Miskolci Egyetem  
Gépészmérnöki és Informatikai Kar  
Általános Informatikai Intézeti Tanszék

A képen szimbólum, embléma, Betűtípus, Grafika látható

Automatikusan generált leírás

**Szakdolgozat**

**HTML5 Canvas alapú Space invaders játék fejlesztése webes környezetben**

**környezetben**

**Készítette:** Nagy Bence  
**Neptunkód:** WH8L7E  
**Szak:** Mérnökinformatikus  
Informatikai rendszermérnök szakirány

**Témavezető:** Dr. Mileff Péter egyetemi docens

Miskolci Egyetem, 2024

Tartalom

[Bevezetés: 3](#_Toc177823947)

[Témaválasztás és jelentőség: 3](#_Toc177823948)

[Célkitűzések: 3](#_Toc177823949)

[Motiváció: 4](#_Toc177823950)

[A játékfejlesztés története: 4](#_Toc177823951)

[A játékpiac helyzete 6](#_Toc177823952)

[Történelmi áttekintés: 6](#_Toc177823953)

[A piac hajtóereje: 7](#_Toc177823954)

[A piac eloszlása: 9](#_Toc177823955)

[A játékfejlesztésnél használt alapfogalmak: 10](#_Toc177823956)

[Játékfejlesztési motorok: 12](#_Toc177823957)

[HTML5 játékfejlesztési technológiák: 15](#_Toc177823958)

[Ismertebb keretrendszerek és könyvtárak: 16](#_Toc177823959)

[HTML grafika: CANVAS, SVG: 17](#_Toc177823960)

[Modern webes adattárolási megoldások: 19](#_Toc177823961)

[Digitális Tartalmak Felhasználása: Licencjogok és Fair Use 22](#_Toc177823962)

[Általános áttekintés: 22](#_Toc177823963)

[Creative Commons Licenc típusok: 23](#_Toc177823964)

[Felhasznált források: 24](#_Toc177823965)

# Bevezetés:

## Témaválasztás és jelentőség:

A játékfejlesztés, mint téma rendkívül fontos a modern digitális kultúrában. A játéktervezés és -fejlesztés amellett, hogy a játékipar egyik legaktívabban fejlődő szegmense, mély elméleti és gyakorlati jelentőséggel is bír. Az elmúlt években a technológiai fejlődése, különösen az internet és a mobil eszközök terjedése, forradalmasította a játékipart is.

Az alábbiakban néhány érvet sorolok fel, amelyek alátámasztják a témaválasztásom relevanciáját:

* kreativitás: A fejlesztőknek ki kell találni új mechanikákat, történeteket, grafikai elemeket, amelyhez nem árt a produktivitás.
* készségfejlesztés és csapatmunka: A programozóknak a játék készítése közben meg kell tanulniuk kódolni, problémamegoldási képességeket kell fejleszteniük, és hatékonyan együtt kell működniük a csapattársaikkal.
* szórakoztatás: Alapvetően a játékokat az emberek szórakoztatására, idejének kellemes eltöltése végett találták ki.
* oktatás: A játékok hatékony eszközei lehetnek a tanulásnak. Például lehet velük tanítani matematikát, idegen nyelvet és akár történelmet is.
* gazdasági lehetőségek: A játékfejlesztés egyre nagyobb piacra tesz szert. Egy sikeres játékkal akár milliókat, még sikeresebbel milliárdokat lehet keresni.

Az elmúlt években az internetes technológiák fejlődése lehetővé tette olyan magas színvonalú játékok létrehozását, amelyeket korábban csak asztali vagy konzolos platformokon lehetett elérni. Az HTML5 alapú játékok könnyen elérhetők és játszhatók bármely modern böngészőben, ami hatalmas piaci lehetőségeket rejthet magában.

## Célkitűzések:

A játékfejlesztéssel kapcsolatos céljaim közé tartozik a személyes fejlődés, a kreativitás kibontakozása és a játékosok szórakoztatása. Saját játék létrehozásával új készségeket és ismereteket szeretnék elsajátítani a tervezéstől kezdve egészen a programozásig, és felfedezni az alkotás örömét és kihívásait.

* **Átfogó megértés:** A projekt fő célja a HTML5 játékfejlesztési folyamatának és technológiájának mélyebb megértése. Ennek érdekében kiterjedt kutatást folytatok a HTML, CSS és JavaScript elemek játékfejlesztésben való felhasználásáról. Ezenkívül részletesen kutattam azokat a konkrét könyvtárakat és keretrendszereket, amelyek lehetővé teszik a játékfejlesztést, hogy átfogó képet kaphassanak ezek működéséről és használatáról.
* **Kreativitás kibontakozása:** A másik lényeges cél a kreativitásom fejlesztése volt a játékfejlesztés során. Célom, hogy kihasználjam a HTML5 adta lehetőségeket, hogy egyedi és innovatív játékélményt alkossak. Ez magában foglalja az animáció, a grafika, a kép és a hang kreatív felhasználását olyan játék létrehozásához, amelyek lekötik a játékosokat, és egyedivé teszik a játékélményt.
* **A felhasználói élmény javítása:** A játékosok elégedettsége és szórakoztatása is lényeges szempont a projekt során. Ezért kiemelt figyelmet fordítok a felhasználói élmény folyamatos javítására, fejlesztésére. Célom, hogy a játék zökkenőmentesen fusson és izgalmas játékélményt nyújtson, miközben intuitív és könnyen használható felületet biztosítok a felhasználóknak.

## Motiváció:

Személyes motivációm a játékok létrehozására több tényezőből fakad. Először is az önfejlesztés vágya motivál. Játékok készítése közben számos új technológiát és programozási nyelvet tanulok, miközben olyan összetett problémákat oldok meg, amelyek javítják technikai és elemzési készségeimet. Sőt, a kreatív megnyilvánulás lehetősége is erős indíttatást jelent számomra. Saját ötleteim és elképzeléseim életre keltése izgalmas és hasznos tevékenység, amely során kamatoztathatom alkotói és tervezői képességeimet. Emellett a játékfejlesztés célja, hogy szórakoztató és élvezetes élményeket nyújtson a játékosoknak. Az a tudat, hogy mások is élvezhetik az általam létrehozott játékot, erőteljes motiváció a fejlesztési folyamatban. Emellett inspiráló az iparág folyamatos fejlődése és a karrierlehetőségek sokszínűsége, mert a játékfejlesztés területén megszerzett tapasztalatok és készségek értékesek lehetnek a munkaerőpiacon számos területen a játékfejlesztésen kívül is. A kihívások is mérvadó ösztönzést jelentenek számomra. A komplex problémák megoldása, a technológiai akadályok leküzdése folyamatosan inspirál és serkent a tanulásra, fejlődésre. Továbbá, a kreatív és innovatív megoldások kidolgozása izgalmas és hálás tevékenység, amely lehetőséget ad arra, hogy a feladat során még tovább fejlesszem képességeimet.

# A játékfejlesztés története:

Az emberek évezredek óta próbálnak szórakoztató és unaloműző tevékenységeket találni. Az egyik legelterjedtebb és legrégebbi módszer erre a játék. A múltban a játékkészítés egyszerű kézzel készített játékokkal kezdődött, és az évszázadok során a technológiai és kulturális változásoknak köszönhetően fejlődött.

A játékfejlesztés történetének korai szakaszában az emberek egyszerű játékokat hoztak létre, például fafaragásokat, homok- és kavicsos játékokat vagy társasjátékokat. Ezek a játékok gyakran kézzel, egyszerű anyagokból készültek. (például: fák, kövek, agyag, termések, növények, szarvak és agyarak) Ezek nagyon népszerűek, és gyakran használtak voltak társadalmi, vallási vagy oktatási célokra. Továbbá, jellemző volt még, hogy a család tagjai, ismerősök vagy éppen saját kezűleg készítették a játékokat.

A játékfejlesztés igazi áttörése az ipari forradalommal és a technológia fejlődésével kezdődött. A 20. század elején megjelentek az első mechanikus játékok, majd később az elektromos játékok és a flippergépek. Ezzel együtt népszerűségük is kezdett egyre elterjedtebbé válni a világon. A játékok anyagát tekintve a természetben is könnyen megtalálható anyagokat felváltotta a műanyag, a fémek és egyéb szintetikus anyagok.

Az első digitális számítógépek megjelenésével, új lehetőségek nyíltak meg a játékfejlesztés számára. Az 1958-ban kifejlesztett „Tennis for Two” például az egyik első digitális számítógépes játék volt, amely egy egyszerű teniszszimulációt nyújtott a játékosoknak.

Az 1970-es években megjelentek az első videojátékok, mint például a „Pong”, amelyeket már a szórakoztató elektronikai iparágban dolgozó vállalatok fejlesztettek ki. A videojátékok az 1980-as években és az 1990-es évek elején robbanásszerűen terjedtek, az első otthoni konzolok és személyi számítógépek lehetővé tették a játékosok számára, hogy otthon is élvezhessék a játékokat.

Az internet térnyerésével és a mobiltechnológia fejlődésével az elmúlt években újabb nagy változások történtek a játékfejlesztés területén. Nem kellett sokat várni a játékforgalmazó platformok megjelenésére sem. Ezeken keresztül elérhetők, letölthetők (ha szükséges) és játszhatók a videójátékok, beleértve az asztali számítógépeket, a konzolokat, a mobiltelefonokat és a böngészőket.

Említésként néhány népszerű játékértékesítő platform:

* Asztali pc játékok: Origin, Epic Games Store, Ubisoft, Electronic Arts App, Itch, Steam (asztali számítógép mellett saját fejlesztésű kézikonzolra is készít játékot)
* Konzol: Xbox, Playstation, Nintendo (kézikonzol is), Asus (csak kézikonzol)
* Mobiltelefon: App Store (MacOS), Google Play Store (Android), AppGalery(Huawei), Galaxy Store (Samsung)

Ezeken is túlhaladva megjelentek az előfizetős platformok, amiknek főbb lényege: hogy legalább 1 vagy több hónapra fizet a felhasználó a felület használatáért. Ennek előnye: jelentősen kevesebb összeget kell kiadnia a vásárlónak mintha megvenné a játékokat. A vállalat részéről pedig: havonta, tehát folyamatosan szerez bevételt. Ilyen platform az: Microsoft Xbox Game Pass szolgáltatása, ami asztali számítógépre és konzolra is elérhető. Továbbá a Playstation Plus szolgáltatása is elérhető a felhasználók számára jelenleg csak és kizárólag konzolra. Ezen felül a Netflix is elkezdte játékok bevezetését a szolgáltatásaik közé.

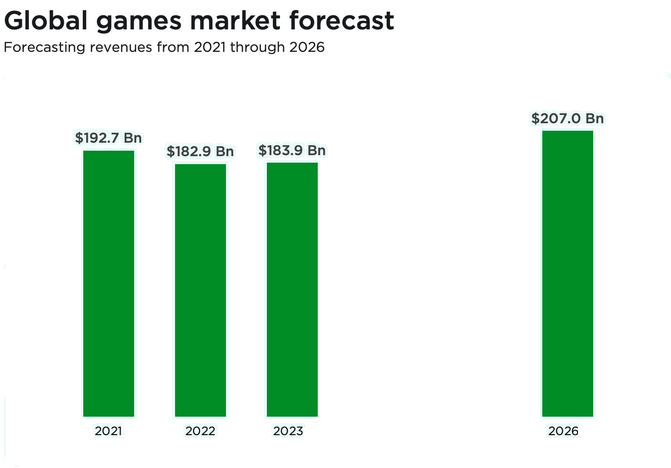
Fontosnak tartom még megemlíteni, hogy az utóbbi években egyre nagyobb hangsúlyt kapott a VR és AR technológia térhódítása is. A virtuális valóság (VR) és a kiterjesztett valóság (AR) technológiák jelentős hatást gyakoroltak a játékfejlesztésre, új dimenziókat nyitva a játékélmények terén. A VR headsetek, mint az Oculus Rift, a HTC Vive, és a PlayStation VR, lehetővé teszik a játékosok számára, hogy teljesen belemerüljenek a digitális világokba, ahol szinte kézzelfoghatóan élhetik át a játékokat. Ezek a technológiák valós idejű, 360 fokos látványt és interaktív élményeket biztosítanak, amelyek messze túlmutatnak a hagyományos képernyős játékok kínálta lehetőségeken.

Így a játékfejlesztés hosszan tartó és változó történetén keresztül az emberiség mindig is arra törekedett, hogy izgalmas és figyelemfelkeltő játékokat hozzanak létre, amelyek segítenek az unalom elűzésében, a szórakozásban és a kikapcsolódásban. A játékfejlesztés folyamatosan fejlődik és változik, és a jövőben is számos lehetőséget fog kínálni mind a játékosok, mind a fejlesztők számára.

# A játékpiac helyzete

## Történelmi áttekintés:

A videojátékok piaca az egyik legjobban növekvő szektor a szórakoztatóipar területén. Az elmúlt évtizedek alatt a videojátékok számottevő átalakuláson mentek keresztül, mind technológiai, mind gazdasági szempontból nézve. Ez a terület ma már nemcsak a gyerekek és a tinédzserek szórakozását szolgálja, hanem széles körű csoportokat vonz, beleértve a felnőtteket és az idősebb generációt is. Az egyszerű, kétdimenziós játékoktól kezdve a mai napig, amikor már fotórealisztikus grafikákat és összetett játékmechanikákat kínálnak, a videojátékok rendkívül sokat fejlődtek.

A piacnak az értéke 2023-ben meghaladta a 183 milliárd dollárt, és az előrejelzések szerint ez az összeg tovább fog növekedni az elkövetkező években. 2026-ra 207 milliárd dollárra számítanak. Bizonyos bevétel-előrejelzések alapján pedig 583 milliárd dollárt becsülnek 2030-ra.

Az elmúlt tíz évben a videojáték-ipar jelentős növekedésen ment keresztül. Például 2013-ban a globális videojáték-piac értéke körülbelül 70 milliárd dollár volt. Ez azt jelenti, hogy tíz év alatt több mint megduplázódott a piac mérete​, köszönhetően a technológiai fejlődésnek, az új üzleti modelleknek és a szélesebb felhasználói bázisnak. Ha pedig visszatekintünk húsz évvel megelőzően, 2003-ban a piac mérete csupán 20 milliárd dollár körül mozgott. Ez a növekedés mutatja, hogy mennyivel nagyobb pénz van most a videojáték-iparban, mint korábban.

## A piac hajtóereje:

**Mobil játékok térnyerése:** A mobil játékok az elmúlt években a videojáték-piac egyik legjelentősebb hajtóerejévé váltak. Az okostelefonok és táblagépek elterjedése lehetővé tette, hogy a játékok széles közönséghez jussanak el. A mobil játékok könnyen hozzáférhetők, gyakran ingyenesen letölthetők, és egyszerűen integrálhatók a mindennapi életbe, így egyre több ember számára jelentenek szórakozási lehetőséget. Az olyan játékok, említésként mint a Candy Crush Saga, a Clash of Clans és a Pokémon GO hatalmas sikert arattak, és milliárd dolláros bevételeket generáltak. A mobil játékok bevételi modelljei, amelyek gyakran mikro tranzakciókra és reklámokra épülnek, tovább növelik a fejlesztők és kiadók jövedelmezőségét. A mobil játékok gyakran úgy vannak tervezve, hogy rövid ideig tartó, de gyakori játmenetet kínáljanak, ami passzol a felhasználók napi rutinjához.

Tehát a mobil játékok közkedveltsége mögött álló tényezők:

* Hozzáférhetőség
* Ingyenes letöltések
* Rövid játékidő

**E-sportok robbanásszerű növekedése:** Az e-sport, vagyis a kompetitív videojátékok népszerűsége drámaian megnőtt az elmúlt években. Az e-sportversenyek hatalmas közönséget vonzanak, a legnagyobb események, több millió online és élő nézőt vonzanak. Az e-sport piac bevételei elsősorban szponzorálásból, reklámozásból, jegyértékesítésből származnak. Fontos vonzerőt jelent a játékosok és csapatok hírneve, valamint a játékok pénzdíja is. Az e-sport ökoszisztéma kialakulása és fejlődése új bevételi forrásokat és marketing lehetőségeket teremtett a játékfejlesztők számára is.

Felkapottságának szempontjai:

* Professzionális versenyzés
* Közösség
* Élő közvetítések
* Jókora reklámozási lehetőség

**Felhőalapú játékok:** A felhőalapú játéktechnológia alkalmazásával futtathatunk játékokat bármilyen eszközön anélkül, hogy nagy teljesítményű hardverre lenne szüksége. A Microsoft xCloud és az NVIDIA GeForce Now olyan szolgáltatásokat nyújtanak, amelyekkel a játékokat futtatni tudjuk távoli szervereken, és a tartalmat megosztani az interneten keresztül a játékosok eszközeire. Ez a technológia különösen vonzó azok számára, akik nem rendelkeznek nagy teljesítményű játékkonzollal vagy PC-vel, de továbbra is szeretnék élvezni a legújabb, nagyobb hardverigénnyel rendelkező játékokat. A felhőalapú játékok előnyei közé tartozik a hozzáférhetőség, a költséghatékonyság és a platformfüggetlenség, amely lehetővé teszi a játékosok számára, hogy bárhol és bármikor játszhassanak.

Keresettségének okai:

* Könnyű hozzáférhetőség
* Költséghatékonyság
* Alacsony hardverigény

**Előfizetéses modellek:** Az előfizetéses modellek egyre nagyobb előretörést érnek el a videojáték-iparban. Az olyan szolgáltatások, mint a Microsoft Xbox Game Pass, a PlayStation Plus, a Ubisoft+ és az EA Play, a játékok széles könyvtárához biztosítanak hozzáférést rögzített havi díj ellenében. Ezen megoldások a játékosok számára költséghatékony hozzáférést biztosít számos játékhoz, míg a fejlesztők és kiadók számára folyamatos bevételi forrást jelentenek és hosszú távú lojalitást hoznak a játékosok részéről . Az előfizetéses modellek lehetővé teszik a játékosok számára, hogy új játékokat próbáljanak ki anélkül, hogy teljes árat kellene fizetniük, és ezáltal növelik a játékok elérésének és kipróbálásának esélyét.

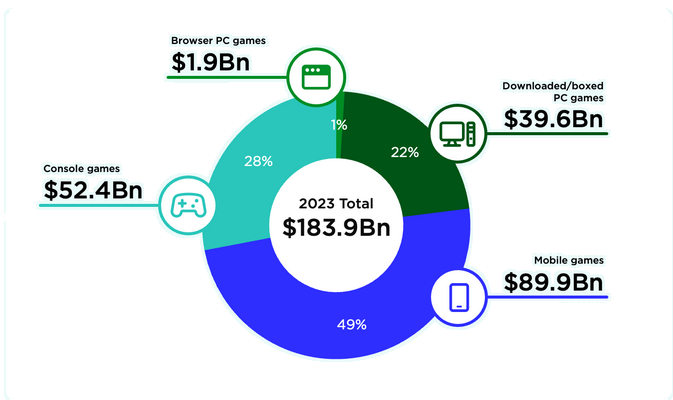
Popularitásának összetevői:

* Folyamatos frissítések, gyakori megújulás
* Költséghatékonyság
* Hűség a szolgáltatás felé
* Bevételbiztonság (fix havi bevétel)

Összefoglalva a piac növekedését számtalan tényező hajtja, köztük a mobil játékok térnyerése, az e-sportok népszerűsége, a felhőalapú játékok elterjedése és az előfizetéses modellek előretörése. Ezek a tényezők nemcsak új lehetőségeket teremtenek a fejlesztők és kiadók számára, hanem a játékosok számára is gazdagabb és változatosabb élményeket kínálnak, ezáltal tovább erősítve a videojáték-ipar pozícióját a szórakoztatóipari szektorban.

## A piac eloszlása:

A játékipar jelentős mértékben eltér az eszközök típusa alapján, amelyeken a játékokat játszák. Az alábbiakban bemutatom a legfontosabb eszközkategóriákat és azok piaci részesedését:

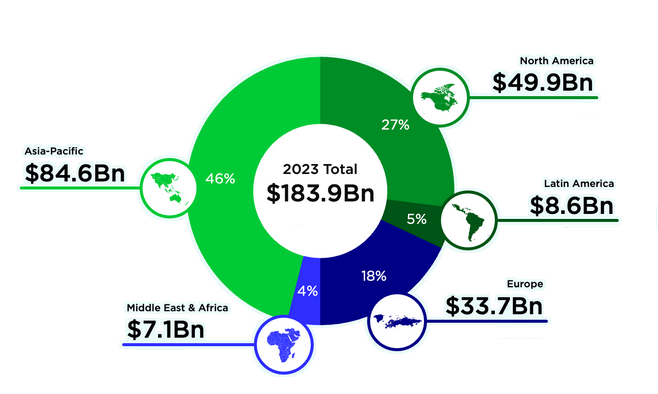


**Mobil Játékok**: A legnagyobb piaci egységet képezik. 2023-ban a mobil játékok a teljes piac közel felét: 49%-át tették ki. Ennek a szegmensnek a növekedését a széleskörű okostelefon-használat, a könnyen hozzáférhető alkalmazásboltok, és az ingyenesen elérhető (freemium) modellek népszerűsége hajtja. A piaci részesedése: 89,9 milliárd dollárt ért el.

**Konzolos Játékok**: Ez a kategória foglalja el a második helyezést a területen, különösen a prémium konzolok (PlayStation, Xbox, Nintendo Switch) esetében. Ezek a játékok az összes videojáték-bevétel körülbelül 28%-át teszik ki. A konzolos játékok előnye, hogy magas színvonalú grafikai élményt nyújtanak, és gyakran kínálnak exkluzív (csak a platformon elérhető) címeket. A konzolok részesedése: 52,4 milliárd dollárt ért el 2023-ban.

**PC Játékok**: A piaci részesedése a PC játékoknak az elmúlt években mindig 20-25% körül mozgott. A tavalyi kimutatás alapján a 23%-ot érte el, amibe beletartozik a böngészőből futó játékokat kezdve a letölthető pc játékokig szinte minden. Az asztali számítógépeken és laptopokon játszott játékok különösen népszerűek a keményvonalas játékosok körében, akik a nagy teljesítményű hardvereket és a széleskörű testre szabási lehetőségeket értékelik a leginkább. Az online elérhető értékesítési helyek, jelentősen hozzájárultak ehhez a területrészhez. A PC játékok: 41,5 milliárd dollárt értek el a mért adatok alapján.

A videojáték-piac eloszlása földrajzi szempontból is megfigyelésre érdemes.



* **Megfigyelhető, hogy az Ázsia, különösen Kína és Japán** a legnagyobb és legdinamikusabban növekvő videojáték-piacot képviseli. Kína önmagában a globális bevételek több mint egyharmadát generálja, főként a mobil játékok révén. Japán pedig a konzolos és mobil játékok egyik legnagyobb piaca. Így a bevételek 46%-át adják az iparnak.
* A következő nagy bevételt jelentő kontinens, nem más mint Észak-Amerika 27%-al. Ez a régió a konzolos és PC-s játékok terén különösen erős.
* Európa szintén számottevő piaci részesedéssel bír (18%), különösen az Egyesült Királyság, Németország, és Franciaország révén. Az európai piac sokszínűsége miatt a konzolos, PC-s, és mobil játékok egyaránt népszerűek.
* Bár latin-amerikai, a közel-keleti és afrikai régió kisebb nyereséget jelentenek a játékipar számára, azonban a növekedési potenciáljuk hatalmas. A mobil játékok különösen népszerűek, mivel az okostelefonok terjednek leginkább a térségben megtalálható országokban.

# A játékfejlesztésnél használt alapfogalmak:

A videojátékok fejlesztése egy összetett folyamat, amely jó néhány különböző eszközt és technológiát igényel. A fejlesztés során gyakran találkozhatunk olyan kifejezésekkel, mint a játékfejlesztési motor, a fejlesztési keretrendszer és a játékfejlesztési technológia. Igaz, ezek a fogalmak gyakran átfedésben állnak egymással, mindegyiknek megvan a maga sajátos jelentése és szerepe. Ebben a részletes kifejtésben bemutatom és megmagyarázom ezen fogalmak közötti különbségeket és hasonlóságokat, amik végül közreműködnek a játékfejlesztési folyamat sikeréhez.

**Játékfejlesztési technológia (Game Development Technology):**

Ezt egy általános kifejezésként használjuk, amely magába foglalja az összes olyan technológiai eszközt, szoftvert és infrastruktúrát, amelyet a videojátékok fejlesztése során használnak. Ez tartalmazhat játékfejlesztési motorokat, keretrendszereket, eszközöket és más segédprogramokat is. A játékfejlesztési technológia széles körű eszközkészletet kínál, amely mindenféle szoftvert és hardvert magába foglalnak, amely a játékfejlesztéshez szükséges. Az integráció és kompatibilitás fontos szempont, hogy a különböző technológiai elemek jól működjenek együtt. A fejlesztési támogatás is lényeges, hiszen támogatást nyújt a játék teljes életciklusában, a koncepciótól a megjelenésig és azon túl is. Ide tartoznak például a verziókövető rendszerek, mint például a Git, az integrált fejlesztői környezetek, mint a Visual Studio, a 3D modellező szoftverek, mint a Blender valamint a hangtervező szoftverek, mint az Audacity és az FMOD.

**Fejlesztési keretrendszer (Development Framework):**

Egy általánosabb szoftvereszközkészlet, amely segít a fejlesztőknek különféle alkalmazások, beleértve nem csak játékok, létrehozásában. Ez a keretrendszer egy előre megírt kódot biztosít, amelyet a fejlesztők újra felhasználhatnak, így felgyorsítva a fejlesztési folyamatot. A moduláris felépítése lehetővé teszi a különböző komponensek könnyű bevezetését és cseréjét, míg a kód újra felhasználhatósága előre megírt kódkomponenseket kínál, amelyeket több különböző projektben is használhatnak. A fejlesztési keretrendszer számos standard funkciót és könyvtárat biztosít, amelyek gyakori feladatokat oldanak meg, és átfogó dokumentációval rendelkezik, hogy segítse a szoftver készítőit a használatában. Példák erre a kategóriára a Django (webfejlesztéshez), az Angular (webes alkalmazásokhoz) és a .NET Framework (általános alkalmazásfejlesztéshez).

**Játékfejlesztési motor (Game Engine):**

A játékfejlesztési motor egy szoftver, amely alapvető funkciókat és eszközöket biztosít a videojátékok fejlesztéséhez. Ezek az eszközök magukban foglalják a grafikai megjelenítést, a fizikai szimulációt, az animációkat, a hangkezelést, a mesterséges intelligenciát, valamint a hálózati kommunikációt. A grafikai motor felelős a 2D és 3D grafikai elemek rendereléséért, míg a fizikai motor szimulálja a fizika törvényeit, mint például a gravitáció, az ütközések lebonyolítása és egyéb mozgások megvalósulása. A hangmotor kezeli a hangok lejátszását, effekteket és a felhasznát zenéket. Az animációs rendszer lehetővé teszi a karakterek és objektumok mozgását és animációit. A játékfejlesztési motoroknál jellemző még, hogy különböző eszközöket is kínálnak a mesterséges intelligencia funkciók megvalósításához, amelyeket a játék karakterei és ellenfelei számára használhatunk fel. Ezen kívül a scripting rendszer lehetővé teszi a fejlesztők számára, hogy kódot írjanak a játékmenet logikájának és interakcióinak kezelésére. Példák erre a kategóriára a Unity, az Unreal Engine, a CryEngine, a Godot és a GameMaker Studio.

Összefoglalva, a játékfejlesztési motorok, fejlesztési keretrendszerek és játékfejlesztési technológiák használata fontos szerepet játszik a játékfejlesztés folyamatában, és különböző eszközöket és szolgáltatásokat nyújt a fejlesztők számára, hogy könnyebben és hatékonyabban tudjanak dolgozni.

# Játékfejlesztési motorok:

A fejlesztésben rengeteg technológia áll rendelkezésre, amelyek által a fejlesztők különböző platformokra és célközönséghez szóló játékokat hozhatnak létre. Ezek közül bemutatom, jellemzem a legnépszerűbb játékfejlesztési motorokat:

**Unity:**

* Platformtámogatás: Az egyik legszélesebb körben használt játékfejlesztő motor, amely támogatja a PC-t, játékkonzolokat, mobileszközöket és VR/AR platformokat. Segít több platform egyidejű fejlesztésében is. A Unity több mint 25 platformot támogat.
* Könnyen megtanulható: Egyszerűen elsajátítható, különösen a kezdők számára. A felhasználói felület intuitív, és rengeteg oktatási anyag áll rendelkezésre. Tanulhatóságának köszönhetően rövid időn belül hatékony játékokat készíthetünk vele.
* Nyelv: A fejlesztők programkódokat írhatnak a C# használatával, amely egy viszonylag könnyen tanulható programozási nyelv.
* Asset Store: A Unity Asset Store lehetővé teszi a fejlesztők számára, hogy előre elkészített 3D modelleket, textúrákat, animációkat és egyéb eszközöket vásároljanak, ami jelentősen felgyorsíthatja a fejlesztést.
* Alkalmazási területek: Az Unity nem csak játékokhoz, hanem interaktív 3D alkalmazásokhoz, szimulációkhoz és VR/AR projektekhez is ideális.
* Népszerű játékok, amiket Unityben készítettek: Hollow Knight, Pokemon GO, Angry Birds 2.

**Unreal Engine:**

* Grafikai minőség: Az Unreal Engine kiváló grafikai minőségéről ismert. Fejlett renderelési képességei miatt gyakran választják az AAA játékok fejlesztésében.
* Blueprints: Egyedülálló Blueprint rendszert kínál, amely lehetővé teszi a fejlesztők számára, hogy programozás nélkül, intuitív módon alakítsanak ki játékmenetet és logikát.
* **Erőteljes fizikai motor:** A beépített fizikai motorja, az Unreal Physics, kiváló valósághű fizikát és interakciókat biztosít.
* Nyelv: Az elsődleges programozási nyelve a C++, amely nagyobb teljesítményt és rugalmasságot kínál, de összetettebb is, mint a Unity C# nyelve.
* **Nyílt forráskód:** forráskódja elérhető, ami nagyfokú testreszabhatóságot biztosít a fejlesztők számára.
* Eszközök és integráció: Rengeteg eszközt kínál, például a Niagara részecskerendszert, amely lehetővé teszi részletes dinamikus hatások létrehozását.
* VR/AR: Az Unreal Engine szintén erősen támogatja a VR- és AR-fejlesztést, tömérdek integrációs és optimalizálási lehetőséggel.
* **Fotorealisztikus vizualizáció:** A fotorealisztikus vizualizációs képességeinek köszönhetően különösen alkalmasak a játékok, valamint a filmek és más médiaiparágak számára is.
* Unreal Engine használatával készült népszerű játékok: Gears of War, Rocket Leauge, Mortal Kombat.

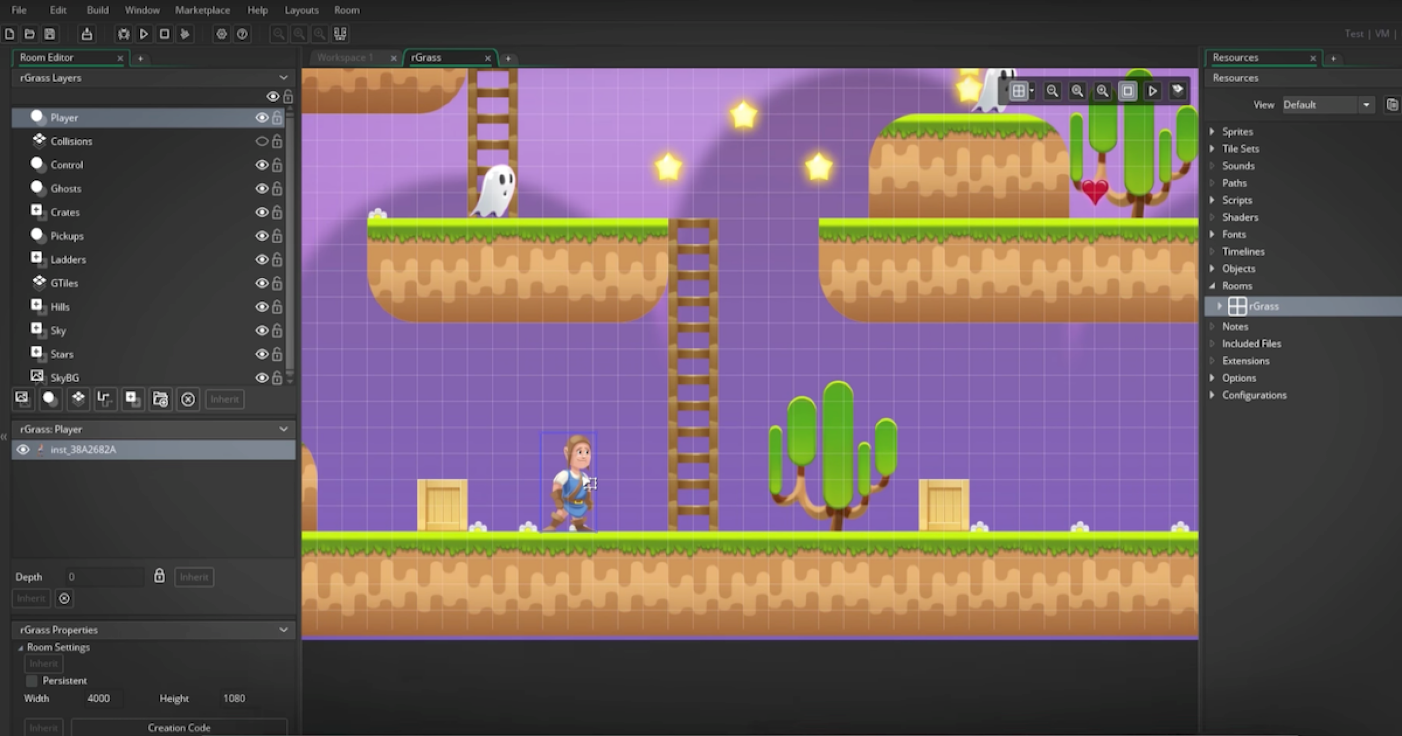


**Godot Engine:**

* Nyílt forráskódú és ingyenes: A Godot Engine egy teljesen ingyenes, nyílt forráskódú szoftver, amely lehetővé teszi a fejlesztők számára, hogy szabadon módosítsák és saját igényeikhez igazítsák az eszközt. A nyílt forráskód miatt a felhasználói közösség folyamatosan fejleszti és karbantartja.
* Támogatja a 2D és 3D játékkészítést: Kiválóan alkalmas 2D és 3D játékok készítésére. Különösen erős 2D-támogatással rendelkezik, így ideális platformerek, kirakós játékok és egyéb 2D játékok készítéséhez.
* A GDScript nyelvet használ: A saját GDScript programozási nyelvét használja, amely hasonló a Pythonhoz. A nyelv könnyen megtanulható és használható, és támogatja az objektumorientált, imperatív és funkcionális programozási paradigma alkalmazását.
* Többplatformos támogatás: Lehetővé teszi játékok létrehozását és exportálását több platformra, beleértve a Windowst, a macOS-t, a Linuxot, az Androidot, az iOS-t és a HTML5-öt.
* A motor felhasználásával készült játékok: Heartbeat, Deponia.

**GameMaker Studio:**

* Célközönség: A GameMaker Studio elsősorban kezdő és középhaladó szintű játékfejlesztőknek készült, széleskörű programozási ismeretek nélkül.
* Az eszköz használata könnyen megtanulható, így a felhasználók viszonylag gyorsan hozhatnak létre játékokat anélkül, hogy nagy programozási szakértelemre lenne szükségük.
* Drag-and-drop interface-el rendelkezik. Az alkalmazás drag-and-drop felületet kínál, amely lehetővé teszi a felhasználók számára, hogy viszonylag egyszerűen alakítsanak ki játékmenetet és logikát mély kódolási ismeretek nélkül.
* Platformfüggetlenség: Lehetővé teszi a játékfejlesztők számára, hogy játékaikat több platformra exportálják, beleértve a PC-t, a mobileszközöket és a konzolokat.
* Az itt készült játékok: Undertale, Hotline Miami.

****

**CryEngine:**

* Grafikai teljesítmény: A CryEngine a grafikus teljesítmény és a valósághű megjelenítés terén jeleskedik. A legújabb verziók összetett fényhatásokat, részletes textúrákat és valósághű animációkat támogatnak.
* Speciális funkciók: A CryEngine egy sor fejlett funkciót és eszközt kínál a fejlesztőknek, beleértve a valós idejű fizikai alapú szimulációt, az AI-eszközöket és a nagyobb méretű világok támogatását.
* A CryEngine általában összetettebb és nehezebben használható, mint a GameMaker Studio, és általában tapasztaltabb fejlesztőknek készült.
* C++ és C# programozási nyelveket használ
* Ideális választás lehet nagyobb költségvetésű projektekhez, ahol fontos a grafikai minőség és a méretezhetőség.
* CryEngine használatával létrejött videójátékok: Far Cry, Crysis, Prey.

# HTML5 játékfejlesztési technológiák:

A HTML5 játékfejlesztés az elmúlt években egyre népszerűbbé vált, mivel lehetővé teszi, hogy játékokat fejlesszenek webalapú platformokra, amelyek széles körben elérhetők asztali és mobil eszközökre egyaránt. A technológia, amely a játékok szerkezetét és megjelenését biztosítja, maga a HTML5. Ez beépített elemeket kínál, mint például a <canvas> elem, amely lehetővé teszi grafikai tartalom rajzolását közvetlenül a böngészőben. A HTML5 játékok alapvető programozási nyelve a JavaScript, amellyel kezelhetők az interakciók, az animációk, a játék logikája és a felhasználói inputok. A modern JavaScript motorok, mint például a Google Chrome V8 motorja, jelentős teljesítménynövekedést eredményeztek a böngésző alapú játékokban, ami hozzájárul a zökkenőmentes játékélményhez.

A HTML5 egyik legfontosabb eleme a játékfejlesztés szempontjából a Canvas API. A <canvas> elem lehetővé teszi 2D grafikák renderelését (grafikus megjelenítését) JavaScript segítségével, ami az alapját képezi a rajzolási műveleteknek, például a sprite-ok megjelenítésének, az animációknak és egyéb vizuális elemeknek. Ha 3D grafikáról van szó, a WebGL (Web Graphics Library) lehetőséget nyújt arra, hogy JavaScript API segítségével 3D grafikákat jelenítsenek meg a böngészőkben, anélkül, hogy bővítményeket kellene telepíteni. A WebGL emellett lehetővé teszi a GPU kihasználását, amit a különösen látványos és teljesítményigényes játékok esetében előnyös kihasználni. Az ilyen típusú játékokban a hangok és zenék integrálása elengedhetetlen, és ebben nyújt segítséget a HTML5 Audio API. Ezen API-k révén a fejlesztők könnyedén kezelhetik a hanghatásokat, háttérzenéket és egyéb hangalapú interakciókat, amelyek a játékélmény fokozásához járulnak hozzá.

Számos keretrendszer és könyvtár áll rendelkezésre, amelyek megkönnyíthetik és felgyorsíthatják a fejlesztési folyamatot, különösen azok számára, akiknek fontos a hatékonyság és a rugalmasság. Ezek a keretrendszerek és könyvtárak rengeteg előre elkészített funkciót kínálnak, amelyekkel a fejlesztők gyorsabban hozhatnak létre játékokat, kevesebb kódolási munka mellett.

## Ismertebb keretrendszerek és könyvtárak:

* **Phaser** az egyik legismertebb és legszélesebb körben használt HTML5 játékmotor, amely kifejezetten 2D játékok fejlesztésére lett kitalálva. A Phaser kiváló választás azoknak, akik gyorsan szeretnének látványos és jól működő játékokat létrehozni, mivel széles körű dokumentációval és egy aktív közösséggel rendelkezik, ami megkönnyíti az elakadt fejlesztők számára a segítségkérést és a problémamegoldást. A motor a beépített funkciók széles skáláját kínálja, beleértve a fizika szimulációkat, animációkat, valamint a hangok kezelését is, így a fejlesztőknek nem kell ezeket az alapvető funkciókat saját maguknak implementálniuk.
* **Three.js** egy másik népszerű JavaScript könyvtár, amely a WebGL technológiára épül. Ez a könyvtár kifejezetten 3D grafikák egyszerű létrehozására lett tervezve, így ideális választás 3D játékok fejlesztéséhez. A Three.js lehetővé teszi a fejlesztők számára, hogy bonyolult 3D-s jeleneteket hozzanak létre, mint például dinamikus fényforrásokat, árnyékokat, textúrákat és animációkat. Ez a könyvtár rendkívül rugalmas, és számos előre elkészített példát, valamint bővítményt tartalmaz, amelyek segítenek abban, hogy a fejlesztők gyorsabban és hatékonyabban valósíthassák meg a játékötleteiket.
* A **PixiJS** egy 2D renderelő könyvtár, amely rendkívül gyors rajzolási lehetőségeket kínál, így ideális választás 2D játékok és interaktív alkalmazások készítéséhez. A PixiJS egyik fő előnye a teljesítményoptimalizálás, mivel lehetővé teszi az összetett grafikus elemek gyors megjelenítését a böngészőkben. Ezt a teljesítményt a WebGL és a HTML5 Canvas kombinálásával éri el, és automatikusan kiválasztja a legmegfelelőbb technológiát a felhasználó eszköze alapján. Ez különösen hasznos lehet azoknak a fejlesztőknek, akik igényes grafikát szeretnének megvalósítani anélkül, hogy a teljesítményben kompromisszumot kellene kötniük.
* A **PlayCanvas** egy teljes értékű, szintén WebGL technológián alapuló 3D-s játékmotor, amelyet kifejezetten a böngészőben való futtatásra terveztek. A PlayCanvast az különbözteti meg egymástól, hogy van egy online szerkesztője, amely lehetővé teszi a fejlesztők számára, hogy valós időben együttműködjenek és megosszák munkájukat. Ez a motor hatékony eszközkészletet kínál, beleértve a fizikai szimulációkat, a grafikai megjelenítést és az animációkat, így ideális választás összetett 3D-s játékok készítéséhez. A PlayCanvas másik előnye, hogy könnyen integrálható más webes technológiákkal, így rugalmasságot biztosít a játékok létrehozásához és közzétételéhez különböző platformokon.

Mindent összevetve, ezek a keretrendszerek és könyvtárak különböző típusú játékfejlesztési igényeket fednek le, legyen szó egyszerű 2D-s, vagy bonyolult 3D-s játékokról. Mivel minden megoldásnak megvannak a saját egyedi előnyei, így a fejlesztőknek célszerű a projektük igényeihez igazodva választaniuk a rendelkezésre álló eszközök közül.

## HTML grafika: CANVAS, SVG:

A modern webfejlesztésben a **HTML5** lehetőséget biztosít különböző grafikai elemek megjelenítésére, két kiemelkedő technológiával: a **Canvas** és az **SVG** (Scalable Vector Graphics) segítségével. Ezek a technológiák különböző módon kezelik a grafikák megjelenítését és interakcióját, így mindkettőnek megvannak az erősségei és gyengeségei, valamint a megfelelő használati esetei.

A **Canvas** egy HTML5 elem, amely lehetőséget ad a fejlesztőknek arra, hogy dinamikus, raszteres grafikákat hozzanak létre JavaScript segítségével. Ez a technológia egy „rajzvásznat” biztosít, ahol különböző grafikai elemek – például vonalak, alakzatok, szövegek, képek és animációk – hozhatók létre. A Canvas egyik erőssége, hogy teljes mértékben képpont alapú, vagyis minden egyes objektum egy adott képpontra rajzolódik ki, hasonlóan a képszerkesztő programok működéséhez.

A technológia legfőbb előnye a nagyfokú rugalmasság és a gyorsaság, amely különösen hasznos, amikor valós idejű grafikai frissítésekre, például játékokban, animációkban vagy adatvizualizációkban van szükség. Mivel a Canvas képekkel és képpontokkal dolgozik, a grafikák egyszerűen, közvetlenül manipulálhatók JavaScript kódon keresztül, ami lehetővé teszi komplex és interaktív grafikai megoldások létrehozását. Ezen túlmenően, a Canvas támogatja az olyan további funkciókat, mint a pixelmanipuláció, amely lehetővé teszi, hogy módosítsuk az egyes pixelek színeit, átlátszóságát vagy más tulajdonságait, ami különösen hasznos képek szűréséhez vagy különleges effektusok létrehozásához. Egy másik kulcsfontosságú előnye a valós idejű animáció létrehozásának lehetősége, amely különösen a dinamikus, mozgó tartalmak megjelenítésekor fontos. Mivel a képernyő közvetlenül frissíthető, a Canvas lehetőséget biztosít arra, hogy gyors és folyékony animációkat hozzunk létre anélkül, hogy a teljes rajz újragenerálása szükséges lenne.

Azonban a képpont alapú megközelítés negatívuma, hogy ha a vásznon megrajzolt grafikát nagyítjuk vagy kicsinyítjük, az veszít a minőségéből, mivel nem méretezhető újra tökéletesen, ahogy a vektoros grafikák. Továbbá, a Canvas statikus természete miatt nem tartja nyilván az egyes rajzelemeket, vagyis ha egy elemet újra szeretnénk rajzolni vagy frissíteni, akkor a teljes vásznat újra kell rajzolni. Ez nagyobb adatmennyiségnél vagy összetett grafikai elemeknél teljesítménybeli kihívásokhoz vezethet.

A **Scalable Vector Graphics (SVG)** egy másik HTML5 technológia a grafikák megjelenítésére, de a Canvas-szal ellentétben az SVG vektoralapú. Ez azt jelenti, hogy az SVG-ben megjelenített grafikus elemek matematikai képletek, nem pedig pixelek alapján jönnek létre. Ez lehetővé teszi a grafika nagyítását vagy kicsinyítését bármilyen méretre a minőség romlása nélkül. Az SVG előre definiált elemeket, például vonalakat, köröket, négyzeteket, sőt, XML formátumban tárolt összetett formákat is képes megjeleníteni. Az SVG előnyei különösen szembetűnőek a statikus, jó minőségű grafikai elemek, például logók, diagramok, ikonok vagy más vektoros alakzatok létrehozásakor. A vektor alapú megközelítés lehetővé teszi az elemek automatikus átméretezését, hogy illeszkedjenek a böngészőablak vagy más képernyő méretéhez anélkül, hogy elveszítenék a részleteket. Az SVG ezen kívül könnyen módosítható CSS-sel és JavaScripttel, így az interaktív grafikák kialakításában is szerepet játszhat.

Canvas és SVG összehasonlítása:

A **Canvas** és az **SVG** közötti legnagyobb különbség az alapvető megközelítésben rejlik: míg a Canvas képpont alapú (raszteres), addig az SVG vektoros formátumot használ. Emiatt a Canvas jobban használható valós idejű, folyamatosan változó grafikák, például játékok, animációk vagy diagramok megjelenítéséhez, ahol a grafikai elemek gyorsan és folyamatosan frissülnek. Az SVG pedig inkább statikus vagy ritkábban frissülő grafikákhoz alkalmas, ahol a részletesség és a minőség fontosabb, például ikonok, logók vagy interaktív diagramok esetén.

* **Canvas** előnyei közé tartozik a nagyfokú rugalmasság és a valós idejű grafikai frissítések, azonban hátránya a minőség romlása nagyításkor és a képpontok korlátozott kezelhetősége.
* **SVG** fölénye, hogy a grafikai elemek bármilyen méretben részletgazdagok maradnak, és könnyen integrálhatók a HTML dokumentum szerkezetébe. Hátránya viszont, hogy nagyobb és összetettebb grafikák esetén teljesítményproblémák léphetnek fel, különösen, ha sok elemről van szó, vagy gyakori frissítést igényel.

Felhasználási esetek:

A Canvas használata akkor ideális, amikor dinamikus, nagy teljesítményű grafikai elemekre van szükség, mint például játékoknál, valós idejű animációknál vagy adatvizualizációknál. Ezzel szemben az SVG tökéletes választás, ha statikus, átméretezhető grafikákat, ikonokat, logókat vagy diagramokat szeretnénk létrehozni, amelyeknek minden kijelzőn meg kell őrizniük minőségüket.

## Modern webes adattárolási megoldások:

A témában említésre méltó még a **Local Storage, SessionStorage és az IndexedDB** szerepe amelyek, olyan esszenciális HTML5 eszközök, amik lehetővé teszik a fejlesztő számára, hogy a webalkalmazásokban helyi adattárolást valósítsanak meg a felhasználók böngészőjében. Ezek a technológiák nagy előrelépést jelentenek a felhasználói élmény javításában, különösen akkor, amikor internetkapcsolat nélküli működésre, vagy gyorsabb, kiszolgálófüggetlen adattárolásra van szükség.

**Local Storage** A Local Storage az egyik legegyszerűbb megoldás a kliensoldali adattárolásra. A HTML5 bevezetésével ez a technológia egy tartósabb megoldást nyújtott a régebbi cookie-alapú tárolás helyett, mivel nagyobb adatmennyiséget (5-10 MB böngészőtől függően) képes kezelni, és nem kerül minden egyes szerverrel való kommunikáció során elküldésre, mint a cookie-k. A Local Storage egy egyszerű kulcs-érték páros rendszer, ahol az adatok a böngészőben maradnak akkor is, ha a felhasználó bezárja az alkalmazást vagy újraindítja a böngészőt.

A Local Storage különösen alkalmas olyan kisebb adatok tárolására, mint például:

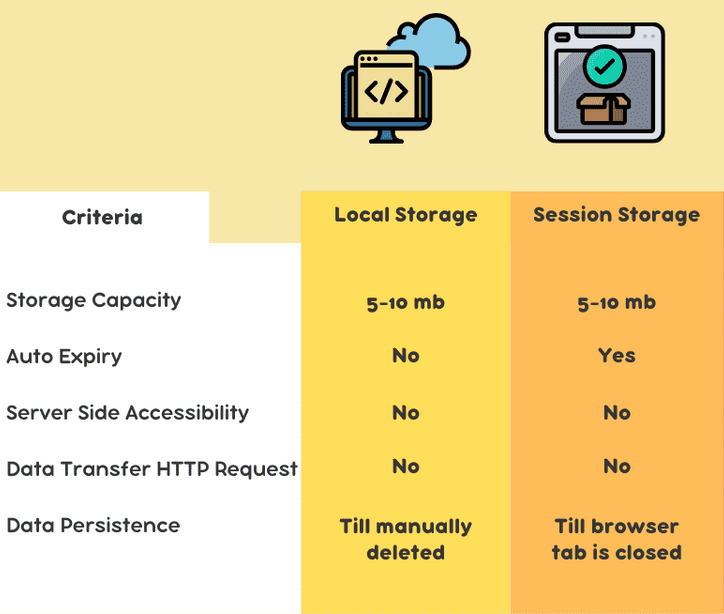
* Felhasználói beállítások (pl. témaválasztás, nyelvi preferenciák)
* Egyszerű statisztikai adatok (pl. elért pontszámok játékokban)
* Munkamenet előzményei vagy cache adatok

Hátránya, hogy az adatok csak szöveges formátumban tárolhatók, így komplex objektumokat JSON formátumba kell konvertálni tárolás előtt, majd újra vissza kell alakítani, ha használni szeretnénk.

A **Session Storage** a HTML5 másik kliensoldali adattárolási megoldása, amely lehetőséget biztosít az adatok ideiglenes tárolására egy adott böngészési munkamenet idejére. Ez a technológia hasonló a Local Storage-hoz, de alapvető különbség, hogy az itt tárolt adatok csak addig maradnak meg, amíg a felhasználó nem zárja be a böngészőablakot. Miután a munkamenet lezárul, az összes adat automatikusan törlődik, így biztosítva, hogy az ideiglenes információk ne maradjanak feleslegesen letárolva. Ezt a tulajdonságot számos felhasználási esetben kihasználhatjuk, amikor átmeneti adattárolásra van szükség.

A Session Storage tipikusan olyan helyzetekben használható, ahol az adatokra csak egy adott böngészési munkamenet során van szükség. Például egy weboldalon a felhasználó bejelentkezési állapotát követhetjük egy böngészési fülön belül anélkül, hogy az adatok megmaradnának a fül bezárása után. Emellett gyakran használják több lépéses űrlapoknál, ahol az adatoknak meg kell maradniuk az oldalak közötti navigáció során, de már nem szükségesek, miután a folyamat befejeződik. Továbbá alkalmas lehet átmeneti bevásárlókosár-adatok tárolására, vagy olyan dinamikus tartalmaknál, amelyek csak egy adott munkamenet alatt szükségesek.

Előnye, hogy az adatok automatikusan törlődnek a munkamenet végén, így nem kell aggódni a régi vagy felesleges információk eltávolítása miatt. Ezzel egy egyszerű és kényelmes megoldást kínál a rövid távú adatkezelésre, mivel az adatokat nem kell kézzel törölni vagy frissíteni, amikor azok már nem szükségesek. Ezen felül biztosítva van, hogy az adatok különállóan kezelhetők legyenek a különböző böngészőfülek között, azaz az egyik fülben tárolt adatok nem lesznek elérhetők egy másikban. Ezzel szemben hátránya, hogy mivel az adatok csak egy munkamenet alatt érhetők el, nem alkalmas hosszú távú adatmegőrzésre. Így például felhasználói beállítások, játékállások vagy egyéb, tartós adattárolást igénylő információk mentésére a Session Storage nem megfelelő. Emellett, bár az adatmennyisége általában 5-10 MB között van, amely elegendő lehet kisebb adatokhoz, nagyobb vagy összetettebb adatok kezelésére nem a legjobb megoldás, különösen olyan esetekben, ahol strukturált vagy relációs adatok tárolására van szükség.



**Az IndexedDB** egy fejlettebb böngésző alapú adatbázis, amely nagy mennyiségű adat tárolására és kezelésére alkalmas. Ez egy NoSQL-alapú adatbázis, amely aszinkron módon működik, és lehetővé teszi strukturált adatok, például objektumok, fájlok és bináris adatok tárolását.

Az IndexedDB különösen alkalmas összetett feladatokhoz:

* Nagy mennyiségű adatot tárolhat, példaként: játékhelyeket, adatbázisokat, képeket, videókat vagy gyorsítótár-fájlokat
* Lokálisan futó alkalmazások (offline módban is használható)
* Alkalmazásokhoz, amelyek megkövetelik, hogy az adatokat strukturált és relációs szerkezetben tárolják

Az IndexedDB támogatja a tranzakciókat, lekérdezéseket és indexeket, lehetővé téve az adatok hatékonyabb keresését és módosítását. Bár a használata bonyolultabb, rugalmassága miatt nagyobb kontrollt biztosít a fejlesztőknek.

**Local Storage és IndexedDB összehasonlítása:**

* **Adatmennyiség:** A Local Storage korlátozottan, néhány megabájt adat tárolására képes, míg az IndexedDB gyakorlatilag korlátlan mennyiségű adatot képes kezelni (a böngésző által megengedett határok között).
* **Használat egyszerűsége:** A Local Storage könnyebben használható, mivel egyszerű kulcs-érték párokat használ, míg az IndexedDB komplexebb API-t igényel.
* **Teljesítmény:** Az IndexedDB nagyobb adatmennyiségek és összetett adatszerkezetek kezelésére optimalizált, míg a Local Storage egyszerűbb feladatokra alkalmas.

Mindkét technológia szerepe kiemelkedő a modern webalkalmazásokban, ahol fontos a hatékony helyi adattárolás. Az egyszerűbb adatokhoz, mint a felhasználói beállítások vagy a pontszámok, a Local Storage teljes mértékben megfelelő, míg a komplexebb, strukturált adatokhoz, például egy játékállás mentéséhez vagy egy offline alkalmazás működtetéséhez, az IndexedDB nyújt használhatóbb megoldást.

**A WebSockets technológia:**

Kulcsponti szerepet játszik a HTML5 játékfejlesztésben: főleg az online multiplayer játékoknál. A WebSocket egy olyan protokoll, amely lehetővé teszi a valós idejű, kétirányú kommunikációt a kliens és a szerver között. Ez azt jelenti, hogy a játékosok közötti interakciók azonnal megjelennek a játékban, függetlenül attól, hogy hol helyezkednek el a világban. Ezen felül a WebSockets hatékonyabb adatátvitelt biztosít, mint a hagyományos HTTP-alapú kommunikáció, mivel a kapcsolat egyszer létrejön, majd nyitva marad, így az üzenetek küldése és fogadása gyorsabban és kevesebb késéssel történik. Ez alapvető fontosságú a sima és zökkenőmentes játékélmény biztosításához, különösen azokban a játékokban, ahol minden pillanat számít, például valós idejű stratégiai vagy lövöldözős játékokban.

Összességében a HTML5 játékfejlesztés egyik legnagyobb előnye a platformfüggetlenség. A HTML5 játékok szinte minden modern böngészőben futtathatók, legyen szó asztali számítógépekről vagy mobil eszközökről. Emellett a fejlesztéshez szükséges eszközök és technológiák széles körben hozzáférhetők, és gyakran ingyenesek, ami különösen vonzóvá teszi őket a fejlesztők és kisebb stúdiók számára. Ez a kombináció lehetővé teszi, hogy a játékfejlesztők széles közönséghez juttassák el alkotásaikat, miközben minimalizálják a fejlesztési költségeket és az időbeli ráfordítást.

# Digitális Tartalmak Felhasználása: Licencjogok és Fair Use

A digitális tartalom előállítása során fontos tisztában lenni a tartalom felhasználásának szabályaival, különösen a különböző licencjogokkal, amelyek meghatározzák, hogyan és milyen feltételek mellett használhatók fel a képek és más művészeti alkotások. Az alábbiakban bemutatom a leggyakrabban használt licencformákat, és részletesen kifejtem azok jelentését és alkalmazását egyaránt.

## Általános áttekintés:

**CC (Creative Commons):**  
Ezen licencek célja, hogy rugalmas lehetőségeket biztosítsanak a szerzői jogok kezelésére, lehetővé téve a művek széleskörű felhasználását, átalakítását és terjesztését. A CC licencek több típusra oszlanak, amelyek különböző feltételeket tartalmaznak a művek felhasználására vonatkozóan. Az alapvető követelmény, hogy a szerző nevét és a forrást megfelelően feltüntessék, de a specifikus licenc feltételei változhatnak attól függően, hogy milyen mértékű engedékenységet vagy korlátozást biztosítanak a felhasználók számára. A legújabb verziók, mint a CC 4.0, általában fejlettebb nemzetközi alkalmazhatóságot és korszerűbb szabályozást kínálnak az előző verziókhoz képest.

**OGA (OpenGameArt Attribution):**  
Az OpenGameArt által kiadott OGA licencek hasonlóak a Creative Commons licencekhez, de kifejezetten a játékfejlesztéshez lettek optimalizálva. Ezek a licencek lehetővé teszik a művek felhasználását játékokban, amennyiben a szerző nevét és az eredeti forrást megfelelően feltüntetik. Az OGA 4.0-s verziója továbbfejlesztett változata a 3.0-s verziónak, és fokozott rugalmasságot és kompatibilitást biztosít a játékfejlesztési projektek számára.

**GPL (General Public License):**  
Szabad szoftverlicenc, amely lehetővé teszi a szoftver szabad felhasználását, módosítását és terjesztését, feltéve, hogy az eredeti licenc feltételeit megőrizzük. A GPL 3.0-s verziója fejlettebb jogi védelmet és nagyobb biztonságot nyújt a felhasználók számára a GPL 2.0-s verziójához képest, biztosítva a szabad szoftverek védelmét és terjesztését.

Ezek a licencek különböző jogosultságokat és korlátozásokat tartalmaznak, ezért fontos, hogy a projekt igényeinek megfelelő licenceket válasszunk.

## Creative Commons Licenc típusok:

A Creative Commons (CC) licencek 6 különböző típusa közül választhatunk, az engedélyezési feltételek fokozatos szigorodása mellett. Az alábbiakban látható a különböző licencek jellemzői és feltételei:

**CC0 (Creative Commons Zero)**  
A CC0 egy közkinccsé tételi eszköz, amely lehetővé teszi a szerzők számára, hogy lemondjanak minden szerzői jogukról, és a művet a világszerte elérhető közkinccsé tegyék. A CC0 lehetővé teszi a mű szabad felhasználását, remixelését és adaptálását bármilyen célra és bármilyen formában, feltételek nélkül.

**CC BY (Creative Commons Attribution)**  
Ez a licenc lehetővé teszi a felhasználók számára, hogy szabadon terjesszék, remixeljék, adaptálják és építkezzenek a műből bármilyen formában vagy médiumon, feltéve, hogy a szerző nevét megfelelően megemlítik. Kereskedelmi célú felhasználás is megengedett. A CC BY licenc az alábbiakat tartalmazza:

* **BY:** A művet felhasználó személy vagy szervezet köteles feltüntetni a szerző nevét és az eredeti mű forrását.

**CC BY-SA (Creative Commons Attribution-ShareAlike)**  
Ez a licenc szintén lehetővé teszi a művek szabad felhasználását, remixelését és adaptálását, amennyiben a szerző nevét megemlítik. A licenc lehetővé teszi a kereskedelmi felhasználást is. Ha módosítjuk a művet, az új változatot azonos feltételek mellett kell licencelni. A CC BY-SA licenc az alábbiakat tartalmazza:

* **BY:** A szerző nevét és a forrást meg kell említeni.
* **SA:** Az adaptált műveket ugyanazon feltételek mellett kell közzétenni.

**CC BY-NC (Creative Commons Attribution-NonCommercial)**  
Ez a licenc lehetővé teszi a művek szabad terjesztését, remixelését és adaptálását, de csak nem kereskedelmi célokra. A szerző nevét itt is meg kell említeni. A CC BY-NC licenc az alábbiakat tartalmazza:

* **BY:** A szerző nevét és a forrást meg kell említeni.
* **NC:** Csak nem kereskedelmi célú felhasználás engedélyezett. (Non-Commercial)

**CC BY-NC-SA (Creative Commons Attribution-NonCommercial-ShareAlike)**  
Ez a licenc hasonló a CC BY-NC licenchez, de további feltétel, hogy a módosított műveket azonos feltételek mellett kell közzétenni. A szerző nevét meg kell említeni, és a kereskedelmi felhasználás nem megengedett. A CC BY-NC-SA licenc az alábbiakat tartalmazza:

* **BY:** A szerző nevét és a forrást meg kell említeni.
* **NC:** Csak nem kereskedelmi célú felhasználás engedélyezett. (Non-Commercial)
* **SA:** Az adaptált műveket ugyanazon feltételek mellett kell közzétenni.

**CC BY-ND (Creative Commons Attribution-NoDerivatives)**  
Ez a licenc lehetővé teszi a mű másolását és terjesztését bármilyen formában, de csak az eredeti, változatlan formában. A kereskedelmi felhasználás engedélyezett. A szerző nevét meg kell említeni. A CC BY-ND licenc az alábbiakat tartalmazza:

* **BY:** A szerző nevét és a forrást meg kell említeni.
* **ND:** Nincsenek engedélyezett módosítások vagy átdolgozások.

**CC BY-NC-ND (Creative Commons Attribution-NonCommercial-NoDerivatives)**  
Ez a licenc a mű másolását és terjesztését kizárólag az eredeti formájában, nem kereskedelmi célokra engedélyezett. A szerző nevét meg kell említeni. A CC BY-NC-ND licenc az alábbiakat tartalmazza:

* **BY:** A szerző nevét és a forrást meg kell említeni.
* **NC:** Csak nem kereskedelmi célú felhasználás engedélyezett.
* **ND:** Nincsenek engedélyezett módosítások vagy átdolgozások.

# Felhasznált források:

<https://hu.wikipedia.org/wiki/Videójáték>

<https://hu.wikipedia.org/wiki/Játékszer>

<https://www.grandviewresearch.com/industry-analysis/video-game-market>

<https://www.bcg.com/publications/2023/drivers-of-global-gaming-industry-growth>

<https://newzoo.com/resources/blog/last-looks-the-global-games-market-in-2023>

<https://en.wikipedia.org/wiki/Game_engine>

<https://spotlighthungary.hu/2024/01/legnepszerubb-jatek-keszito-eszkozok-es-platformok/>

<https://docs.unity3d.com/Manual/index.html>

<https://dotnet.microsoft.com/en-us/apps/games/unity>

<https://www.unrealengine.com/en-US/unreal-engine-5>

<https://developer.mozilla.org/en-US/docs/Games/Introduction_to_HTML5_Game_Development>

https://medium.com/stackanatomy/svg-vs-canvas-a-comparison-1b58e6c84326

<https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>

https://www.loginradius.com/blog/engineering/guest-post/local-storage-vs-session-storage-vs-cookies/

https://creativecommons.org/share-your-work/cclicenses/