count, int *Pride_Points, int *Level)

Ez a függvény a játék újratekészekor alaphelyzetbe állítja a játékmenet folyamán használt változókat, hogy a program elkerüljön esetleges maradék adatokat melyek zavarhatják az új játékot.

void Play_Menu(Machine *Player, Machine *Opponent, int P_DST, int P_SPD, int ADV, int P_SP, int O_SP, int Timer, int O_SPD, int Level)

A játékmenet közbeni felhasználói felület megjelenítésére szolgál. Megjeleníti a játékos és ellenfelének tulajdonságait, a menetből hátralévő időt (Timer), a játékos sebességét (P_SPD), a játékos és ellenfele közti távolságot (ADV), a játékos és ellenfelének életét (P_SP, O_SP) és a jelenlegi szintet (Level).

void Play(Machine *Player, int Load_State);

A játék legfőbb függvénye. Ez a függvény bonyolítja le a játékmenet során a teljes versenyeket. A Load_State változó adja meg a függvények, hogy új játékot kezdett-e a felhasználó, vagy pedig egy előző mentést töltött be.

A program működése során 3 darab globális változót alkalmaz.

GameState – Felelős a játék jelenlegi állapotáért, vezérli a main függvényt a program futása alatt

LoadState – Megmutatja a programnak, hogy a felhasználó új játékot kezd vagy egy régit folytat.

Turns – A játék pályájának adatait tartalmazza.

Ha a program memória kezelési hibába ütközik azt tudatja a felhasználóval, képernyőre írt szöveges hibüzenetekkel.

Fejlesztőkörnyezet:

Nyelv: C

Program: Code::Blocks 16.01

Compiler: MinGW