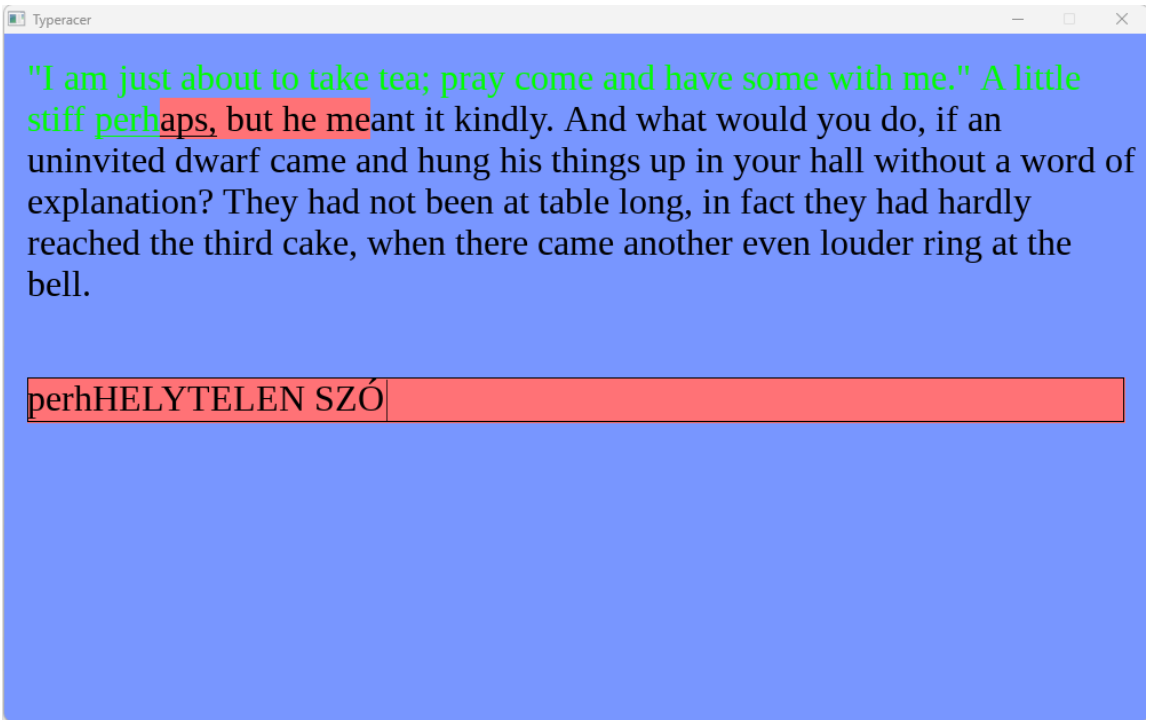
Typeracer

Szerző: Gaál Botond
Neptun: CRQEYD
Dátum: 2023.11.12.

Nagy HF félkész részfeladat

A program SDL könyvtár felhasználásával készült, így annak futtatásához és fordításához szükség van az SDL-re: https://infoc.eet.bme.hu/sdl_telepito/.

A félkész állapotban a Typeracer játéknak a 'motorja' készült el: generálódik egy véletlenszerűen kiválasztott szövegrészlet, amelyet a játékosnak be kell helyesen írnia.



Egy-egy szöveget egy-egy dinamikusan foglalt tömbbe tárolunk el, mint stringek tömbje, szavakra bontva, ez a Text típus. Definícója, valamint az összes szöveg generáláshoz tartozó függvény a text.c és text.h fájlokban található. A szövegek pedig szintén egy dinamikus tömbben vannak tárolva, amely a TextArray típus. Ami után már nincs rájuk szükség, fel kell szabadítani a tömböket a free_textarray függvénnyel. Az inicializálás után tehát egy TextArray-t kell feltölteni Text-ekkel, azokat pedig szavakkal. Ehhez a bemenő fájlban előre meg van adva, hány szóból áll egy Text. A különböző szövegrészletek tehát egy WordCount-al kezdődnek, és \n karakter jelzi a végüket, például:

WordCount: 67
Took ancestors must have taken a fairy wife…. \n

Miután betöltöttük a szövegeket tartalmazó fájl tartalmát egy TextArray-be, abból véletlenszerűen kiválasztunk egy elemet, mint játék célt, és belépünk az eseményhurokba, ahol azokat az eseményeket figyeljük, amik arra adnak okot, hogy a szöveget újra ki kelljen rajzolni:

- Gomb lenyomás: két megkülönböztetett billentyű van. A backspace, mivel ezzel módosítjuk az inputot, valamint a space, mivel, ha a begépelendő szó és az input mezőben lévő szó megegyeznek, akkor léptetni kell a szövegben a szó számlálót.
- Szöveg bevitel
- Következő szóra léptetés

Minden alkalommal, mikor elkapunk egy eseményt, újra rajzolódik a szöveg beviteli doboz is, és a begépelendő szöveg is: ezt a render_input és a render_Text függvények végzik. A render_input kirajzolja a beviteli dobozt, vagy fehérrel vagy pirossal, attól függően, helyes-e a gépelés, majd rár rajzolja arra a begépelt szöveget. A render_Text végig iterál a szöveg szavain, eldönti, hogy ugyanabba a sorba kifér-e a szó, majd ennek fejében rendereli őket, megfelelő kinézettel:

- zöld a szó, ha már beírtuk helyesen
- alá van húzva az a szó, amit éppen be kell gépelni
- fekete az a szó, amihez még nem ért el a játékos
- helytelen beírás esetén annyi szöveget kell piros háttérrel renderelni, ahány helytelen karakter van az input stringben

Egy-egy szó a render_string_blended, vagy render_string_shaded függvénnyel kerül renderelésre, amelynek a visszatérése a négyszög, amibe beleírjuk a szót: ez használható arra, hogy tudjuk, a következő szót mennyivel kell odébb kezdeni.

Ha végig ért a szavakon a játékos, kilépünk a while ciklusból, be lehet zárni az SDL által létrehozott ablakot, és fel lehet szabadítani a foglalt memóriát.