

Nagy házi feladat – Megoldásvázlat

A projektem célja dióhéjban:

Egy olyan szövegalapú kalandjáték létrehozása, ahol a történet a felhasználó döntéseitől függően bontakozik ki, emellett a történet egyes fázisaiban harcra is sor kerülhet, ezeknek a megvalósítása. Fontos még, hogy végül bináris fával oldom meg a programot, a megoldásvázlat is azon alapul már.

Szöveges fájljaim:

- story.txt

Csomópont indexe	A történethez tartozó szöveg, mely standard outputra kerül
#x (-jelöli, hány érték van)	
42	Végül adott a koldusnak aranyat, ezt a tettet nem fogja elfelejteni. (J) Ha északnak mész tovább (B) Ha kelet felé tart utad.
52	Belépve a kunyhóba egy szűk szobába találsz magad, a jobb oldaladon egy alvó embert látsz. (J) Ha rátámadsz, míg alszik (B) Ha inkább kionsz a házból.
...	...

Minden egyes csomóponthoz (azaz a történetben egy adott stádiumhoz) tartozik egy szöveg. Ez a szöveg tartalmazza a történetet előre vivő történéseket, cselekményeket, továbbá minden egyes ilyen szöveg végén felkínál 2db választási lehetőséget. A két lehetőség mindig a „J” és „B” lesz, egyrészt azért, mert nincsenek se túl közel se túl távol egymáshoz a billentyűzeten (ezzel kiküszöbölve, hogy véletlen rosszat írjunk be + azért legyen kényelmes), továbbá ez reprezentálja, hogy a bináris fában jobbra vagy ballra megyünk.

- creatures.txt

Lény indexe	Lény életereje	Lény ügyessége
#x (-jelöli, hány érték van)		
1	4	2
2	5	7
...

Történet folyamán a harcokhoz elengedhetetlen tudni, hogy akikkel szemben állunk, mire is képesek. A harcrendszerrel később.

- fa.txt

Node indexe	Bal leszármazott	Jobb leszármazott	Harcidx=Lényidx	Boltidx
45	32	77	1	0
64	126	66	0	3
...

Értelemszerűen, az első oszlop a gyökeret jelöli a fában, a következő kettő meg a bal és jobb oldali leszármazottját annak. A harcidx megmutatja, hogy (ha van harc= harcidx!=0), melyik lényel találkozunk. A boltidx nem feltétlen boltot jelöl a történetben, pl: 3-nál felajánlja a program, hogy adhatunk pénzt egy hajléktalannak. Ezt meg mi döntjük el, hogy megtegyük-e.

Használt struktúrák:

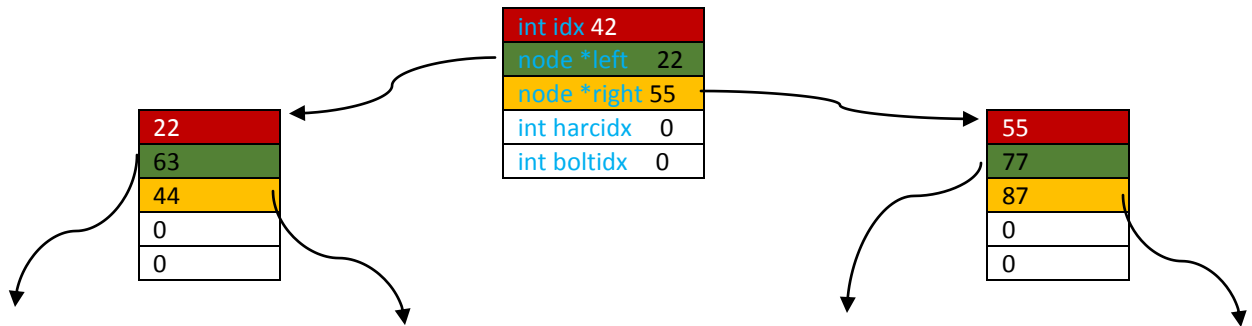
struct story

int idx	szöveg indexe
char szoveg[4000]	szöveg

struct creature

int idx	lény indexe
int hp	lény élete
int agi	lény ügyessége

struct node



Illetve van még két darab struktúra, egy, amelyben a mi dolgaink vannak (életerő, ügyesség, stb.) illetve egy másik mely a táskánkat tartalmazza, de ezek még nincsenek végleges állapotban. De lényegében a játékos dolgai ugyanaz, mint egy creature struktúra, csupán tartalmaz még egy inventory=táska struktúrát, amelyben meg integerek találhatóak majd, melyek azt fogják jelölni, hogy adott dologból mennyi van.

Függvények:

+info: A program használatához a következő billentyűk szükségesek majd: j és k a mozgáshoz, k az evéshez, t a táskánk tartalmának megnézéséhez, a és b fizetésnél a döntéshez.

void fatolt – Megalkotja a bináris fát a fa.txt-ből. For ciklus segítségével addig olvasom be a szükséges adatokat míg a file végéhez nem érek. Aztán a kapott adatokat mindig felhasználom a csomópontok elkészítéséhez. Ehhez felhasználom a **createnode**, **insertleft**, **insertright** függvényeket.

creature* creaturetolt – Dinamikusán tárolt tömböt tölt fel a creature.txtből. For ciklus segítségével addig olvasom be a szükséges adatokat, míg a file végéhez nem érek. Aztán a kapott adatokat belerakosgatom egy creature típusú tömb i.-ik elemébe.

story* storytolt – Dinamikusán tárolt tömböt tölt fel a story.txtből. For ciklus segítségével addig olvasom be a szükséges adatokat, míg a file végéhez nem érek. Aztán a kapott adatokat belerakosgatom egy story típusú tömb i.-ik elemébe.

node* createnode – Létrehoz dinamikusan egy csomópontot. Dinamikusán helyet foglal egy node típusnak, egy indexxel „megjelöli” továbbá beállítja leszármazottjait NULL pointernek.

node* insertleft – Adott csomópontnak beállítja a bal oldali leszármazottját.

node* insertright – Adott csomópontnak beállítja a jobb oldali leszármazottját.

node* lepes – Lehetővé teszi, hogy a bináris fán lépni tudjunk, nyilván paraméterként kap egy irányt a függvény. Majd a gyökért beállítja az eredeti gyökér adott irány béli leszármazottjának.

void mesel – A dinamikusan tárolt tömbből kinézi, hogy a jelenlegi indexhez melyik story tartozik, és kiírja standard outputra.

karakter vizsgal – Megnézi, hogy a jelenlegi csomópontoz tartozik egy szignifikáns harcidx/boltidx. Ha nem, nem történik semmi, megy tovább a program.

karakter csata – Amennyiben tartozott 0-tól különböző harcidx, meghívja ezt a függvényt. Ez a függvény tartalmazza a harcot. A harc nem igényel a felhasználótól semmilyen inputot, ezzel kényelmessé téve a dolgát, csupán a harc menetét és kimenetelét látja. A harc úgy néz ki, hogy mindkét fél dob két dobókockával. Külön-külön összeadjuk a dobott számokat, ehhez hozzáadjuk a szemben álló felek ügyesség pontjait, és amelyik fél megnyeri a

„kört” az megtámadja a másikat, ezzel az ellenfél 2 életerőt veszít. A harcnak akkor van vége, ha valamelyik fél elveszíti az összes életerejét.

karakter penzes -Amennyiben tartozott 0-tól különböző boltidx, meghívja ezt a függvényt. Ez a függvény tartalmazza a játékban a mindenféle pénzzel kapcsolatos dolgokat. A boltidx meghatározza, hogy pontosan mi is fog történni. Itt nem feltétlen fog táskánk új tárgyakkal bővülni, elképzelhető, hogy csak pénzt veszítünk, hogy pénzt veszítünk, de a történetben tovább haladunk, illetve, hogy pénzt szerzünk(bár ez remélhetőleg ritka lesz). Mindenesetre, fizetési szituációknál eldönthetjük, hogy fizetünk-e, ezt a program mindig megkérdezi majd, hogy mit is szeretnénk csinálni/mit fogunk dönteni.

void mivanbenne -Megnézhetjük táskánk tartalmát

karakter kaja -Ehetünk ezzel felemésztve egy élelmiszert, mely a táskánkba volt eltárolva, viszont életerőnket tudjuk így növelni.

Jelenleg minden más a mainben található, az átláthatóság kedvéért: az alap függvények meghívása, illetve a választásokat kezelő részek.