**ENERGETIKAI TECHNIKUM ÉS KOLLÉGIUM**

**Arkanoid**

**2023.**

|  |  |
| --- | --- |
| **Konzulens:** | **Készítette:** |
| **Tóth Zoltán** | **Molnár Csaba** |

Nyilatkozat

Alulírott Molnár Csaba büntetőjogi felelősségem teljes tudatában nyilatkozom arról, hogy az itt szereplő záródolgozat saját, önálló munkám eredménye és sem részeiben sem egészében nem került még kereskedelmi forgalomba, ill. publikálásra, a GPL licenszelésű programrészek kivételével.

Paks, 2023. április 14.

Molnár Csaba

Tartalomjegyzék

[I. Bevezetés 4](#_Toc1)

[II. Témaválasztás indoklása 4](#_Toc2)

[III. Fejlesztői dokumentáció 5](#_Toc3)

[1. Specifikáció 5](#_Toc4)

[2. Az alkalmazott fejlesztői eszközök 5](#_Toc5)

[3. Adatmodell leírása 6](#_Toc6)

[4. Algoritmusok 7](#_Toc7)

[Fizika 7](#_Toc8)

[Téglák 8](#_Toc9)

[Játékmódok 9](#_Toc10)

[Pontrendszer 10](#_Toc11)

[Időmérő 11](#_Toc12)

[Adatbáziskezelés 12](#_Toc13)

[5. Forráskód 13](#_Toc14)

[6. Tesztelési dokumentáció 15](#_Toc15)

[Fizika teszt 15](#_Toc16)

[IV. Felhasználói dokumentáció 15](#_Toc17)

[1. A program általános specifikációja 15](#_Toc18)

[2. Rendszerkövetelmények 15](#_Toc19)

[3. A program telepítése 16](#_Toc20)

[4. A program használatának a részletes leírása 18](#_Toc21)

[V. Összegzés 19](#_Toc22)

[VI. Irodalomjegyzék, hivatkozásjegyzék 20](#_Toc23)

# Bevezetés

A záródolgozatom témája egy videojáték, amely az Arkanoid című klasszikus játéktól vesz erős inspirációt, a saját design ötleteimmel felfrissítve.

Feladatomat interaktív weboldal képében valósítom meg, ennek érdekében pedig természetesen HTML, CSS, illetve JavaScript nyelveket fogok alkalmazni. Annak az oka, hogy nem C# illetve Python nyelvet használok, hogy bár objektum orientált megoldást szeretnék kivitelezni, a Windows Forms nem ad elegendő rugalmasságot a felhasználói interfész kidolgozására, valamint személyes véleményem szerint egyik nyelv sem megfelelő videojáték fejlesztésre.

A CSS kellően sokszínű és könnyen használható, a JavaScript pedig rendelkezik azokkal az objektum orientált eszközökkel, amelyek megkönnyítik a feladat elvégzését.

# Témaválasztás indoklása

Bár a korombéli fiatalok a kétezres évek végét, tízes évek elejét a legújabb Xbox 360 illetve Playstation 3 konzolokkal töltötték, az én családomnak egy nyolcvanas évek béli Nintendo rendszer jutott. Ennek következménye, hogy leginkább a már akkor retrónak számított játékvilággal ismerkedtem meg, és ez a periódus maradt az érdeklődési köröm központjában évekkel később is.

A záródolgozat feladatkiosztásakor azonnal tudtam, hogy valamiféle videojáték programot szeretnék csinálni. Ennek több oka is van.

Először is, a segédprogramok illetve a webshopok világán belül kevés ötletem volt, hiszen az ilyesfajta problémákra aligha tudnék egyedi és kitűnő megoldást találni. Továbbá az is fontos, hogy a hónapokon át tartó fejlesztési eljárást ne unjam meg, hiszen, ha elmegy a kedvem a szoftver kidolgozásától, akkor azzal a szakdolgozatom is bukik.

Ez okból célratörő lenne a játékfejlesztés, hiszen a debugolás és az új funkciók hozzá adása alatt állandóan játszanom is kellene, ami egyben feltárja a program hibáit, és szórakoztat is.

A téma tehát megvan. Már csak azt kell leszögezni, milyen stílusú játékot szeretnék fejleszteni.

Az első gondolatom egy bullet-hell shoot-‘em-up akciójáték volt, hiszen a feladat kiosztása idején ezzel a műfajjal foglalkoztam. Ez sajnos problémát jelentett, hiszen ilyen téren semmiféle tapasztalatom nincs, és rengeteg rendszert el kellett volna sajátítsak rövid időn belül ahhoz, hogy ez megvalósuljon. Gondolkodtam még szerepjátékon illetve platformeren, viszont végül a puzzle műfajon landoltam.

Ezen belül két nyilvánvaló lehetőségem volt. Egy Tetris inspirált véget nem érő kirakós, ami egyre nehezebb lesz az idő teltével, vagy egy Arkanoid inspirált, rövidebb terjedelmű de sokkal több lehetőséggel bíró projekt, amelybe bele fűzhetem a saját ötleteimet. Mivel egy osztálytársam a Tetris mellett döntött, elhatároztam, hogy a második opcióval élek.

# Fejlesztői dokumentáció

## Specifikáció

Ahhoz, hogy minden, amit 5 év alatt elsajátítottam megjelenjen a programomban, leszögeztem pár alap funkciót, amit implementálok:

* HTML alapú webinterfész.
* Hangulatos interfész kialakítása CSS használatával.
* Pontrendszer és ranglista, amelyet SQL adatbázissal raktározok el.
* Több játékmódos játék, amelyet a JavaScript kihasználásával valósítok meg.
* PHP használatával létrehozott dinamikus oldalak.

Úgy vélem, hogy a választott téma, és annak kivitelezése elegendően demonstrálja a szoftverfejlesztő képességeimet, illetve a problémák elegáns megoldásához való szakértelmemet.

## Az alkalmazott fejlesztői eszközök

Fejlesztői környezetnek Visual Studio Code-ot használtam, mivel egy webapp fejlesztésére tökéletes eszközökkel rendelkezik. Támogatja az összes nyelvet, amellyel megvalósítom az ötletem, beépített Git támogatásának köszönhetően könnyű rendszerezni a kódom, valamint bővítmények használatával az SQL szerverrel is egyszerűen tudtam kommunikálni.

A debugolás Firefox és Chrome webböngészőkben történt, a beépített fejlesztői eszközöket használva. Ezekkel tudtam a CSS stíluslap hibáit kiküszöbölni, valamint a változókat nyomon követni, ami a fizika kidolgozásához segített.

Mivel PHP és SQL is szerepel a projektemben, kellett egy webszerver ami minél több platformon elfut és felhasználóbarát telepítési procedúrával rendelkezik. Ezért választottam az XAMPP webszerver csomagot, amely önmagába foglal többek között egy Apache szervert, amin a PHP kód is elfut, és egy MySQL adatbázis kezelő programot.

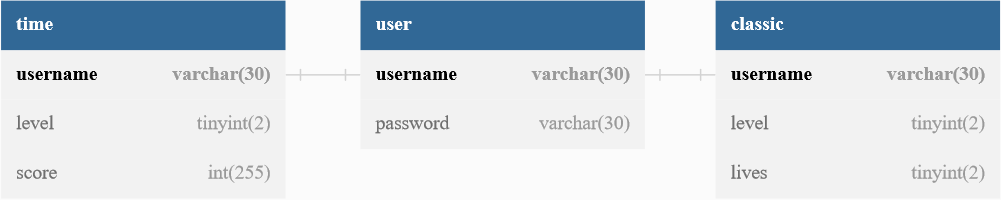
Ezen kívül az adatbázis kezelését megkönnyítette a phpMyAdmin grafikus interfész, amely szintén az XAMPP csomag tartozéka. Ezt használva tudtam figyelem alatt tartani a játékékosok adatait, a játék közben szerzett pontszámokat, és a különböző lekérések érvényességét.

Magát a dokumentációt OnlyOffice segítségével dolgoztam ki.

Külső modulokat vagy keretrendszereket nem használtam.

## Adatmodell leírása

A játékban szereplő adatbázis az entitás-kapcsolat modell alapján működik.



3 fő tábla létezik az adatbázison belül.

A „user” tábla a játékosok adatait tartja számon. Ezek az adatok a játékosok nevei (username), és jelszavai (password). A játékos nevet és a jelszavat a weboldal első oldalán lehet megadni. Amennyiben a megadott adatok alapján nem létezik felhasználó, a program hozzá adja az adatokat a táblához. Ha a felhasználónév létezik, de a jelszó helytelen, hibaüzenet fogadja a felhasználót. Ha mindkét adat létezik, a felhasználó belépést nyer az oldalra. A játék során elért pontszám felülírja az adott játékos rögzített pontszámát, feltéve, ha az elért szám magasabb, mint a rögzített szám.

A másik két tábla a két külön játékmód eredményeit követik.

A „classic” tábla tartalmazza a klasszikus játékmód adatait. Ezek az adatok a játékos neve (username), a legnagyobb elért pontszáma (score), és a szint amin a játékos elérte a rögzített pontszámot (level). Ez a tábla akkor frissül, ha a bejelentkezett játékos nagyobb pontszámot ér el, mint a rögzített érték, vagy ha nagyobb az elért szint.

A „time” tábla tartalmazza az időfutam játékmód adatait. Ezek az adatok a játékos neve (username), a legnagyobb elért szint (level), és a játékos hátra lévő életei az idő lejárta után (lives). Ez a tábla akkor frissül, ha a bejelentkezett játékos nagyobb szintet ér el, mint a rögzített érték, vagy ha több maradék élettel fejezi be a kört.

A zökkenőmentes felhasználói élmény érdekében a program automatikusan létrehozza az adatbázist és annak tábláit, ezáltal a felhasználó nem kell a phpMyAdmin felületén keresztül importálja azokat. A biztonság kedvéért eltároltam az adatbázist .sql formátumban, ha szükség lenne manuális importálásra.

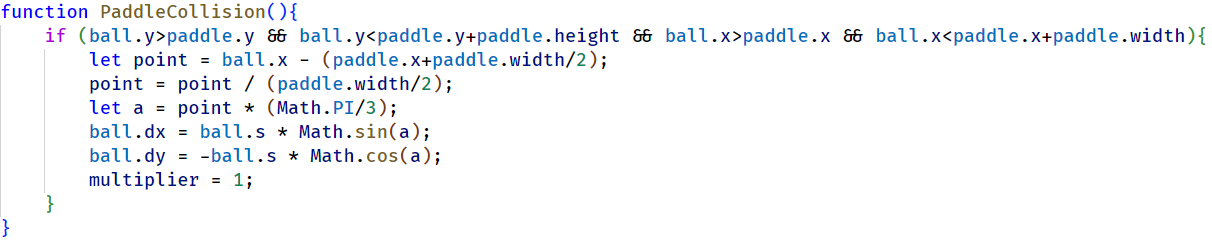
## Algoritmusok

A program megvalósításához ezek voltak a fő elvárásaim:

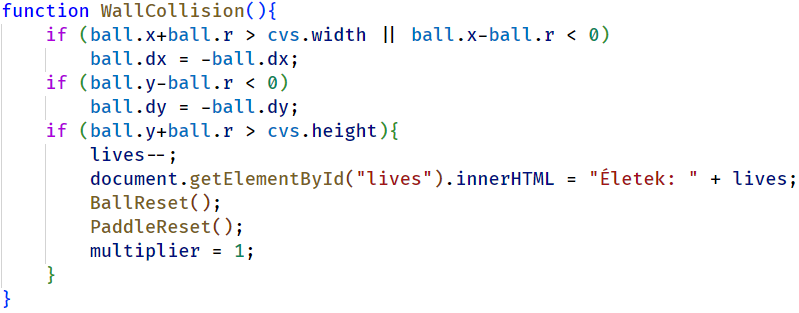
* Fizikai motor, amely megfelelően kezeli a labda mozgását és ütközését
* Téglák, amelyek ütközésnél törnek
* Választható játékmódok
* Pontrendszer
* Időmérő rendszer
* Adatbázis az eredmények követésére

### Fizika

A fizika két részből áll, ütközés ellenőrzésből, és vektor kiszámításból.



A program először ellenőrzi, hogy a labda koordinátái érintik-e azt a területet, ahol a játékos karakter helyezkedik el. Ha igen, kiszámítja pontosan hol találta el a játékost, majd ezt a találati pontot normalizálja -1 és 1 között. Ezt az értéket egy szinusz és koszinusz metódusnak átadva, a labda sebességét is bele számítva, megkapjuk pontosan milyen irányba milyen gyorsasággal kell pattanjon a labda.



A fallal való ütközet sokkal egyszerűbb, hiszen a szög (és ezáltal a sebesség) nem változik. Egyszerűen a falnak függvényében megfordítjuk a labda horizontális, illetve vertikális irányát. Kivétel az alsó fal, hiszen azzal ütközve életet vesztünk és a játék újra indul.

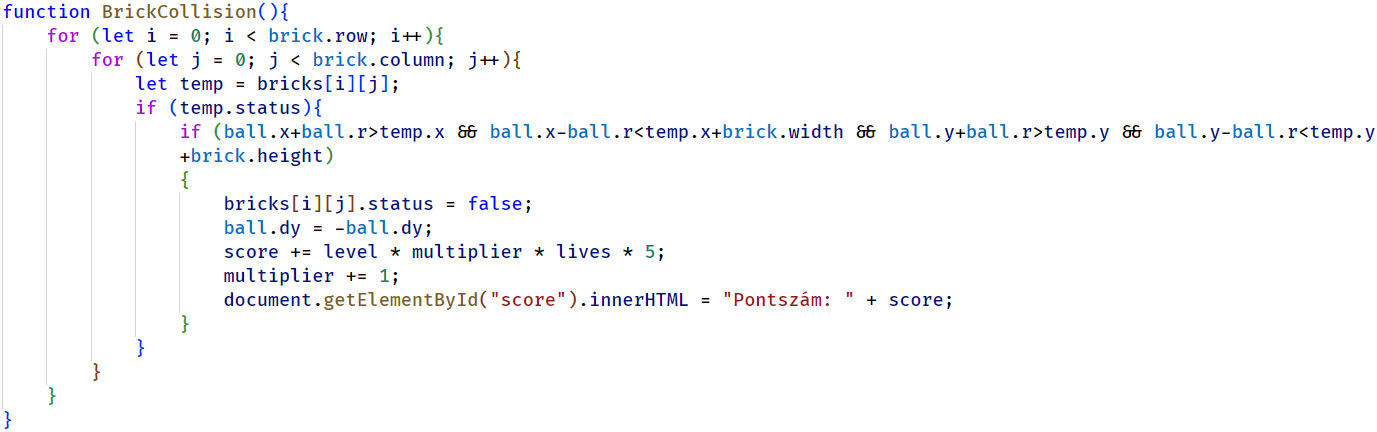
### Téglák



A téglák egy tömb változóban tárolódnak el. Ezen a tömbön a program sorokként és oszlopokként átmegy, majd az adott tégla elhelyezését kiszámítja annak méreteiből, bele számolva a köztük lévő kihagyásokat. Ezek után a tégla státusza true értékűvé változik, ezzel jelezve a tégla létezését.



Kirajzoláskor a program figyelembe veszi az aktuális szintet, és a téglák színével jelzi a nehézséget. Az első szint a legkönnyebb, így zöld téglákat rajzol, de ahogy egyre több sornyi tégla jelenik meg, a színeik átmennek sárgába, majd narancsba, és végül vörösbe.



A téglával való ütközés hasonló, mint a fallal való ütközés. A program ellenőrzi, hogy a labda koordinátái érintik-e a tégla koordinátáit, és ha igen, a tégla státusza false értéket kap, ezzel effektíve megszűnik létezni. Ezen kívül a labda iránya is megváltozik.

### Játékmódok

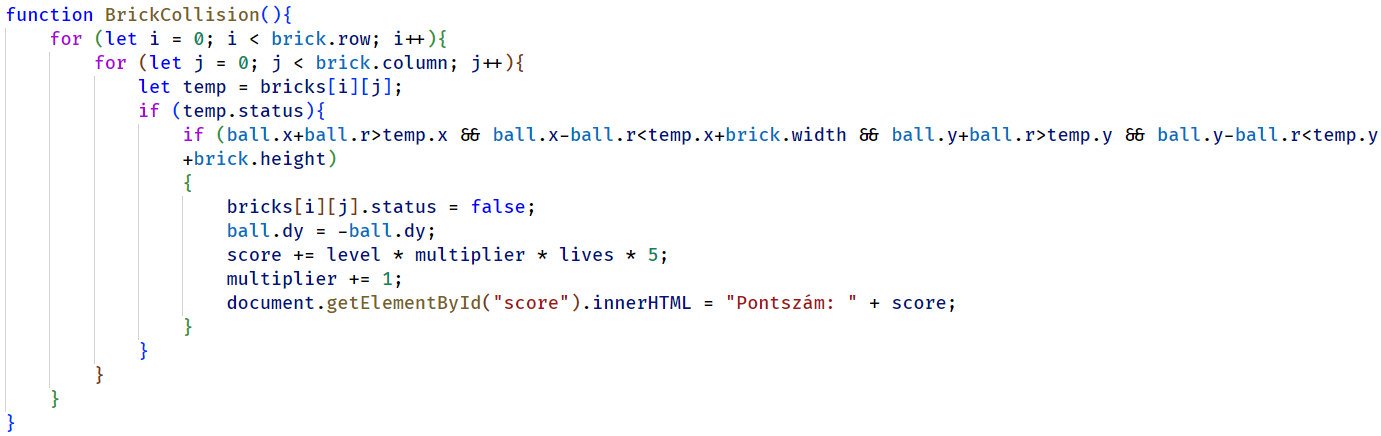
A felhasználónak két játékmód között van választása. A két mód ugyan arra a kód bázisra alapul, de eltérő szabályok alapján dől el a nyereség, illetve a veszteség.

Közös a két módban a fizika rendszer, a 3 élet, és a nehézség fokozatos növekedése a sebesség és téglák számának növelésével. Ezek voltak az alapok, amelyek befejezése után implementáltam a két választható módot.

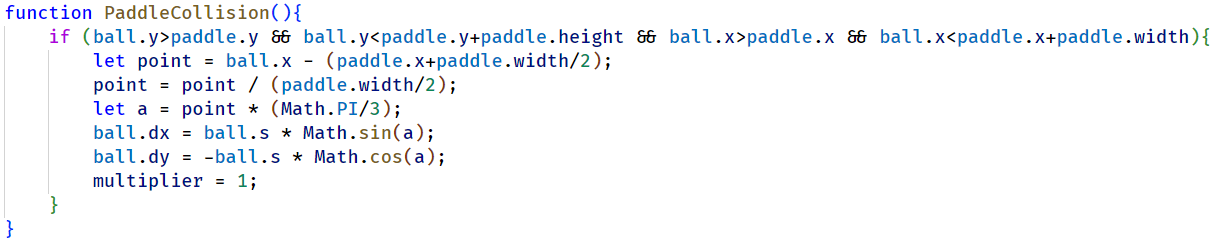
A klasszikus mód különlegessége a pontrendszer, míg az időfutam módban egy időmérő számolja a hátra lévő időt.

### Pontrendszer

A pontrendszer kidolgozásához nem külön álló metódusokat használtam, hanem a meglévő metódusokat alakítottam át.

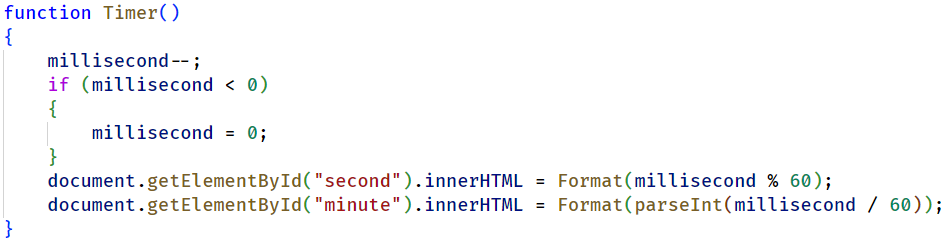


Egy tégla lerombolása után kettő dolog történik. Először is a program kikalkulálja azon tégla értékét, amit az alap érték (5), a maradandó életek száma és az aktuális szint szorzata dönt el. Ezt a számot még össze szorozzuk a pont szorzó értékkel. A szorzó minden tégla lerombolása után növekszik egyel, tehát ha egymás után rombolunk le több téglát, a pontszám drasztikusan megugrik.

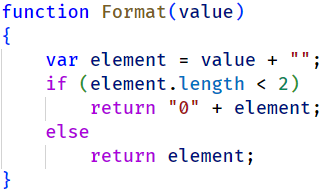


Ez a szorzó visszavált az alap értékére, ha a labda ütközik vagy a játékos ütőjével, vagy az alsó fallal, ami élet levonást is jelent.

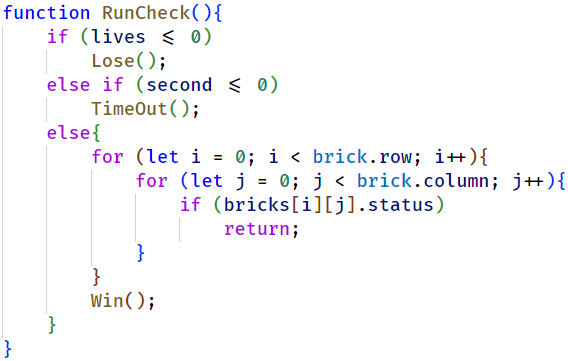
### Időmérő



Az időmérő rendszer fő metódusa másodpercenként egyszer fut le, ezzel számolva vissza a másodperceket. Egy kör 5 perces, tehát 300 másodpercet kell vissza számolnia.



Segítő metódus a Format, amivel konzisztensebb az időmérő megjelenése a weblapon. Egyszerűen hozzá fűz egy nullát a perc, illetve a másodperc értékhez, ha érzékeli, hogy a szám egyetlen számjegyű.



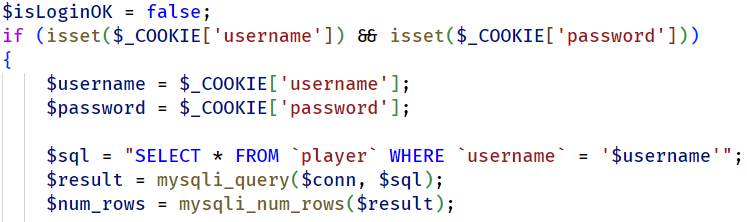
Az időmérő mód a klasszikussal ellentétben többféleképpen is véget érhet. A játéknak vége van, ha elfogy a játékos összes élete, vagy ha lejár az idő. Az életek elfogyása ebben az esetben nem vezet az eredmények elmentésére. A

### Adatbáziskezelés

Ahhoz, hogy a program követni tudja a játékosok eredményeit, először ki kellett dolgozzak egy bejelentkezési rendszert. Ez rengeteg fejfájást okozott, mivel nem tudtam a JavaScript program és az SQL webserver között adatokat cserélni, hiszen a JavaScript kliens oldali, míg az SQL szerver oldali nyelv.

Végül süti alapú megoldást választottam. A sütik olyan adatok, amiket a webszerver a felhasználó gépén tárol el, és tudja őket módosítani, illetve olvasni. Ez tökéletes összeköttetést biztosít a szerver és kliens között.

A weblap első megnyitásakor ellenőrzi, hogy van-e elmentve a bejelentkezéshez szükséges süti adat, a felhasználónév és a jelszó.



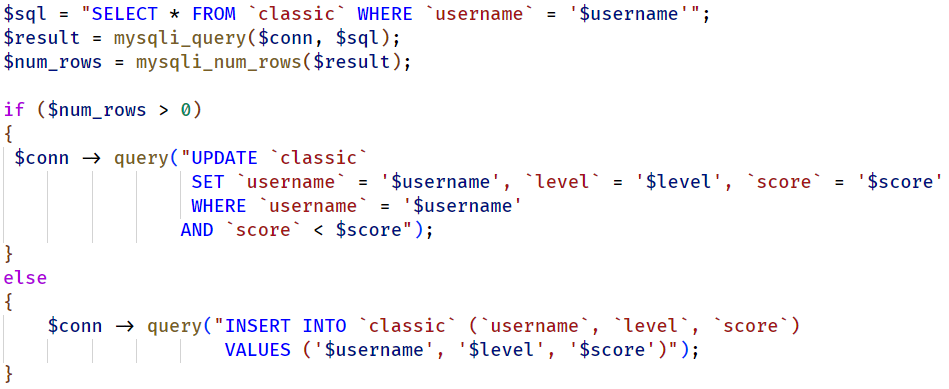
Ha a program talál megfelelő süti adatokat, egy SQL lekérdezés futtatásával ellenőrzi, hogy az adott felhasználónév létezik-e a játékosok adatbázisában. Ha a program talál egyezést a felhasználónévre az adatbázisban, egy újabb SQL lekérdezés futtatás történik, ahol nem csak a felhasználónevet, hanem a jelszót is ellenőrzi.



Ha létezik ilyen felhasználó, és helyes a megadott jelszó, akkor egy bool változó true értékével jelezzük, hogy minden rendben van, és a felhasználó bebocsátást nyer a programba.

Ha létezik ilyen felhasználó, de a megadott jelszó nem egyezik meg a felhasználónévhez párosított jelszóval, hibaüzenet fogadja a felhasználót, és biztonsági és stabilitási okokból törlésre kerülnek az eddig elmentett sütik.

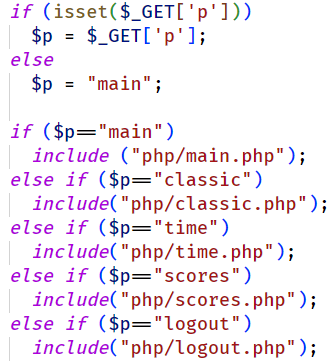
Ha sem a felhasználónév, sem a jelszó nem létezik az adatbázisban, mindkét adat regisztrálásra kerül, és a felhasználó bebocsátást nyer a programba.



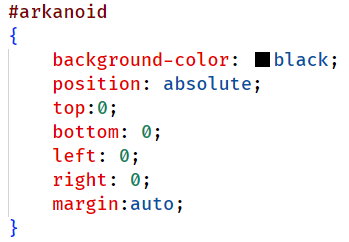
A játékmód táblák frissítése ugyan ezen az ötleten alapszik. Egy SQL lekérdezéssel ellenőrizzük, hogy létezik-e az aktuális felhasználó nevén rögzített eredmény. Ha igen, a program frissíti a megfelelő rekordot, feltéve, ha az új eredmény felülmúlja a rögzített eredményt. Ha nem, létrehozza a rekordot.

## Forráskód

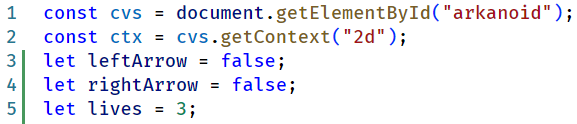
A weboldal dinamikus felépítésének köszönhetően nem kell minden HTML fájlba külön felépíteni az oldal elrendezését, illetve tartalmát. A webszerver által megnyitott index.php file tartalmazza az oldal alapjait, a navigációs sávot, a címsávot és a fő megjelenített tartalom ablakát. Ebbe az ablakba fog megjelenni az a tartalom, amit az aktuálisan megnyitott menüpont tartalmaz.



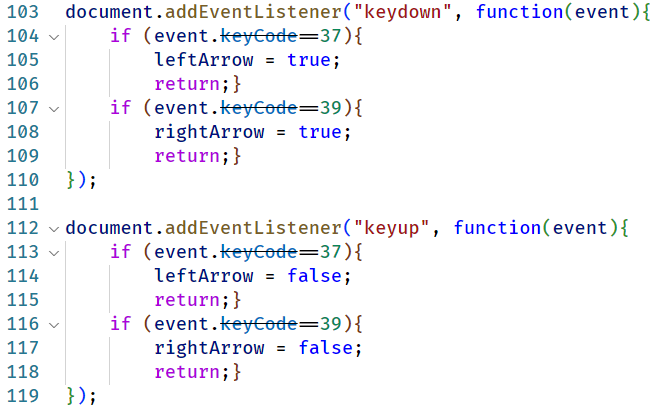
A retro érzet elérése érdekében 4:3 képarányba formáztam a canvas elemet, ami lehetővé teszi a 2D formák kirajzolását a monitorra. Az ezt követő script tag megidézi a játékmódhoz tartozó JavaScript fájlt.



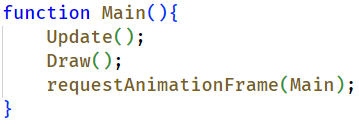
CSS stílus formázás csak minimális formában jelenik meg a projektben, törekedtem úgy tervezni a programot, hogy csak minimális korrigálás legyen szükséges.



A játéklogika kidolgozása előtt pár fontosabb változó van deklarálva, amik később fontos szerepeket töltenek be. Ilyen az élet számláló, az iránygombok státuszai, valamint a canvas context, ami a 2D elemeket rajzolja a képernyőre.



Az irányítás egyszerű EventListener metódus használatával működik. Gomblenyomáskor illetve felengedéskor ellenőrzi melyik gomb státusza változott, és az alapján egy bool változó értékét igazítja.



A fő metódus csupán a logika frissítését és a képernyőre rajzolás meghívását tartalmazza, majd egy animációs loopot hív aminek hála folyamatosan futni fog a program.

## Tesztelési dokumentáció

### Fizika teszt

Fallal való ütközéskor a labda a várt módon reagált, hibák nélkül. Megjegyeztem, hogy rendkívül magas sebességeknél a labda kitört a falak határán kívülre, de ez átlagos játékmenet alatt nem okoz problémát.

Játékossal való ütközéskor a labda eleinte váratlan hibákat mutatott, kiszámíthatatlan módon változott a sebessége. Ezt a hibát ki tudtam küszöbölni, és a labda gyorsasága az egész játék alatt konzisztens.

A téglákkal való ütközés szintén a várt eredményt mutatta. A tégla kollíziója megegyezik a faléval, majd ütközéskör a tégla példány eltűnik a játéktérről.

# Felhasználói dokumentáció

## A program általános specifikációja

Ez a rész a program fontosabb jellemzőit és funkcióit tartalmazza. A cél, hogy a leendő felhasználó ezt a fejezetet elolvasva el tudja dönteni, hogy a program megfelelő-e a számára.

Ajánlott terjedelem: 1-2 oldal.

## Rendszerkövetelmények

Hardver követelmények

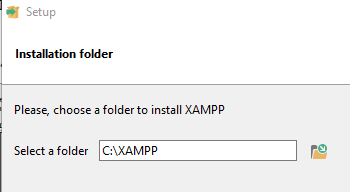
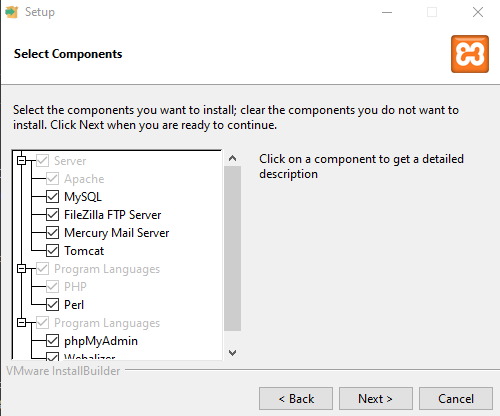
* **GPU**: Intel UHD Graphics 620 / AMD Radeon Vega 3
* **CPU**: Intel Pentium G7400 / AMD Ryzen 3 1200
* **RAM**: 4 GB DDR3

Szoftver követelmények

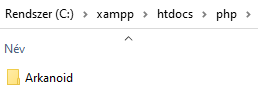
* **Operációs rendszer**: Windows 10 / macOS / Linux
* **Webszerver**: XAMPP (MySQL, Apache)
* **Webböngésző**: Chrome / Firefox

## A program telepítése

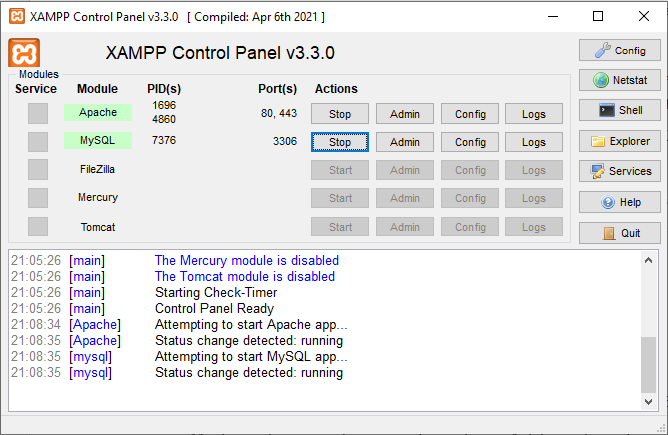
Az első lépés a webszerver telepítése. Bármilyen webszerver használható, de ajánlom az XAMPP által nyújtott Apache szervert, valamint a MySQL adatbázis programot. A CD-n található telepítő fájl futtatása során ki kell választani a szükséges komponenseket (a biztonság kedvéért hagyjuk mindet kiválasztva), majd meg kell adni a telepítési mappát (alapértelmezett elérési út a „C:/XAMPP/”).



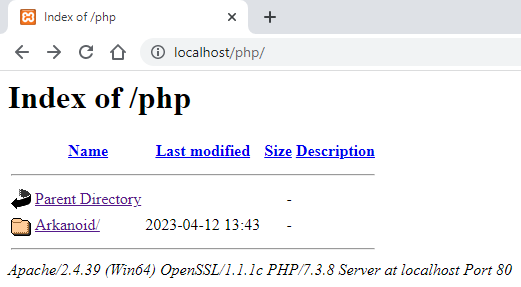
A telepítést követően helyezzük a játék mappáját a telepítési mappában lévő „htdocs/php/” mappába. Ha az alapértelmezett mappát, a teljes elérési út „C:/XAMPP/htdocs/php/”.



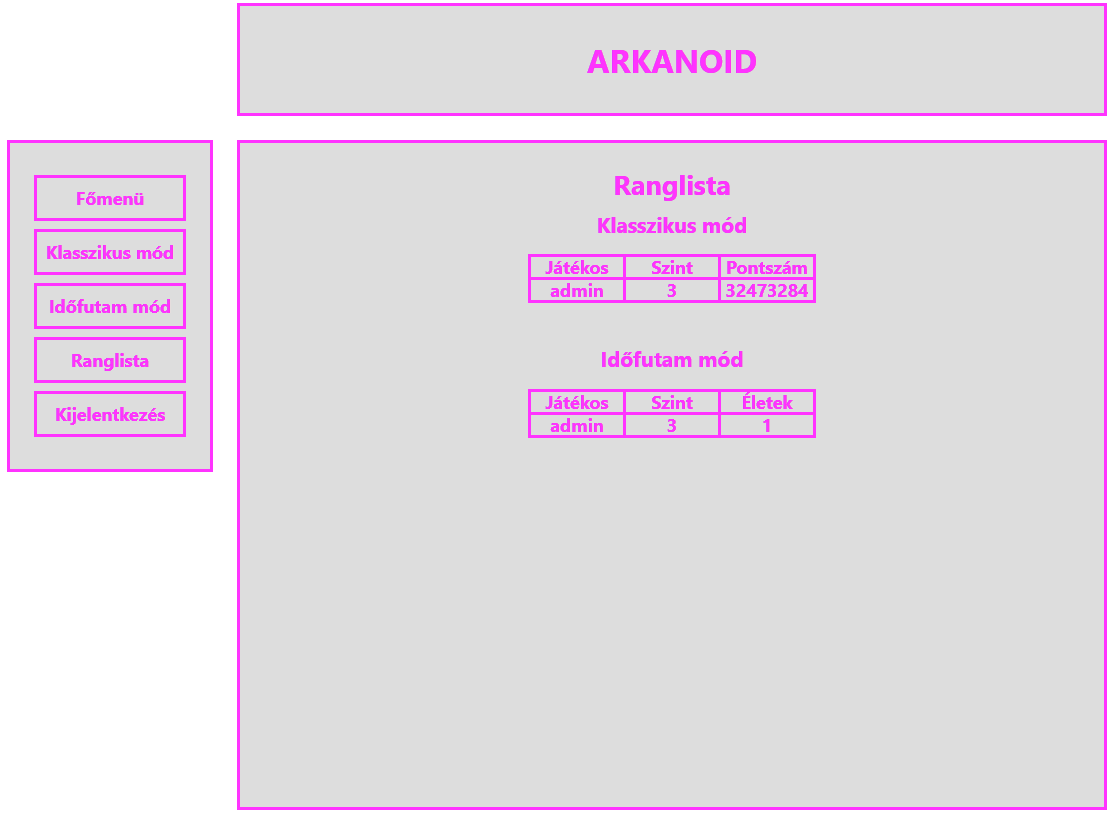
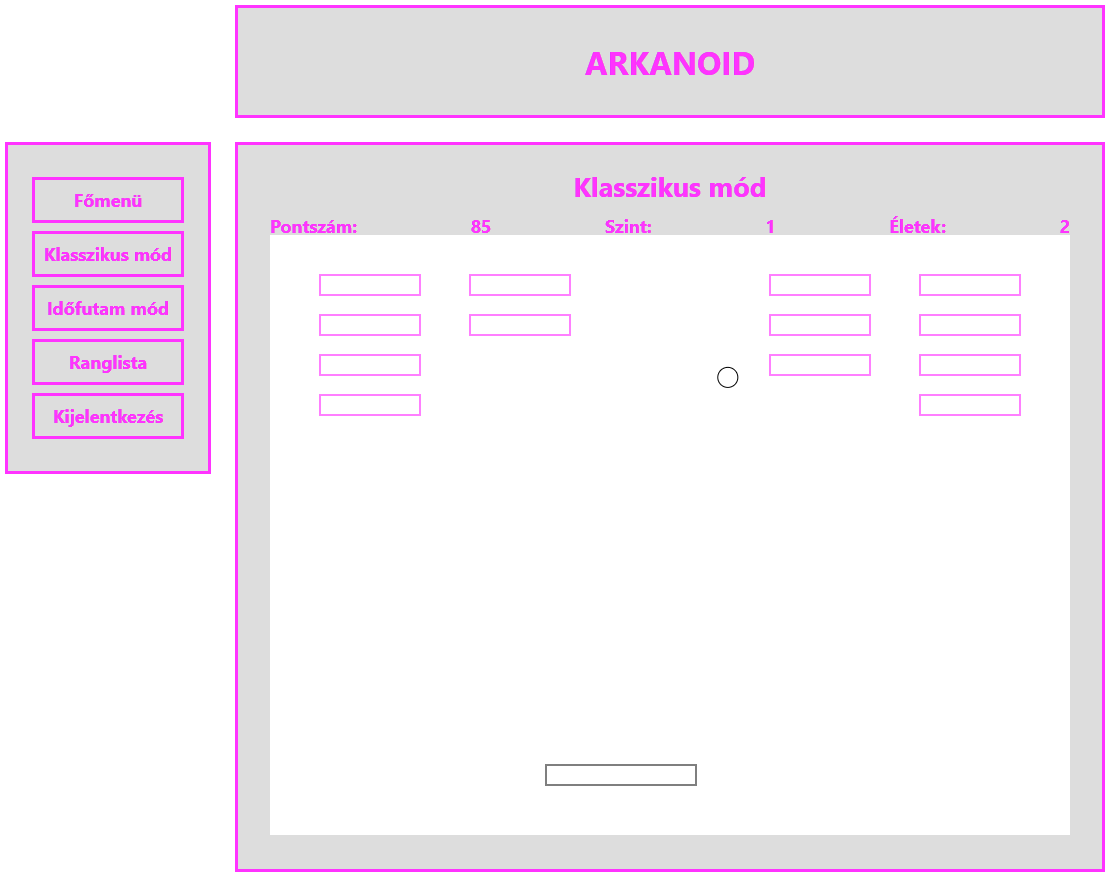
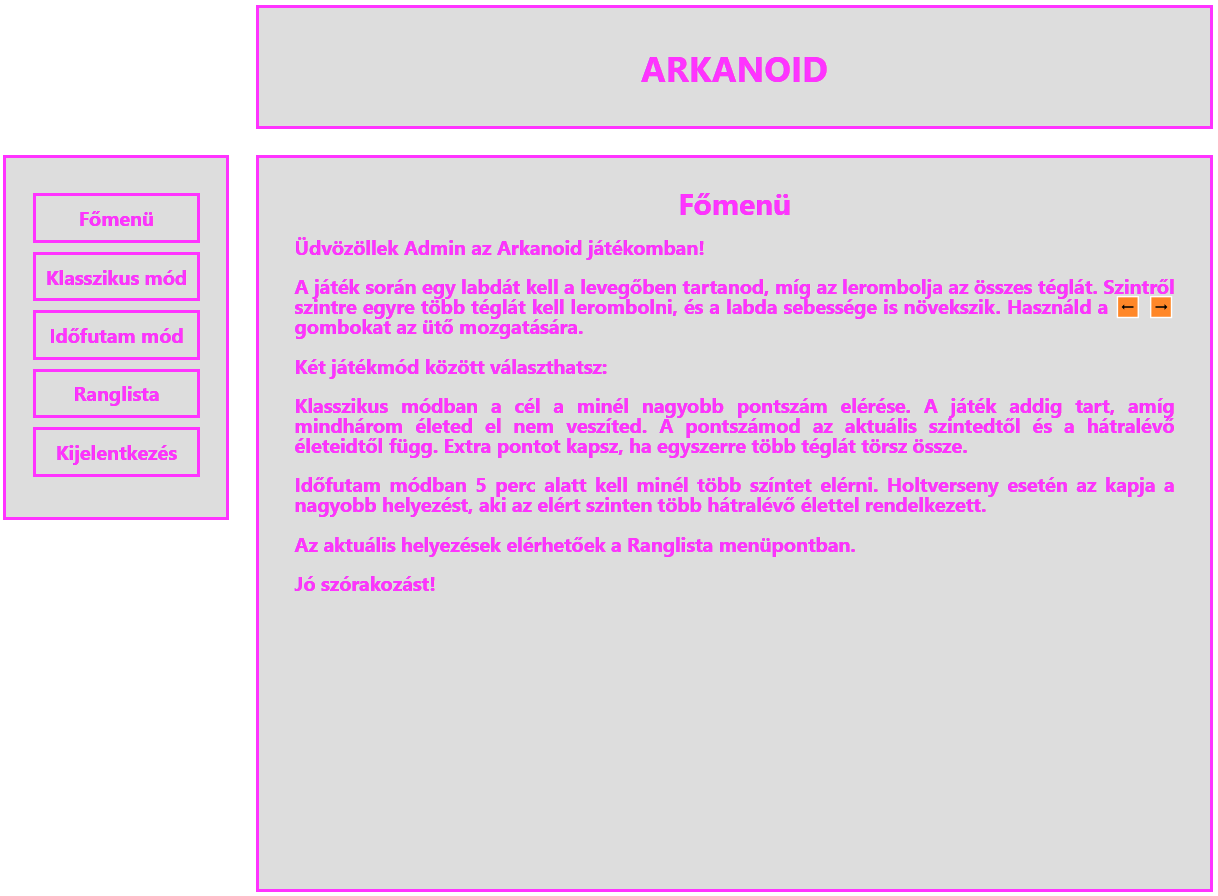
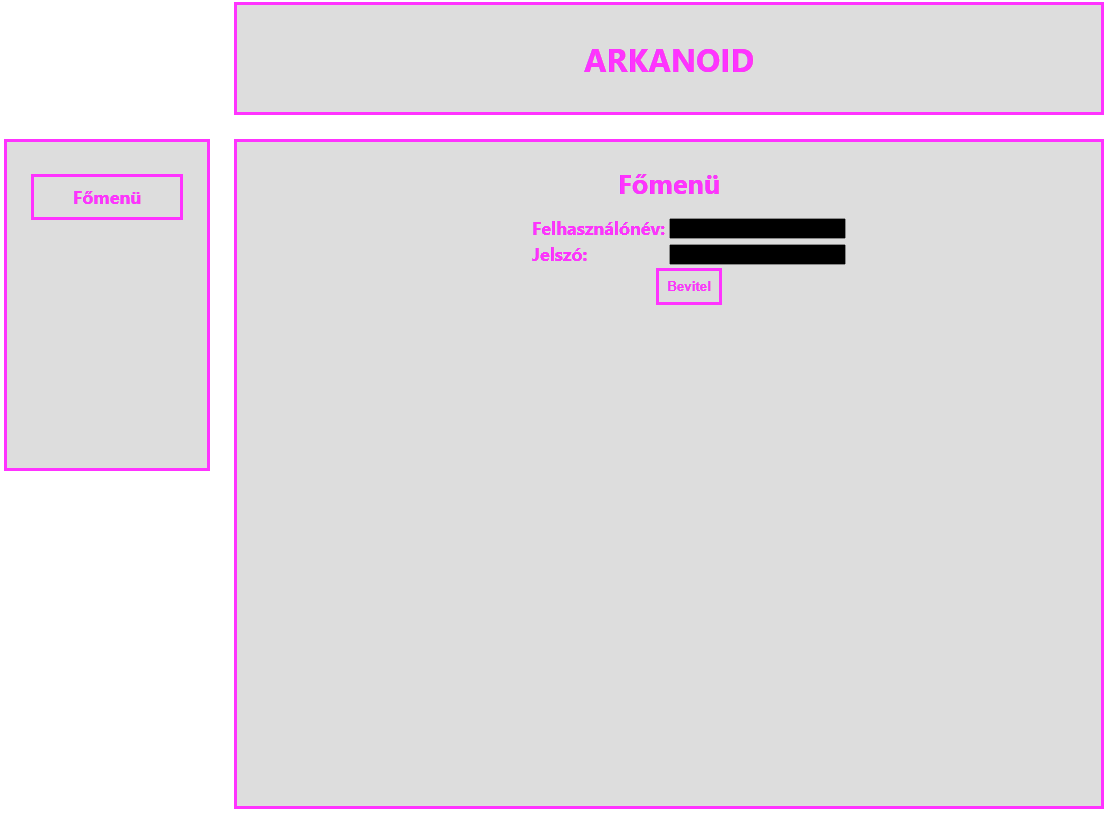
Indítsuk el az XAMPP vezérpultot az „xampp-control.exe” futtatásával, majd indítsuk el az Apache és MySQL szervereket.



Utolsó lépésként nyissuk meg tetszőleges webböngészőnket, és navigáljunk a „localhost/php” linkre, ahol megtaláljuk a játék mappáját. Erre kattintva azonnal elindul a program.



## A program használatának a részletes leírása



Alapszabályok:

• Amit leprogramoztál, azt a dokumentációban írd is le, ne legyenek eltitkolt funkciók.

• Minden pontosan, „szájbarágósan” legyen leírva. A dokumentáció alapján a teljesen kezdő, vagy laikus felhasználóknak is használniuk kell tudni a programot.

• A stílus legyen pontos és közérthető, vedd figyelembe, hogy a felhasználói dokumentáció nem szakembereknek készül.

• Ugyanakkor kerüld a laza stílust: rövidítések, smilie-k, szleng kizárva.

• Alkalmazz ábrákat, screenshot-okat , de a ne legyen túlzott a képek aránya a szöveghez képest. Kb. 2-3 oldalanként egy ábra megfelelő.

Ajánlott terjedelem: 10-15 oldal, ábrákkal együtt.

Ajánlott terjedelem: 1-2 oldal

# Összegzés

Összességében elégedett vagyok a teljesítményemmel. A kiszögezett pontok nagyrészét teljesíteni tudtam, sőt, túlszárnyaltam azokat az elvárásokat, amiket magammal szemben szabtam ki.

Az oldalak dinamikus felépítése, és a bejelentkezési rendszer eredetileg nem voltak tervben, fejlesztés közben pattantak ki a fejemből. Ezek megoldása sok időbe telt, de úgy vélem közel sem lenne annyira felhasználóbarát a végső program, ha nem valósítottam volna meg őket.

Szintén nehézkes volt az oldal elrendezésének megformálása CSS nyelv használatával, hiszen nem vagyok annyira jártas a nyelvben. Ennek ellenére el tudtam készíteni egy minimalista, de stílusos kinézetet, ami illik a játék hangulatához.

Sajnos voltak funkciók, amiket nem tudtam implementálni, legfőképp azért, mert kifutottam az időből.

A legnagyobb funkció, amit kihagytam a power-up rendszer. Szerettem volna, ha a felhasználó játszás közben különböző erőkre tehetne szert, amik megváltoztatják a játékmenetet. Ilyen lett volna például 2 labda 1 helyett, szélesebb vagy gyorsabb ütő, esetleg képes lenni gombnyomásra lövedéket kibocsátani, ami lerombol téglákat.

Kihagytam továbbá a grafikai- és hangelemeket, amelyeket én magam hoztam volna létre szerkesztő programok segítségével. Mivel a játék grafikája minimalista, nem lett volna értelme textúrákat vagy képeket elhelyezni benne. Szerettem volna hanganyagot szerkeszteni, például ütés hangokat vagy szintlépés zenét, de ezt szintén a határidő miatt nem tudtam implementálni.

A jövőben szeretnék finomítani a programon, hiszen ezek a funkciók sokkal élvezhetőbbé tennék a játékélményt. Mindezek nélkül is meg vagyok elégedve önmagammal, és remélem hogy a játékosok is épp annyira élvezik felhasználni a programom, mint amennyire én élveztem elkészíteni.

# Irodalomjegyzék, hivatkozásjegyzék