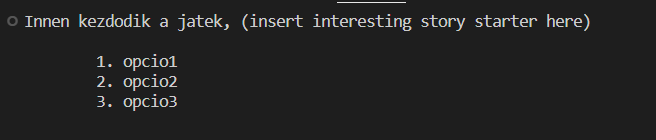
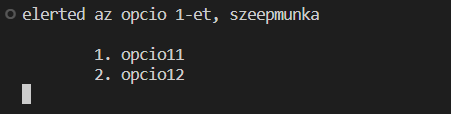
**Kalandozások Középföldén**

A program célja a szórakoztatás, akit érdekel Tolkien fantasy világa, annak biztosan remek időtöltés lesz ez a kalandozás.

Felhasználói dokumentáció

A program egy szöveget ír ki a felhasználónak, és opciókat ad a kaland folytatására. A felhasználó ezek közül az opciók közül választhat.

A felhasználó az opció számát beírva tud választani a lehetőségek közül. A történet ennek megfelelően folytatódik.

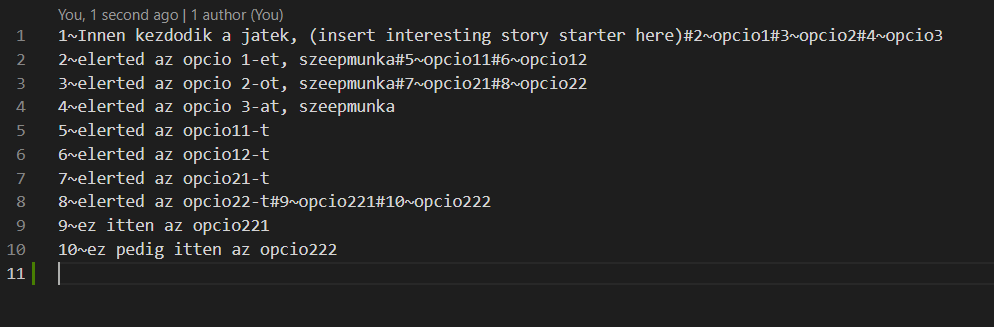
Tegyük fel, hogy a felhasználó beírt egy 1-est. Akkor a kimenet a következőképpen alakul:

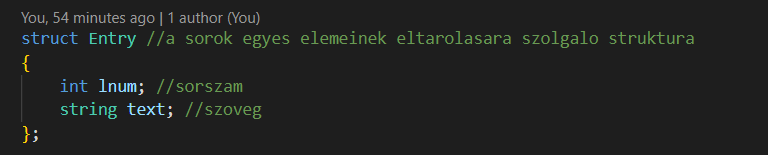
A játék így folytatódik egészen addig, amíg nincs több választási lehetőség.

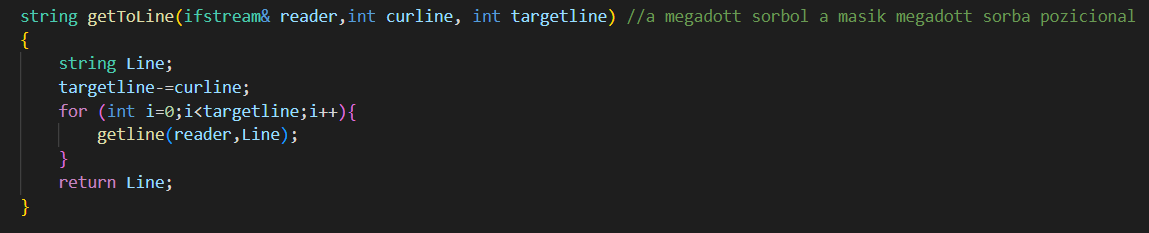


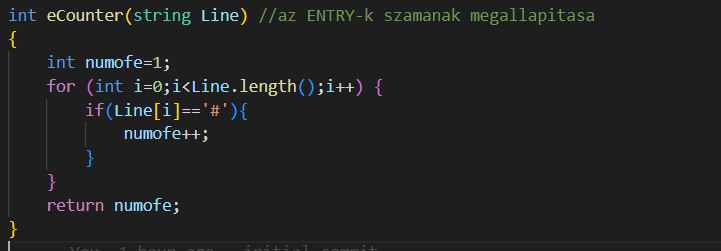
A következő oldalakon a fejlesztői dokumentáció kerül bemutatásra

Fejlesztői dokumentáció

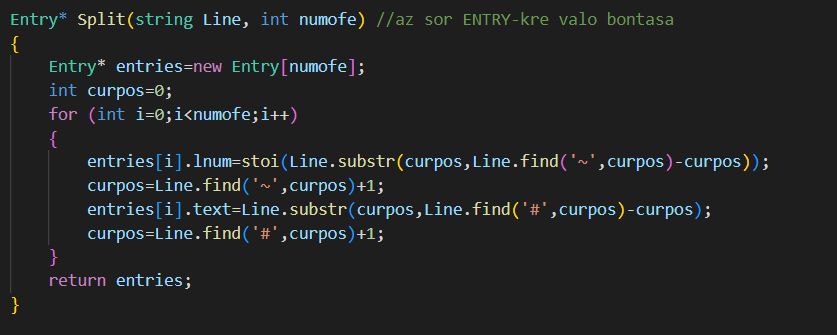
A történet egy szöveges fájlba van eltárolva. A tilde jel előtti számok azonosítják a sorokat, az egyes opciók #-el vannak elválasztva a kiírni való szövegtől.

Az egyes kiírandó szövegeket (és a választható opciókat) a program az Entry nevezetű struktúrában tárolja el, ahol minden objektumnak van egy azonosító száma, ami arra utal, hogy melyik sorban található meg és van egy stringje ami magát a szöveget tárolja el.

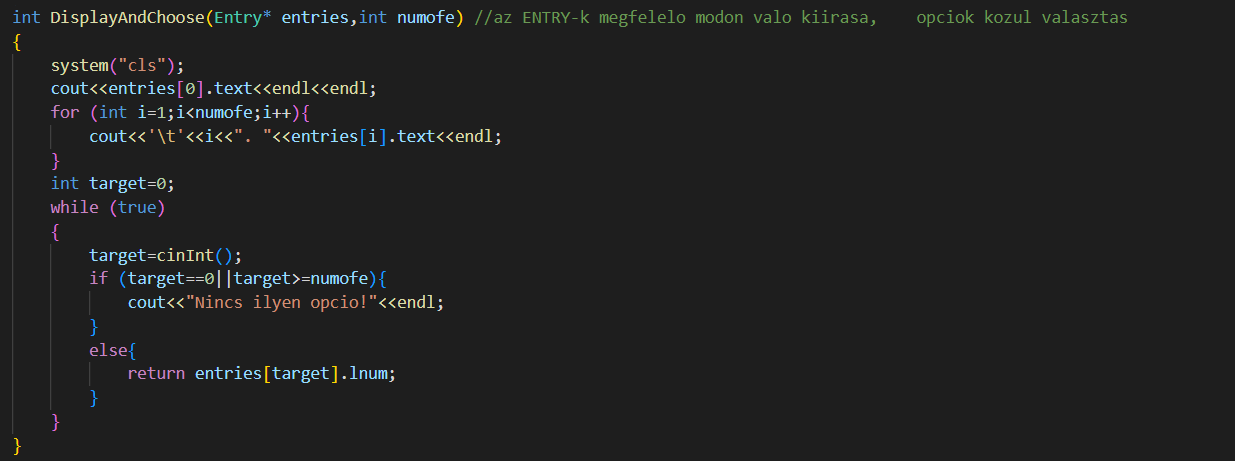
Az első meghívott függvény a getToLine (), ami paraméterként megkapja az olvasó fájl streamet ,annak aktuális pozícióját (hányadik sorban van) és a célpozícióját (hányadik sort kell kiolvasnia). Eredményül a kiolvasott célsort adja vissza.

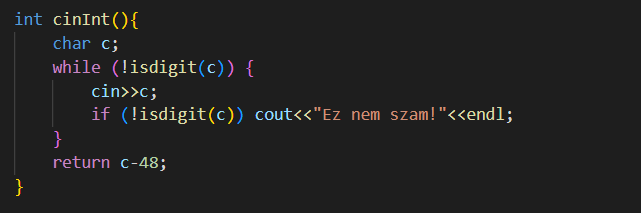
Ezek után meghívásra kerül az eCounter () nevezetű függvény, ami a sorban található #-ek segítségével megállapítja, hogy azt hány részre kell darabolni. Eredményül a darabszámot adja vissza int formátumba.

Ezek után következik a beolvasott sor feldarabolása, amelyet a Split nevű függvény hivatott elvégezni. A függvény paraméterként várja a beolvasott sort, valamint eCounter () által megállapított darabszámot. Első lépésként a függvény dinamikus memóriakezeléssel létrehoz egy Entry tömböt. Ezek után a ~ és a # jelek segítségével feltölti ezt a tömböt olyan módon, hogy a szövegek és hozzájuk tartozó sorszám ugyanazon objektum megfelelő részeibe kerülnek. Eredményként az Entry tömb mutatóját adja vissza.



Az utolsó, a játék interaktív részét képező függvény a DisplayAndChoose (). Eme függvény paraméterként várja a Split () által visszaadott Entry tömböt, valamint a tömb elemeinek a számát. Kezdetben kiírja a szöveget és a választható opciókat megfelelő módon (ld. felhasználói dokumentáció).

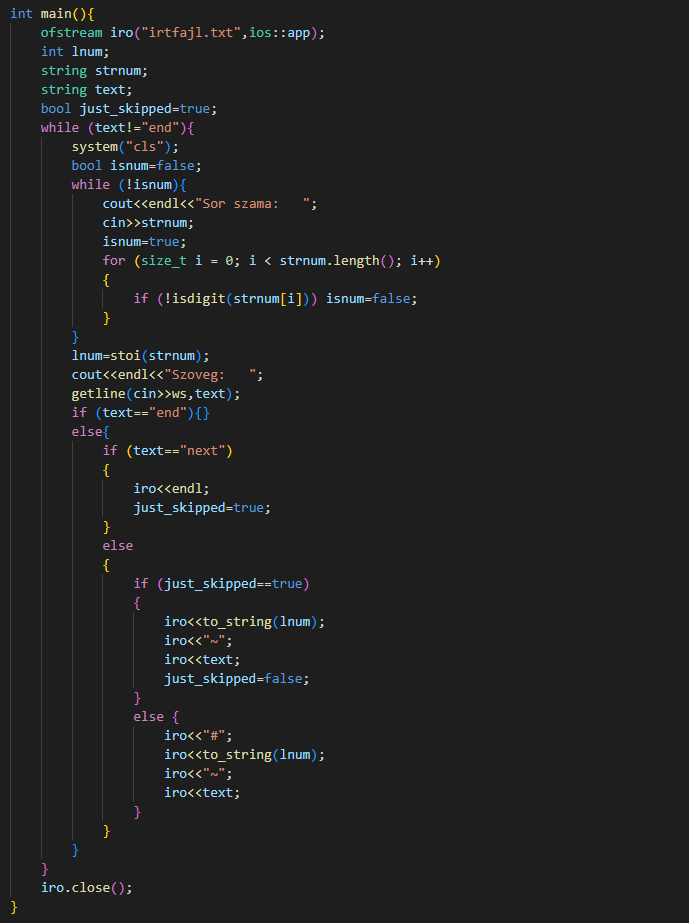
Ezek után elindul ciklus, ami addig fut amíg nem lesz a függvénynek visszatérési értéke (a felhasználó a kiírt opciók valamelyikének a számát adja meg).

Ennek a függvénynek van egy segédfüggvénye az úgynevezett cinInt (), ami elvégzi a beolvasást és egyúttal ellenőrzi, hogy a beolvasott karakter szám-e.

A main a következőképpen néz ki:

A main függvény főciklusa addig fut, amíg olyan esetbe nem ütközik, hogy nincsen a játékos számára választható opció (csak a kiírt szöveg van). Ebben az esetben a ciklus kilép és a játéknak vége.

A történet tárolására szolgáló szöveges fájlnak megszerkesztésében egy másik program segít, ami std::cin segítségével felveszi a szövegeket és a hozzájuk rendelt sorszámokat. Ezek után a fájl író stream számok, szövegek, ’~’ és ’#’ szimbólumok megfelelő sorba rendezésével készíti el a szöveges fájlt.

Ennek a programnak a forráskódja a következő: