**Montessori tornyok**

**Készítette:**

* Oláh Gergő
* Vasilescu András
* Gombos Benedek

[Bevezetés 1](#_Toc917222746)

[Ötletszerzés/Találás 2](#_Toc1034964650)

[Tervezés 2](#_Toc708597995)

[Megvalósítás 2](#_Toc239476315)

[Tesztelés 2](#_Toc398443133)

[Felmerülő problémák 2](#_Toc1935905976)

[Összegzés 2](#_Toc2083298037)

[Értékelés 2](#_Toc1088691786)

[Lezárás 2](#_Toc1304514254)

[Felhasznált irodalom 2](#_Toc1423995871)

# Bevezetés

Csapatunk újabb projektje egy kódbázisról választott feladat volt.

JavaScript feladatnak kellett lennie, ahol tudtuk alkalmazni az eddig tanul JavaScript, HTML, CSS tudásunkat egy nehezebb, moderatív feladatnál.

Főképp a JavaScript tudásunkat kellett előtérbe hejeznünk ebben a feladatban.

Több, mint 12 külön feladat közül tudtunk választani, melyek közül mindegyik lényege a weboldal készítés volt csak néhol a JavaScript, máshol pedig a HTML főbb tudása volt a prioritás, de az összes lényege az volt, hogy az eddig szerzett tudásunkat érvényesítsük egy haladó típusú weboldal készítési feladatban.

# Ötletszerzés/Találás

Az ötletünket a kódbázis oldalról kellett kiválasztanunk, mivel a feladat célja volt, hogy arról az oldalról egy kiválasztott minta alapján készítsük el a feladatunkat.

A csapatunk, először egy regisztrációs feladatot nézett, mivel kihívóan nézett ki és egy olyan oldalnak melyet nehezen, de megbírnánk csinálni, ámde ezt a feladatot már más csapat kiválasztotta így értelmetlen lett volna egy és ugyanazt a feladatot ketten csinálni.

Végül egy Montessori torony nevezetű oldal lett a választásunk, mivel senki nem akarta bevállalni nehézsége miatt, ezért döntöttünk úgy, hogy akkor majd mi megcsináljuk.

A példa nehézségét látván egy haladó szintű feladatnak tűnt, így minél nagyobb volt a kihívás annál többet tudunk belőle tanulni, ez volt a fő indok a választás és ötletkeresésnél.

# Tervezés

A tervezés kapcsán figyelembe vettük főképp a feladat nehézségét, mivel tudtuk, hogy a választott példa egy nagyobb szintű feladat, mint amivel eddig dolgoztunk.

Először leszögeztük az alapokat, kell egy CSS, egy JavaScript és persze a HTML oldalunk, 3 torony kellett, amit megadunk alapnak és ezek lesznek azok, amiket változtatni fogunk egy függvénnyel JavaScript kódban.

A 3 torony randomizáltan fog legenerálódni és úgy lekészítünk majd 3-9-ig blokkokat a tornyokon, ezeket vagy “Drag-on” egér mozgatással, vagy “On-click” kattintással lenne a blokkok mozgatása.

A Drag-on opció egy nehezebb, de látványosabb lenne a sima On-click opció meg egy egyszerű, de szimpla döntés lenne.

Tervezésnél meg előre megbeszéltük, hogy egy Excel fájlba leírjuk a hibákat és elvárt eredményeket ezzel megkönnyítve a jövőbéli munkánkat.

JavaScriptre tudtuk, hogy nagyobb figyelmet kell fektetnünk, így emiatt úgy terveztük, hogy a kódolás részét együtt csináljuk, de közben persze foglalkozunk a tesztdokumentációval is, így egyszerre vagyunk képesek két feladatot nyugodtan és megfelelően elkészíteni, két legyet egy csapásra.

Dokumentációra meg elég egy ember is, így azt csak egy személynek osztottuk ki.

# Megvalósítás

A feladatok kiosztása után jött, hogy milyen sorrendben haladjunk, először úgy döntöttük el, hogy elkezdjük és majd út közben kiderül mit prioritizáljunk, végül a legtöbb feladatot együtt végeztük.

Bővebben menjünk bele a részletekbe.

1. **Kódolás**

Kódolás során először értelemszerűen a HTML kódot kellett megírnunk, mivel, ha nincs alap oldalunk mire írjuk a CSS-t és JS-t? HTML kódot nem akartuk túlbonyolítani így csak az oszlopokat raktuk bele és egy alap indexet adtunk mindegyiknek, illetve a JS-t és CSS-t hozzákötöttük és lényegében a HTML része befejeződött.

Következő lépés a JS volt, megakartuk csinálni az alap folyamatokat, bár itt még dilemmába voltunk a generálás szempontjából. Gondolkodtunk a tervben megoszlott véleményen és randomizáltan oldottuk meg az oszlopok legenerálását, így jóval több választék volt arra, hogy hogyan rendeződnek a blokkok így talán könnyebb vagy nehezebb játékmenetet adva számunkra. Következő volt a blokkok úgymond “irányítása”. Tervben megbeszélt Drag-on ötlet tűnt a legjobbnak így az mellett döntöttünk, ám a megvalósítása már okozott némi problémát, mivel néhol az indexelt elemet nem észlelte a function, de ezt letudtuk egyből, mivel rossz index volt megadva egész végig.

Végül a JS kódunkat is befejeztük, sok feladattal és megmérettetéssel, de bebírtuk fejezni és léphettünk is az utolsó részre a kódolásban.

Utolsó és nem nagyon említésre méltó rész a CSS volt. A CSS-ben formáztuk meg a, oszlopokat, blokkokat és úgymond az egész oldalunkat. Hátteret adtunk neki, a blokkoknak és az oszlopoknak egy rendes formát és kinézetet, illetve kicsit jobban szétszedtük őket, hogy ne legyenek annyira közel egymáshoz így több tér legyen a mozgáshoz. Reszponzívvá tettük a weboldalt, hogy bármilyen eszközt is használunk, akkor is megfelelő formában láthassuk a “játékunkat”.

És így a kódolás résszel végeztünk.

1. **Dokumentálás**

Dokumentálás során 2 dokumentáción dolgoztunk miközben a kódolás zajlott.

A tesztdokumentáció Benedek és András írta főképp, mivel ahogy írták a kódot és ahogy javították a hibákat vagy netán kezelték őket úgy tudták közbe a “listába” írni azt, amit találtak és eközben meg íródott a sima Projekt dokumentáció is, de a tesztdokumentáció csak a végén került bele a főbb dokumentációba.

A Projekt dokumentációt 8 részre szedtük, hogy bővebben bele bírjunk menni a részleteke és kibírjuk fejteni azt, ahogy dolgoztunk. Bevezetéstől kezdve a lezárásig le akartunk írni mindent, hogy a folyamat is teljesen átlátható, illetve megérthető legyen. A dokumentációt kódolás közben is írtuk, ahogy említve lett, mivel sokszor előfordult, hogy változtatni kellett rajta, mert lehet más döntésünk lett, máshogy terveztük meg a dolgokat, vagy netán máshogy valósult meg a tervünk, így nem akartunk badarságot írni a dokumentációba ezért számtalanszol változott megvalósítás és még lezárás közben is.

# Tesztelés