

SZAKDOLGOZAT



MISKOLCI EGYETEM

Az én szakdolgozatom

Készítette:

Szendrei Gábor

Programtervező informatikus

Témavezető:

Dr. Vadon Viktória

MISKOLC, 2023

MISKOLCI EGYETEM

Gépészmérnöki és Informatikai Kar

Alkalmazott Matematikai Intézeti Tanszék

Szám:

SZAKDOLGOZAT FELADAT

Szendrei Gábor (V9ZK10) BSc programtervező informatikus jelölt részére.

A szakdolgozat tárgyköre: kulcsszavak, hasonlók

A szakdolgozat címe: A dolgozat címe

A feladat részletezése:

Ide kell a feladatkiírásban szereplő szöveget betenni.

(Kisebb tagolás lehet benne, hogy jól nézzen ki.)

Témavezető: Témavezető neve (beosztása)

A feladat kiadásának ideje:

.....
szakfelelős

EREDETISÉGI NYILATKOZAT

Alulírott **Szendrei Gábor**; Neptun-kód: **V9ZK10** a Miskolci Egyetem Gépészmérnöki és Informatikai Karának végzős Programtervező informatikus szakos hallgatója ezennel büntetőjogi és fegyelmi felelősségem tudatában nyilatkozom és aláírással igazolom, hogy *Szakedolgozat Címe* című szakdolgozatom saját, önálló munkám; az abban hivatkozott szakirodalom felhasználása a forráskezelés szabályai szerint történt.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- szószerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát megismertem, és tudomásul veszem, hogy plágium esetén szakdolgozatom visszautasításra kerül.

Miskolc, év hó nap

.....
Hallgató

- | | |
|-------|---------------|
| | |
| dátum | témavezető(k) |

- | | |
|------------------------------|-----------------------------|
| témavezető (dátum, aláírás): | konzulens (dátum, aláírás): |
| | |
| | |
| | |

- | | |
|-------|---------------|
| | |
| dátum | témavezető(k) |

- | | |
|-------|---------------|
| | |
| dátum | témavezető(k) |

- dátum szakfelelős

- Miskolc,
a Záróvizsga Bizottság Elnöke

Tartalomjegyzék

1. Bevezetés	1
2. Konceptió	3
2.1. Bevezetés a Játékfejlesztés Világába	3
2.1.1. Mit jelent játékot fejleszteni?	3
2.1.2. A játékfejlesztés kihívásai és lehetőségei	3
2.1.3. Az alkalmazott fejlesztési módszerek áttekintése	4
2.2. Játékfejlesztő Könyvtárak és Eszközök	4
2.2.1. Unity: A 3D játékfejlesztés platformja	4
2.2.2. Java: Népszerű programozási nyelv játékfejlesztéshez	4
2.2.3. C#: Népszerű programozási nyelv játékfejlesztéshez	5
2.2.4. C++: A játékfejlesztők egyik kedvelt nyelve	5
2.3. Python a játékfejlesztésben	5
2.3.1. Miért jó a Python?	5
2.3.2. Python Alapjai	6
2.3.3. A Python és a Játékfejlesztés	6
2.4. Pygame	6
2.4.1. Pygame lehetőségei	7
2.4.2. Pygame További Tudnivalók	7
2.5. Pygame és Ren'Py Összehasonlítása	7
2.5.1. Célcsoport	7
2.5.2. Felhasználói felület	8
2.5.3. Felépítmény	8
2.5.4. Funkciók	8
2.5.5. Egyéb szempontok	8
2.5.6. Összegzés	8
2.6. Az Elérhető Eredmények	9
2.6.1. Pygame és Más Eszközökkel Készített Játékok Bemutatása . . .	9
2.6.2. Sikeres Projektek és Kreatív Alkotások Példái	9
2.6.3. Kutatási és Tanulási Célkitűzések	9
2.6.4. Oktatás	10
2.6.5. Kognitív kutatás	10
2.6.6. Szórakoztató és kreatív projektek	10
2.6.7. Prototípusok és koncepciók	11
2.6.8. Következtetés	11
2.7. Összegzés	11

3. Tervezés	12
3.1. Táblázatok	12
3.2. Ábrák	12
3.3. További környezetek	12
4. Megvalósítás	14
5. Tesztelés	16
6. Összefoglalás	17
Források	18

1. fejezet

Bevezetés

A számítógépes játékok már régóta a szórakozás egyik legnépszerűbb formáját képviselik. Az idők során folyamatos fejlődésen mentek keresztül, és ma már számtalan lehetőséget kínálnak a játékosoknak. Ez az oldal arra szolgál, hogy bemutassa a számítógépes játékok világát, és megmutassa, hogy milyen élményeket nyújtanak.

A számítógépes játékok lehetővé teszik számunkra, hogy elmerüljünk olyan világokban, amelyekben különböző fantáziavilágokban kalandozhatunk, izgalmas cselekmények részesei lehetünk, és versenyezhetünk a világ minden tájáról érkező játékosokkal. Az interaktivitás a játékok egyik fő jellemzője, hiszen a játékosok döntéseket hozhatnak, irányíthatják a karaktereket és befolyásolhatják a játék alakulását, így személyes élményeket és kalandokat élhetnek át.

A számítógépes játékokban a grafika és a hang is kulcsfontosságú szerepet játszik. A modern grafikus motorok és a kiváló minőségű hangeffektek valósághű és lenyűgöző világokat teremtenek, amelyekbe a játékosok könnyedén belefeledkezhetnek. A 3D-s grafika és a virtuális valóság (VR) technológia pedig még inkább fokozza a valóságérzetet, lehetővé téve, hogy teljes mértékben elmerüljünk a játék világában.

A számítógépes játékoknak számos műfaja létezik, így mindenki megtalálhatja a saját ízlésének megfelelő játékot. Akciójátékok, stratégiai játékok, szerepjátékok, sportjátékok, horror játékok és még sok más közül választhatunk, mindegyiknek saját kihívásai és élményei vannak.

Emellett a számítógépes játékoknak társadalmi szerepük is van. A többjátékos módban lehetőség van csapatban játszani, vagy akár versenyezni más játékosokkal online. Ez segít kapcsolatokat építeni és barátokat szerezni a világ minden tájáról érkező emberekkel.

Az innováció és a fejlesztés is fontos szerepet játszik a számítógépes játékok világában. A technológia fejlődése új és izgalmas lehetőségeket kínál, mint például a lenyűgöző grafika és hanghatások, valamint a mesterséges intelligencia alkalmazása a játékokban. Az adatgyűjtés és az adaptáció révén a játékfejlesztők képesek személyre szabni a játékelményt a játékosok számára.

Az eSport egyre népszerűbbé válik, és hatalmas növekedési potenciállal rendelkezik. Versenyek és ligák alakulnak ki, ahol a legjobb játékosok hatalmas pénzdíjakért versenyeznek, és mindezt élvezhetik a nézők is.

Azonban nem szabad elfelejtenünk, hogy a számítógépes játékoknak vannak kritikusai is, akik aggodalmakat vetnek fel a játékfüggőséggel és más negatív hatásokkal kapcsolatban. Fontos, hogy mértékkel játszunk, és törekedjünk a kiegyensúlyozott életmódra.

Össességében a számítógépes játékok kiváló lehetőséget nyújtanak a szórakozásra, és a technológia folyamatos fejlődésének köszönhetően még évekig fontos szerepet fognak játszani a szórakoztatóiparban.

A szakdolgozatom célja egy 2-dimenziós akció-kaland játék fejlesztése, melyben a játékosok teljesítsenek különféle izgalmas küldetéseket, miközben felfedezik a szigetet és barlangjait és kihívásokkal teli kalandokban vesznek részt.

Dolgozatomban részletes betekintést nyújtok a Python programozási nyelv és a pygame könyvtár használatába. Emellett összehasonlító elemzést végezek a pygame és a Ren'py könyvtárak között, és bemutatom a játék implementációjának részleteit is.

2. fejezet

Koncepció

2.1. Bevezetés a Játékfejlesztés Világába

A játékfejlesztés a modern szórakoztatóipar egyik legszerteágazóbb és izgalmasabb területe, amely számtalan lehetőséget rejt magában a kreativitás kibontakoztatására és az élmények megteremtésére. Ennek a fejezetnek a kezdetén tekintsünk részletesebben arra, hogy mit jelent játékot fejleszteni, milyen kihívásokkal jár, és milyen lehetőségeket kínál.

2.1.1. Mit jelent játékot fejleszteni?

Játékot fejleszteni egy olyan folyamat, amely során virtuális vagy fizikai játékokat tervezünk, készítünk és tesztelünk. Ezek a játékok szórakoztatnak, kihívások elé állítanak, vagy éppen történeteket mesélnek el a játékosoknak. A játékfejlesztés során számos különböző területet érintünk, mint például a grafika, a hang, a programozás, a játéktervezés és a narratíva.

A játékfejlesztés során a következő elemeket kell figyelembe venni:

Játéktervezés: A játékmechanizmusok, pályatervezés, karakterek és sztori kidolgozása. Grafika és dizájn: A játékvilág megtervezése, karakterek és tájak kinézetének megalkotása. Hang és zene: A játékhangulat meghatározása zenei és hanghatásokkal. Programozás: A játék mechanizmusainak és logikájának implementálása.

2.1.2. A játékfejlesztés kihívásai és lehetőségei

A játékfejlesztés izgalmas, de komplex folyamat, amely számos kihívást rejt magában:

Technikai kihívások: A játékfejlesztéshez fejlett szoftveres és hardveres ismeretekre van szükség. A játék motorok, programozási nyelvek, és grafikai eszközök használata összetett feladatokkal jár.

Kreativitás: A jó játékok egyediséget és kreativitást követelnek meg a játéktervezőktől. Az új és izgalmas játékmechanizmusok kitalálása kulcsfontosságú.

Projektmenedzsment: A játékfejlesztés projektek hosszú és komplex folyamatok, amelyek határidőket és költségvetéseket igényelnek. A megfelelő projektmenedzsment kritikus fontosságú.

Felhasználói élmény: A játékosok elégedettségének és élvezetének biztosítása kiemelten fontos. A játéktervezés és felhasználói felület optimalizálása elengedhetetlen.

Ugyanakkor a játékfejlesztésben óriási lehetőségek rejlenek:

Kreatív kibontakozás: A játékfejlesztés lehetőséget ad a kreativitás megnyilvánulására, ahol csak a képzelet szab határt.

Közösség és verseny: A játékfejlesztők részt vehetnek aktív közösségekben, és akár versenyeken is, ahol megmutathatják tehetségüket és fejlődésüket.

Szórakoztatás: A jó játékok milliók számára jelentenek kikapcsolódást és szórakozást, és hűséges rajongótábort hozhatnak létre.

2.1.3. Az alkalmazott fejlesztési módszerek áttekintése

[1] A játékfejlesztés során többféle módszert lehet alkalmazni a hatékony és strukturált munka érdekében. Néhány példa:

Vízesés modell: Az összes fejlesztési fázist lineárisan követi egymás után, például tervezés, implementáció, tesztelés, kiadás.

Agilis fejlesztés: Rugalmasabb megközelítés, ahol a fejlesztés ciklusokban (sprintekben) történik, és gyakran kapunk visszajelzéseket, amelyek alapján alkalmazkodunk.

Iteratív fejlesztés: Az alkalmazott módszer a projektenként változhat, és a folyamatos visszajelzés és fejlesztés a kulcsa.

A megfelelő módszer kiválasztása a projekt jellegétől, méretétől és a csapatdinamikától függ. A játékfejlesztés során fontos az alkalmazott módszer hatékony és rugalmas használata a kihívások leküzdésére és a célkitűzések elérésére.

Ez a fejezet célja, hogy bevezessen a játékfejlesztés izgalmas világába, megértsük a kihívásokat és lehetőségeket, valamint megismerjük az alkalmazott fejlesztési módszerek alapjait. A további fejezetekben részletesen megvizsgáljuk az egyes területeket és technológiákat, amelyek a játékfejlesztéshez.

2.2. Játékfejlesztő Könyvtárak és Eszközök

A játékfejlesztéshez elérhető könyvtárak és eszközök rendkívül fontosak a fejlesztők számára, hiszen segítenek a játékok fejlesztésében és optimalizálásában. Ebben a részben áttekintjük néhány közismert könyvtárat és eszközt, amelyeket játékfejlesztéshez használnak.

2.2.1. Unity: A 3D játékfejlesztés platformja

[4] Az Unity az egyik legnépszerűbb és legelismertebb játékfejlesztési platform, amely lehetővé teszi 2D és 3D játékok készítését. Az Unity fejlesztői számára egy teljes körű fejlesztői környezetet biztosít, amely integrált fejlesztői eszközöket, grafikai motorokat és folyamatos támogatást tartalmaz.

Az Unity könnyen használható, és támogatja a cross-platform fejlesztést, így játékokat készíthetünk számítógépekre, mobil eszközökre és konzolokra egyaránt. Emellett erős vizualizációs és fizikai motorokat is kínál, amelyek segítik a valósághű játékelmény kialakítását.

2.2.2. Java: Népszerű programozási nyelv játékfejlesztéshez

[2] Java egy platformfüggetlen, objektumorientált programozási nyelv, amelyet széles körben használnak játékfejlesztéshez, különösen Android platformra. Java játékok

készítéséhez a Java ME (Micro Edition) keretrendszert is felhasználhatjuk, amelyet mobiljátékokhoz és egyéb kis méretű alkalmazásokhoz fejlesztettek ki.

A Java platformfüggetlensége lehetővé teszi, hogy a játékokat több platformra, például számítógépre és mobil eszközökre is exportáljuk. A Java ME segítségével kifejezetten mobiljátékokat is fejleszthetünk. A Java erős közösséggel rendelkezik, így könnyű hozzáférést nyújt különböző fejlesztői eszközökhöz és könyvtárakhoz.

2.2.3. C#: Népszerű programozási nyelv játékfejlesztéshez

C# egy modern, objektumorientált programozási nyelv, amelyet széles körben használnak játékfejlesztéshez, különösen az Unity motorban. Az Unity C#-t használ a játéklógika és a szkriptelés megvalósításához, ami lehetővé teszi a fejlesztők számára a cross-platform játékok készítését.

C# erős és könnyen tanulható nyelv, amely támogatja a modern programozási paradigmákat. Az Unity motor segítségével a C# nyelvet használó fejlesztők könnyedén exportálhatnak játékokat több platformra, például számítógépekre, konzolokra és mobil eszközökre.

2.2.4. C++: A játékfejlesztők egyik kedvelt nyelve

[3] A C++ egy erős és hatékony programozási nyelv, amelyet gyakran alkalmaznak játékfejlesztéshez, különösen a nagy teljesítményű játékokhoz és konzolplatformokhoz. A C++ segítségével közvetlenül a hardverre lehet programozni, ami nagy szabadságot és teljesítményt biztosít.

A C++ a játékfejlesztők egyik kedvelt nyelve, mivel lehetővé teszi a nagy teljesítményű játékok létrehozását és a hardverhez való közvetlen hozzáférést. Emellett számos játékfejlesztési könyvtár és motor támogatja a C++ nyelvet.

Ezen eszközök és nyelvek kiválasztása a projekt specifikus igényeitől és a fejlesztői készségektől függ. Fontos megérteni, hogy minden eszköznek és nyelvnek megvannak a saját előnyei és korlátai, és azokat a projekt céljaihoz és a csapat készségeihez kell igazítani.

2.3. Python a játékfejlesztésben

A Python egy kiválóan használható programozási nyelv a játékfejlesztéshez, és sok előnnyel rendelkezik a fejlesztők és a játéktervezők számára. Ebben a fejezetben kifejtjük, miért érdemes Pythonnal dolgozni játékok tervezésekor, és milyen alapvető tulajdonságok és lehetőségek teszik ezt a nyelvet vonzóvá a játékfejlesztés világában.

2.3.1. Miért jó a Python?

1. Olvashatóság és Egyszerűség: Python kódot írni könnyű és gyors. A Python nyelv szintaxisa rendkívül olvasható és hasonlít az angol nyelvre, ami megkönnyíti a kód értelmezését. A könnyű olvashatóság a fejlesztési időt csökkentheti és csökkentheti a hibák számát.

2. Gyors Fejlesztés: A Python lehetővé teszi a gyors prototípusok létrehozását. A gyors prototípusok segítenek a játék ötleteinek gyors validálásában és tesztelésében, mielőtt hosszú fejlesztési ciklusokba kezdünk.

3. Széles Közösség és Támogatás: A Python rendelkezik egy nagy és elkötelezett fejlesztői közösséggel, ami számos kiegészítő könyvtárat és eszközt kínál a játékfejlesztőknek. Ez a közösség folyamatosan fejleszti és frissíti a nyelvet és az eszközöket.

2.3.2. Python Alapjai

- **Változók és Adattípusok:** A Python változókat dinamikusan típusozott nyelvként kezeli, ami azt jelenti, hogy nem kell előzetesen meghatározni az adattípusukat. Például, egy változó lehet szám, szöveg vagy akár egy lista.
- **Függvények:** A függvények segítségével újrafelhasználható kódot írhatunk. Függvényekkel könnyen strukturáltá tehetjük a kódot, ami különösen hasznos a játékfejlesztés során.
- **Listák és Szótárak:** A Python támogatja a listákat (list) és szótárakat (dictionary), amelyek nagyszerűen alkalmasak adatok tárolására és kezelésére. A listák sorozatokat tartalmaznak, míg a szótárak kulcs-érték párokat tárolnak.
- **Ciklusok és Feltételek:** A Python támogatja a ciklusokat és a feltételeket, amelyek lehetővé teszik a kód futtatását bizonyos feltételek vagy események alapján. A ciklusok és a feltételek segítségével a játékfejlesztők irányítani tudják a játékmenetet.
- **Objektumorientált Programozás:** A Python objektumorientált nyelv, amely lehetővé teszi az osztályok és objektumok létrehozását. Az osztályok és objektumok segítségével a játékfejlesztők könnyen kezelhetik a játékbeli elemeket.
- **Modulok és Csomagok:** A Python modulokat és csomagokat használ a kód szervezéséhez és strukturálásához. A modulok és csomagok lehetővé teszik a kód újrafelhasználását és a fejlesztési idő csökkentését.

2.3.3. A Python és a Játékfejlesztés

Pythonnal a játékfejlesztők számos különböző könyvtárat és keretrendszert használhatnak játékok készítéséhez. Egyik ilyen könyvtár a "Pygame", amely egy népszerű és könnyen tanulható keretrendszer a 2D játékokhoz. A Pygame számos olyan eszközt és funkciót nyújt, amelyek segítik a játékfejlesztőket a grafikai megjelenítés, hangkezelés és inputkezelés terén.

A Python és a Pygame kombinációja lehetővé teszi a játékfejlesztők számára, hogy gyorsan és hatékonyan hozzanak létre játékokat, és egy erős fejlesztői közösség és dokumentáció áll rendelkezésre a támogatáshoz és az útmutatáshoz. Az egyszerűség és az olvashatóság révén a Python segíti a fejlesztőket abban, hogy a játékmechanizmusokra és a játékelményre koncentráljanak anélkül, hogy túl sok időt kellene tölteni a kód megértésével és karbantartással.

2.4. Pygame

A Pygame egy olyan Python modulokból álló gyűjtemény, amely a videójátékok és multimédiás programok fejlesztését teszi lehetővé a Python programozási nyelven. A

Pygame lényegében egy réteget ad az SDL (Simple DirectMedia Layer) kiváló könyvtár fölé, amely lehetővé teszi, hogy teljesen kiépített játékokat és multimédiás alkalmazásokat hozzunk létre Pythonban.

2.4.1. Pygame lehetőségei

A Pygame rendkívül sokoldalú eszközöket és lehetőségeket kínál a játékfejlesztők számára:

Grafikai megjelenítés: A Pygame lehetővé teszi a grafikus elemek létrehozását és kezelését, beleértve a sprite-okat, háttérképeket és rajzolási funkciókat is.

Hangkezelés: A könyvtár lehetővé teszi hangfájlok lejátszását, hanghatások és zene hozzáadását a játékhoz.

Inputkezelés: Pygame segítségével könnyedén kezelhetők a felhasználói inputok, például a billentyűzet és egér események.

Multiplatform támogatás: Pygame rendkívül hordozható, és szinte minden platformon és operációs rendszeren fut, beleértve Linuxot, Windowst, MacOS-t és másokat.

Hordozhatóság: Pygame alkalmazható számos eszközön és operációs rendszeren, beleértve a kézi eszközöket, játékkonzolokat és a One Laptop Per Child (OLPC) számítógépet is.

Egyszerűség: A Pygame könnyen megtanulható és használható, és kiválóan alkalmas fiatalabb és idősebb játékfejlesztők számára egyaránt.

Modularitás: A Pygame lehetőséget ad arra, hogy a különböző modulokat külön-külön inicializálja és használja, így testreszabhatja a fejlesztést az igényeinek megfelelően.

2.4.2. Pygame További Tudnivalók

Moduláris Felépítés A Pygame rendkívül moduláris felépítésű, ami azt jelenti, hogy könnyen testre szabható és bővíthető. A fejlesztők kiválaszthatják azokat a modulokat, amelyekre szükségük van a projektjükben, és elhagyhatják azokat, amelyekre nincs szükségük. Ezáltal minimalizálhatják a projekt méretét és optimalizálhatják a teljesítményt.

Széleskörű Közösség és Források A Pygame hatalmas fejlesztői közösséggel rendelkezik, ami azt jelenti, hogy rengeteg dokumentáció, tutorial és fórum áll rendelkezésre a segítségnyújtáshoz és a problémamegoldáshoz. Az aktív közösség folyamatosan fejleszti és frissíti a Pygame-et, így a fejlesztők mindig naprakész forrásokhoz férhetnek hozzá.

Könnyen Tanulható A Pygame olyan egyszerűen használható, hogy akár gyerekek és fiatalabb játékfejlesztők is könnyen megtanulhatják a használatát. A kezdeti lépések után a fejlesztők gyorsan építhetnek fel játékokat és alkalmazásokat a Pygame segítségével.

2.5. Pygame és Ren'Py Összehasonlítása

2.5.1. Célcsoport

Pygame: Általános célú keretrendszerként szolgál, ami azt jelenti, hogy szinte bármilyen típusú játék készítéséhez használható. Legyen szó akció-játékról, platformjátékról, vagy akár puzzle-játékról, a Pygame alkalmas erre. **Ren'Py:** A Ren'Py viszont kifejezetten a

szövegalapú visual novelek (interaktív történetek) készítésére specializálódott. Itt a fő hangsúly a narratíván, karakterek párbeszédein és képeken van.

2.5.2. Felhasználói felület

Pygame: A Pygame használata viszonylag egyszerű, de bizonyos alapvető programozási ismeretekre van szükség. Fejlesztőknek ismeretekre van szükségük a Python nyelvben és a játékfejlesztés alapjaiban. Ren'Py: A Ren'Py használata rendkívül intuitív és egyszerű, még a programozásban nem jártas felhasználók számára is. Ez teszi lehetővé a könnyű interaktív szöveges játékok létrehozását.

2.5.3. Felépítmény

Pygame: A Pygame egy objektumorientált keretrendszer, ahol a játékot különböző objektumokból építik fel, és a fejlesztőknek az objektumok közötti interakciókat kell kezelniük. Ren'Py: A Ren'Py egy szöveg-alapú keretrendszer, ahol a játékot szövegből és grafikából építik fel. Az objektumok és interakciók kezelése itt kevésbé hangsúlyos, a fókusz a narratívára összpontosul.

2.5.4. Funkciók

Pygame: A Pygame számos alapvető funkciót kínál, mint például a grafika, a hang és a bevitel kezelése. Fejlesztőknek nagyobb szabadságot ad azáltal, hogy saját logikát és rendszereket hozhatnak létre. Ren'Py: A Ren'Py speciális funkciókat kínál a visual novelek készítéséhez, mint például a szöveg effektek, a zene és a képkockák kezelése. A keretrendszer célja a visual novel műfaj specifikus igényeinek kielégítése.

2.5.5. Egyéb szempontok

Pygame: A Pygame nyílt forráskódú és ingyenesen elérhető, valamint több platformon futtatható, beleértve a Linuxot, Windowst és MacOS-t is. Rugalmas és testreszabható, ami lehetővé teszi a fejlesztők számára a saját játékmotorjuk létrehozását. Ren'Py: A Ren'Py mellett elérhető kereskedelmi licenc is, amely további támogatást és lehetőségeket kínál. Főként Windowson és macOS-en futtatható, és kifejezetten a visual novelek készítésére specializált, széleskörű szövegkezelési funkciókat nyújt.

Játékstílus: A Pygame és a Ren'Py különböző játékstílusokhoz alkalmas. A Pygame ideális akció-, platform- és puzzle-játékokhoz, míg a Ren'Py a visual novelekhez. Tapasztalat: A Pygame használatához bizonyos alapvető programozási ismeretekre van szükség. A Ren'Py viszont könnyen használható még a programozásban nem jártas felhasználók számára is. Költség: A Pygame nyílt forráskódú és ingyenesen elérhető. A Ren'Py mellett elérhető kereskedelmi licenc is, amely további támogatást és lehetőségeket kínál.

2.5.6. Összegzés

A Pygame és a Ren'Py két népszerű keretrendszer a játékfejlesztéshez. A Pygame általános célú keretrendszer, amely bármilyen típusú játék készítéséhez használható. A Ren'Py pedig egy visual novelek készítésére specializált keretrendszer.

A két keretrendszer kiválasztásakor a következő szempontokat érdemes figyelembe venni:

Játékstílus: Milyen típusú játékot szeretne készíteni? Tapasztalat: Mennyi programozási tapasztalattal rendelkezik? Költség: Mennyi pénzt szán a keretrendszerre? A megfelelő keretrendszer kiválasztása segíthet abban, hogy gyorsabban és könnyebben készítsen professzionális minőségű játékokat.

2.6. Az Elérhető Eredmények

A Pygame és más eszközökkel készített játékok bemutatása nyit egy ablakot a játékfejlesztés izgalmas világába. Ebben a fejezetben megismerkedünk néhány sikeres projekttel és kreatív alkotással, amelyeket a Pygame és más hasonló eszközök segítségével hoztak létre. Emellett megvizsgáljuk, hogy milyen kutatási és tanulási célkitűzéseket lehet elérni ezen eszközökkel.

2.6.1. Pygame és Más Eszközökkel Készített Játékok Bemutatása

A Pygame egy olyan eszköz, amely lehetővé teszi a fejlesztők számára, hogy kreatív és sokoldalú játékokat hozzanak létre. Néhány példa sikeres Pygame projektekre:

- Flappy Birdklón: Az egyszerű, mégis addiktív játékstílus népszerűségét kihasználva sok fejlesztő készített saját változatot a Pygame segítségével.
- Bástion(Supergiant Games): A Bástion egy kritikailag elismert akció-RPG, amelynek fejlesztése során a Pygame-t használták a prototípusok készítéséhez és a játékmechanikák teszteléséhez.
- Darwinia(Introversion Software): A Darwinia egy sikeres stratégiai játék, amely részben a Pygame inspirációjából született, bár más eszközökkel fejlesztették.

2.6.2. Sikeres Projektek és Kreatív Alkotások Példái

A játékfejlesztők és alkotók széles skáláját inspirálta a lehetőség, hogy Pygame és hasonló eszközökkel alkotói ötleteiket valósítsák meg. Néhány kreatív példa:

- Undertale(Toby Fox): Az Undertale egy különleges és elbűvölő szerepjáték, amely különböző stílusokat és interakciós mechanizmusokat kombinál a Pygame segítségével.
- Minecraft(Mojang): Bár a Minecraft egy saját motorral rendelkező játék, a Pygame inspirálta a megjelenésekor, és az alfa verziója Pygame segítségével készült.

Kreatív vizuális művészetek: A Pygame és hasonló eszközök lehetővé teszik a művészek számára, hogy interaktív műveket hozzanak létre, amelyek például installációk, képkollázsok vagy interaktív festmények.

2.6.3. Kutatási és Tanulási Célkitűzések

A Pygame és más játékfejlesztő eszközök használata több területen is alkalmazható kutatási és tanulási célokra:

2.6.4. Oktatás

A Pygame és más játékfejlesztő eszközök egyre népszerűbbek az oktatásban. A játékok fejlesztése során a diákok számos készséget sajátíthatnak el, például:

- Programozás: A játékok fejlesztése során a diákok megtanulhatják, hogyan használják a programozási nyelveket és technikákat a játékok létrehozásához.
- Dizájn: A játékok fejlesztése során a diákok megtanulhatják, hogyan tervezzenek hatékony és élvezetes játékokat.
- Projektmenedzsment: A játékok fejlesztése során a diákok megtanulhatják, hogyan szervezzék és irányítsák a játékfejlesztési projekteket.

A játékfejlesztés egy élvezetes és motiváló módja a készségek elsajátításának. A diákok gyakran jobban motiváltak arra, hogy tanuljanak, ha a játékfejlesztésben vesznek részt.

Példák a Pygame oktatási alkalmazására:

- Iskolák és egyetemek: A Pygame-et egyre több iskolában és egyetemen használják a játékfejlesztés tanítására.
- Kódhatalók: A Pygame-et gyakran használják kódhatalók és játékfejlesztő versenyek során.
- Öntanulás: A Pygame egy ingyenes és nyílt forráskódú eszköz, amely alkalmas az öntanulásra.

2.6.5. Kognitív kutatás

A játékok kiváló eszközök lehetnek a kognitív folyamatok, például a tanulás, a memória és a döntéshozatal tanulmányozására. A játékfejlesztők gyakran használnak játékokat a kognitív folyamatok tesztelésére és modellezésére.

Példák a Pygame kognitív kutatási alkalmazására:

- Tanulás: A játékokat használják a tanulás hatékonyságának tanulmányozására.
- Memória: A játékokat használják a memória működésének tanulmányozására.
- Döntéshozatal: A játékokat használják a döntéshozatal folyamatának tanulmányozására.

2.6.6. Szórakoztató és kreatív projektek

A játékfejlesztés egy kiváló módja a szórakozásnak és a kreativitás kifejezésének. A játékfejlesztő eszközök lehetővé teszik az emberek számára, hogy saját játékaikat készítsék el, amelyeket megoszthatnak másokkal.

Példák a Pygame szórakoztató és kreatív projektekre:

- Egyéni projektek: Az emberek gyakran használják a Pygame-et egyéni játékok készítésére.
- Csapatprojektek: A Pygame-et gyakran használják csapatjátékok készítésére.
- Közösségi projektek: A Pygame-et gyakran használják közösségi játékok készítésére.

2.6.7. Prototípusok és koncepciók

A játékfejlesztés lehetővé teszi prototípusok készítését és koncepciók tesztelését a valóságban. A prototípusok és koncepciók tesztelése segíthet a fejlesztőknek abban, hogy javítsák a játékaikat, mielőtt azok nagyszabású fejlesztésbe kerülnek.

Példák a Pygame prototípusok és koncepciók tesztelésére:

- Új játékmechanikák: A Pygame-et gyakran használják új játékmechanikák tesztelésére.
- Új játéktílusok: A Pygame-et gyakran használják új játéktílusok tesztelésére.
- Új játékvilágok: A Pygame-et gyakran használják új játékvilágok tesztelésére.

2.6.8. Következtetés

A Pygame és más játékfejlesztő eszközök széles körű lehetőségeket kínálnak az oktatásban, a kognitív kutatásban, a szórakozásban és a kreativitásban. A megfelelő eszköz kiválasztásával bárki készíthet saját játékokat, függetlenül attól, hogy milyen tapasztalattal rendelkezik.

2.7. Összegzés

A fejezet tartalma témától függően változhat. Az alábbiakat attól függően különböző arányban tartalmazhatják.

- Irodalomkutatás. Amennyiben a dolgozat egy módszer kidolgozására, kifejlesztésére irányul, akkor itt lehet részletesen végignézni (módszertani vagy időrendi bontásban), hogy az eddigiekben milyen eredmények születtek a témakörben.
- Technológia. Mivel jellemzően kutatásról vagy szoftverfejlesztésről van szó, ezért annak a jellemző elemeit, technikai részleteit itt kell bemutatni. Ez tehát egy módszeres bevezetés ahhoz, hogy ha valaki nem jártas a témakörben, akkor tudja, hogy a dolgozat milyen aktuálisan elérhető eredményeket, eszközöket használt fel.
- Piackutatás. Bizonyos témáknál új termék vagy szolgáltatás kifejlesztése a cél. Ekkor érdemes annak alaposan utánanézni, hogy aktuálisan milyen eszközök érhetők el a piacon. Ez szoftverek esetében a hasonló alkalmazások bemutatását, táblázatos formában történő összehasonlítását jelentheti. Szerepelhetnek képek és észrevételek a viszonyításként bemutatott alkalmazásokhoz.
- Követelmény specifikáció. Külön szakaszban érdemes részletesen kitérni az elkészítendő alkalmazással kapcsolatos követelményekre. Ehhez tartozhatnak forgatókönyvek (*scenario*-k). A szemléletesség kedvéért lehet hozzájuk képernyőkép vázlatokat is készíteni, vagy a használati eseteket más módon szemléltetni.

Az olvasóról annyit feltételezhetünk, hogy programozásban valamilyen szinten jártos, és a matematikai alapfogalmakkal sem ebben a dolgozatban kell megismertetni. A speciális eszközök, programozási nyelvek, matematikai módszerek és jelölések persze jó, hogy ha említésre kerülnek, de nem kell nagyon belemenni a közismertnek tekinthető dolgokba.

3. fejezet

Tervezés

Itt kezdődik a dolgozat lényegi része, úgy érte, hogy a saját munka bemutatása. Jellemzően ebben szerepelni szoktak blokkdiagramok, a program struktúrájával foglalkozó leírások. Ehhez célszerű UML ábrákat (például osztály- és szekvenciadiagramokat) használni.

Amennyiben a dolgozat inkább kutatás jellegű, úgy itt lehet konkretizálni a kutatási módszertant, a kutatás tervezett lépéseit, az indoklást, hogy mit, miért és miért pont úgy érdemes csinálni, ahogyan az a későbbiekben majd részletezésre kerül.

Ebben a fejezetben az implementáció nem kell, hogy túl nagy szerepet kapjon. Ez még csak a tervezési fázis. (Nyilván ha olyan a téma, hogy magának az implementációnak a módjával foglalkozik, adott formális nyelvet mutat be, úgy a kódpéldákat már innen sem lehet kihagyni.)

3.1. Táblázatok

Táblázatokhoz a `table` környezetet ajánlott használni. Erre egy minta a 3.1. táblázat. A hivatkozáshoz az egyedi `label` értéke konvenció szerint `tab:` prefixszel kezdődik.

3.1. táblázat: Minta táblázat. A táblázat felirata a táblázat felett kell legyen!

a	b	c
1	2	3
4	5	6

3.2. Ábrák

Ábrákat a `figure` környezettel lehet használni. A használatára egy példa a 3.1. ábrában látható. Az `includegraphics` parancsba Az ábrák felirata az ábra alatt kell legyen. Az ábrák hivatkozásához használt nevet konvenció szerint `fig:-`el célszerű kezdeni.

3.3. További környezetek

A matematikai témájú dolgozatokban szükség lehet tételek és bizonyításaik megadására. Ehhez szintén vannak készen elérhető környezetek.



3.1. ábra: A Miskolci Egyetem címere.

3.1. definíció. Ez egy definíció

3.2. lemma. *Ez egy lemma*

3.3. tétel. *Ez egy tétel*

Bizonyítás. Ez egy bizonyítás

□

3.4. következmény. *Ez egy tétel*

3.5. megjegyzés. Ez egy megjegyzés

3.6. példa. Ez egy példa

4. fejezet

Megvalósítás

Ez a fejezet mutatja be a megvalósítás lépéseit. Itt lehet az esetlegesen előforduló technikai nehézségeket említeni. Be lehet már mutatni a program elkészült részeit.

Meg lehet mutatni az elkészített programkód érdekesebb részeit. (Az érdekesebb részek bemutatására kellene szorítkozni. Többségében a szöveges leírásnak kellene benne lennie. Abból lehet kiindulni, hogy a forráskód a dolgozathoz elérhető, azt nem kell magába a dolgozatba bemásolni, elegendő csak behivatkozni.)

A dolgozatban szereplő forráskódrészletekhez külön vannak programnyelvenként stílusok. Python esetében például a 4.1. programkódban látható egy formázott kódrészlet.

Programkód 4.1. Python példa

```
import sys

if __name__ == '__main__':
    pass
```

A stílusfájlok a **styles** jegyzékben találhatók. A stílusok között szerepel még C++, Java és Rust stílusfájl. Ezek használatához a **dolgozat.tex** fájl elején **usepackage** paranccsal hozzá kell adni a stílust, majd a stílusfájl nevével megegyező környezetet lehet használni. További példaként C++ forráskód esetében ez így szerepel.

```
#include <iostream>

class Sample : public Object
{
    // An empty class definition
}
```

Stílusfájlokból elegendő csak annyit meghagyni, amennyire a dolgozatban szükség van. Más, C szintaktikájú nyelvekhez (mint például a JavaScript és C#) a Java vagy C++ stílusfájlok átszerkesztésére van szükség. (Elegendő lehet csak a fájlnevet átírni, és a fájlban a környezet nevét.)

Nyers adatok, parancssori kimenetek megjelenítéséhez a **verbatim** környezetet lehet használni.

```
$ some commands with arguments
1 2 3 4 5
$ _
```

A kutatás jellegű témáknál ez a fejezet gyakorlatilag kimaradhat. Helyette inkább a fő vizsgálati módszerek, kutatási irányok kaphatnak külön-külön fejezeteket.

5. fejezet

Tesztelés

A fejezetben be kell mutatni, hogy az elkészült alkalmazás hogyan használható. (Az, hogy hogyan kell, hogy működjön, és hogy hogy lett elkészítve, az előző fejezetekben már megtörtént.)

Jellemzően az alábbi dolgok kerülhetnek ide.

- Tesztfuttatások. Le lehet írni a futási időket, memória és tárigényt.
- Felhasználói kézikönyv jellegű leírás. Kifejezetten a végfelhasználó szempontjából lehet azt bemutatni, hogy mit hogy lehet majd használni.
- Kutatás kapcsán ide főként táblázatok, görbék és egyéb részletes összesítések kerülhetnek.

6. fejezet

Összefoglalás

Hasonló szerepe van, mint a bevezetésnek. Itt már múltidőben lehet beszélni. A szerző saját meglátása szerint kell összegezni és értékelni a dolgozat fontosabb eredményeit. Meg lehet benne említeni, hogy mi az ami jobban, mi az ami kevésbé jobban sikerült a tervezettnél. El lehet benne mondani, hogy milyen további tervek, fejlesztési lehetőségek vannak még a témával kapcsolatban.

Források

- [1] Pedro Isaias és Tomayess Issa. *High Level Models and Methodologies for Information Systems*. 2014, 4–6. old.
- [2] Oracle Corporation. *Java Documentation*. 2023. URL: <https://docs.oracle.com/en/java/>.
- [3] The C++ Standards Committee. *C++ Standard Library Reference*. 2023. URL: <https://en.cppreference.com/w/>.
- [4] Unity Technologies. *Unity Documentation*. 2023. URL: <https://docs.unity3d.com/>.

CD Használati útmutató

Ennek a címe lehet például *A mellékelt CD tartalma* vagy *Adathordozó használati útmutató* is.

Ez jellemzően csak egy fél-egy oldalas leírás. Arra szolgál, hogy ha valaki kézhez kapja a szakdolgozathoz tartozó CD-t, akkor tudja, hogy mi hol van rajta. Jellemzően elég csak felsorolni, hogy milyen jegyzékek vannak, és azokban mi található. Az elkészített programok telepítéséhez, futtatásához tartozó instrukciók kerülhetnek ide.

A CD lemezre mindenképpen rá kell tenni

- a dolgozatot egy `dolgozat.pdf` fájl formájában,
- a LaTeX forráskódját a dolgozatnak,
- az elkészített programot, fontosabb futási eredményeket (például ha kép a kimenet),
- egy útmutatót a CD használatához (ami lehet ez a fejezet külön PDF-be vagy Markdown fájlként kimentve).