Szegedi Tudományegyetem

Informatikai Intézet

**SZAKDOLGOZAT**

Hallgató:

Török Dániel

2025

Szegedi Tudományegyetem

Informatikai Tanszékcsoport

2D souls-like platformer játék a Godot játék motorban

Szakdolgozat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Készítette: |  | Témavezető: |  |
|  | Török Dániel |  | Jász Judit dr. |  |
|  | programtervező informatikus szakos hallgató |  | adjunktus |  |

Szeged

2025

FELADATKIÍRÁS

A szakdolgozat célja egy 2D-s platformer souls-like videójáték fejlesztése, amelyben a játékos egy nyitott világban tud felfedezni, illetve ellenségekkel küzdeni, miközben fejleszti a karakterét, tárgyakat gyűjt és készít, képességeket old fel, amikkel később további részeket old fel a nyitott világból. Fontos, hogy a játék állást el lehessen menteni. továbbá betölteni.

Tartalmi összefoglaló

* ***A téma megnevezése***

2D souls-like platformer játék a Godot játék motorban

* ***A megadott feladat megfogalmazása***:

Egy olyan videójáték fejlesztése, amely lehetővé teszi a játékos számára, hogy fejlessze a karakterét, tárgyakat készítsen, ellenségekkel küzdjön, továbbá a játék tartalmazzon platformer elemeket (pl.: fal ugrás, dupla ugrás, platformok), a karakter halálakor a világban levő ellenfelek álljanak vissza (kivétel a főellenség(-ek)), egy rendszer, amely lehetővé teszi a játékos számára, hogy teleportálhasson előre meghatározott helyekre. Fontos funkciók fejlesztése példáúl a játék állás mentése, játék betöltése.

* ***A megoldási mód:***

Az játék fejlesztése Godot játék motor segítségével történt. A pixeles képek megrajzolásához a Pixeloroma program lett használva, audió fájlok a freesound.org webhelyről lettek beszerezve, verzió kezeléshez a github.com lett használva. A pálya, illetve minden más egyéb fájl a játék motoron belül lett elkészítve.

* ***Alkalmazott eszközök, módszerek:***

Pixelorama

Godot engine 4.2, 4.3, 4.4, 4.5

Visual Studio Code fejlesztői környezet

GDScript beépített programozási nyelv

* ***Elért eredmények:***

Az elkészült játék lehetővé teszi a játékos egy előre elkészített pálya felfedezését, amely több részre van bontva, közben ellenségekkel küzdjön meg beleértve a főllenség(ek)-et, csapdákat kerüljön ki, fejlessze karakterét, tárgyakat hozzon létre/ gyűjtsön össze, elmentse a játék állását, betöltse a játék állását, teleportálhasson előre meghatározott pozíciókra. Az ellenfelek visszaállításra kerülnek a játékos halálakor. Platformer elemek implementálásra kerültek.

* ***Kulcsszavak:***

Godot, GDScript, souls-like, souls, platformer, tárgyak, játék, videojáték, nyitott világ

Tartalomjegyzék

[FELADATKIÍRÁS 2](#_Toc211103009)

[Tartalmi összefoglaló 3](#_Toc211103010)

[Tartalomjegyzék 5](#_Toc211103011)

[BEVEZETÉS 6](#_Toc211103012)

[1. Technológiák bemutatása 6](#_Toc211103013)

[1.1. Programnyelvek, technológiák 6](#_Toc211103014)

[1.1.1. Godot 6](#_Toc211103015)

[1.1.2. CSS 7](#_Toc211103016)

[Irodalomjegyzék 8](#_Toc211103017)

BEVEZETÉS

A videojáték-fejlesztés az elmúlt évtizedekben az informatikai ipar egyik legdinamikusabban fejlődő területévé vált. A játékkészítés mára nem csupán szórakoztatóipari tevékenység, hanem komplex szoftverfejlesztési folyamat, amely ötvözi a programozást, a grafikai tervezést, a hangdizájnt és a mesterséges intelligenciát. A játékfejlesztő eszközök és motorok fejlődése lehetővé tette, hogy kis fejlesztőcsapatok vagy akár egyéni alkotók is professzionális minőségű játékokat hozzanak létre.

A szakdolgozat célja egy 2D, úgynevezett „souls-like” platformer játék megvalósítása a Godot Engine segítségével. A „souls-like” kifejezés olyan játékstílust jelöl, amelyet a Dark Souls sorozat inspirált: a jellemzői közé tartozik a magas nehézségi szint, a pontos időzítést igénylő harcrendszer, valamint a fokozatosan felfedezhető, összefüggő játéktér. A cél egy olyan prototípus létrehozása, amely hűen visszaadja ennek a műfajnak a hangulatát és mechanikai elemeit, ugyanakkor bemutatja a Godot Engine fejlesztési lehetőségeit és hatékonyságát.

A választás indoka kettős: egyrészt a Godot Engine nyílt forráskódú és szabadon felhasználható, így ideális környezet az oktatási és kísérleti célú fejlesztésekhez; másrészt a souls-like műfaj összetett játékmenete kiváló lehetőséget biztosít a különböző játékrendszerek — például harc, mesterséges intelligencia, fizika és felhasználói interfész — integrálásának bemutatására. A projekt során a hangsúly a játékmenet logikai felépítésén, a karaktervezérlésen, az ellenségek viselkedésén és a pályarendszer kialakításán lesz.

A dolgozat a fejlesztési folyamat elméleti és gyakorlati aspektusait egyaránt tárgyalja. Az első fejezet a Godot játékmotor működését, felépítését és programozási környezetét mutatja be. Ezt követően részletesen ismertetésre kerül a játék tervezése, az implementáció folyamata, valamint az alkalmazott technológiák. A záró fejezet a fejlesztés során szerzett tapasztalatokat, valamint a lehetséges továbbfejlesztési irányokat foglalja össze.

1. Technológiák bemutatása

1.1. Programnyelvek, technológiák

1.1.1. Godot

A **Godot Engine** egy modern, nyílt forráskódú játékmotor, amelyet Juan Linietsky és Ariel Manzur fejlesztett ki, és 2014-ben jelent meg az első stabil változata. A motor célja egy egységes, platformfüggetlen fejlesztői környezet biztosítása 2D és 3D játékok készítéséhez. A Godot alapja **C++ nyelven** íródott, és **GDScript** nevű, kifejezetten a játékfejlesztéshez tervezett, magas szintű szkriptnyelvet használ. A projektet a **Godot Foundation** és egy aktív közösség tartja fenn, amely folyamatos frissítésekkel és kiegészítésekkel biztosítja a rendszer fejlődését és hosszú távú támogatását.

A Godot a **jelenet- és csomópontalapú architektúrára** épül, amely moduláris felépítést és magas fokú újrafelhasználhatóságot tesz lehetővé. A motor beépített komponensei között megtalálható a **fizikai rendszer**, az **animációkezelés**, a **vizuális shaderrendszer**, valamint a **felhasználói felületek kialakítását** segítő eszköztár. A fejlesztők több szkriptnyelv közül választhatnak, így a GDScript mellett **C# és C++** is használható, ami megkönnyíti a különböző fejlesztési igényekhez való alkalmazkodást.

A Godot Engine előnyei közé tartozik, hogy **teljesen platformfüggetlen**, és támogatja a játékok exportálását **Windows, Linux, macOS, Android, iOS és Web (HTML5)** rendszerekre. A motor jól integrálható külső eszközökkel, például **Blenderrel, REST API-kkal vagy adatbázisokkal**, továbbá lehetőséget nyújt **VR- és AR-fejlesztésekre**, valamint **multiplayer rendszerek** kialakítására is.

A Godot választása azért indokolt, mert egy **ingyenesen elérhető, átlátható és professzionális fejlesztőkörnyezetet** kínál, amely magas fokú rugalmasságot biztosít. A hivatalos dokumentáció részletes útmutatókat és példákat tartalmaz, míg az aktív fejlesztői közösség fórumokon, blogokon és közösségi platformokon keresztül folyamatosan bővíti a tudásbázist. A motor nyílt forráskódjának köszönhetően a fejlesztők teljes kontrollt gyakorolhatnak a projekt felett, ami különösen fontos a kutatás-fejlesztési és oktatási célú felhasználás során.

1.1.2. CSS

A weboldal megjelenítéséhez a CSS (Cascading Style Sheets) stílusleíró nyelvet használtam, amely lehetővé teszi a tartalom és a forma szétválasztását. Különös figyelmet fordítottam a színek összehangjára, az elemek rendezett elhelyezésére, valamint az űrlapok és navigációs elemek esztétikus kialakítására. A stílusokat külső CSS fájlokba szerveztem, külön az egyes felületek számára, így a módosításokat egyszerűen, központilag lehetett elvégezni. A reszponzív megjelenítéssel biztosítottam, hogy a weboldal különböző eszközökön – mobilon, tableten, és asztali gépen – is megfelelően jelenjen meg. A modern CSS lehetőségei, mint a flexbox és az animációk, tovább javították a felhasználói élményt és a vizuális megjelenítést. [6, 9]

Irodalomjegyzék

[1] <https://docs.godotengine.org/en/stable/>

[2] <https://docs.godotengine.org/en/stable/classes/index.html#nodes>

[3] <https://docs.godotengine.org/en/stable/classes/index.html#resources>

[4] <https://orama-interactive.itch.io/pixelorama>

[5] <https://en.wikipedia.org/wiki/Godot_(game_engine)>

[6] <https://en.wikipedia.org/wiki/Soulslike>

[7] <https://gameprogrammingpatterns.com/>

[8] <https://gameprogrammingpatterns.com/singleton.html>

[9] <https://gameprogrammingpatterns.com/state.html>

[10] <https://freesound.org/>

[11] <https://firebase.google.com/docs/hosting/quickstart>

[12] <https://developer.mozilla.org/en-US/docs/Web/HTML>

[13] <https://www.w3schools.com/html/>

[14] <https://www.w3schools.com/angular/>