SpaceWalk The Game

Péter Bence, Vasánszki Milán, Székely Katalin, Tóth Balázs
A Foobar Reloaded csapata
Modern szoftverfejlesztési eszközök
GKNB_INTM006

October 2022

Contents

1 Alapinformációk

Ez a specifikáció egy szöveges, szerepjátékos űrkalandjátékhoz készült. A játék- ban mozoghatsz szobáról szobára, gyűjthetsz tárgyakat és új utakat nyithatsz meg a játék világának felfedezéséhez. Találkozhatsz és interakcióba léphetsz NPC-kkel. A program a felhasználótól vár egyszerűbb parancsokat, amelyeket később eseményként megjelenít. A történet 20 mélységű. Az alkalmazás a felhasználó által adott helytelen bemenetre hibaüzenettel válaszol.

2 Történetvázlat

Az év 2094. A mesterséges intelligencia teljes mértékben átvette az irányítást a Föld nevezetű bolygó felett 30 évvel ezelőtt. De nem úgy ahogy gondolod kedves PLAYER-NAME. 2064-ben kifejlesztették az Oázis AI-t, ami egyedül azt a cél szolgálja, hogy az emberek kényelemben éljenek. A tökéletes utópisztikus világ, ahol az emberek azért jönnek a Földre, hogy jól érezzék magukat, azzal foglalkozzanak, amivel csak akarnak, és akkor, amikor akarnak. Minden unalmas és monoton feladat elvégzését átvették a robotok és az OAI. Ebbe beletartozik a teljesen automatizált házvezetés, takarítás, bevásárlás, és még a főzés is. Nem kevés következménnyel járt ez az átállás. Beolvasztották az összes fegyvert, megszüntették az autókat, és csak is mindenki buszokkal, vagy kijelölt szállítási eszközzel közlekedhet. A 2045-ös jégkorszak után nagymértékben megcsappant a Föld lakossága. A tervezett 9,44 milliárd ember helyett kevesebb, mint 1 milliárd maradt. Itt kezdődik a napod. Ma van a huszonötödik születésnapod, és nagy terveid vannak a mai napra. Megnyitott az első űr bár és szálló, amit pár nappal ezelőtt állítottak Föld körüli pályára, és benne vagy a szerencsés 100 ember között, akik részt vehetnek a megnyitó rendezvényen. A SpaceWalk™ nevezetű bár óriási hírnévnek örvend. Több tíz éve beszélnek róla, hogy folyamatosan építik, részről részre szobáról szobára. Most ért véget az építkezés, és mindenki részt akar venni a megnyitón. Nagyon sok feltételnek meg kellett felelni, és ezen kívül egy pályázatot is be kellett adni, hogy miért te legyél a kiválasztott, aki részt vehet ezen a rendezvényen. Az, hogy izgatott vagy az egy enyhe alábecslés. Miket is ígér a SpaceWalk™? Kezdésnek egy űrsétával kezdődik az este, ami a legelső dolog, miután megérkezel az űrállomáshoz. Ezek után svédasztalos vacsora, korlátlan bár fogyasztás, diszkó, és az éjszaka végén kényelmes ágyak várják a vendégeket a hotel szobákban. Az este nagyon jól telt, viszont az éjszaka folyamán, rázkódásra ébredsz, és nem tudod mi történik. Vajon egy aszteroida találhatta el az állomást? Vagy valami meghibásodás állhat a háttérben? Lehet szabotázs? Elképzelhető, hogy ennél sokkal sötétebb történések állhatnak a háttérben? Derítsük ki!

3 Követelmények

Table 1: Követelmények

Eset	*Leiras*	*Prioritás*
1. Szobarendszer	A játék világának a felépítése, szobákból álljon.	Magas
2. Mozgás	A játékos tudjon mozogni szobárol szobára.	Magas
3. Környezeti interakció	A játékos tudjon interakcióba lépni környezetével. Szobák átkutatása, tárgyak megvizsgálása és felvétele, a játékban élő NPC-kel való interakció.	Közepes
4. Történet vonalának követése	Küldetéseken keresztül tudjon a játékos a sztorival haladni.	Magas

4 Használati esetek

Table 2: Használati esetek

Eset	*Leiras*	*Prioritás*
1. Move (player)	Mozgás szobáról szobára.	Magas
2. Search room (player)	Szoba átkutatása tárgyak és NPC-k után.	Magas
3. Pick up item (player)	Talált tárgyak felvétele, megviszgálása.	Magas
4. Interact with NPC (player)	Talált NPC-vel való kommunikáció.	Alacsony
5. Accept mission (player)	Talált NPC küldetésének felvétele.	Alacsony
6. Unlock Room (game)	Ha a játékosnál van kulcs a szobához, a játék automatikusan kinyitja azt.	Alacsony
7. Enter Room (game)	Ha a játékos kiválasztotta a mozgás egyik szobárol a másikra opciót és van nála kulcs, a játék átlépteti a kiválsztott szobába.	Magas
8. Finish Mission (game)	Ha a küldetés feltételei teljesülnek, akkor a játék automatikusan lezárja a küldetést.	Közepes

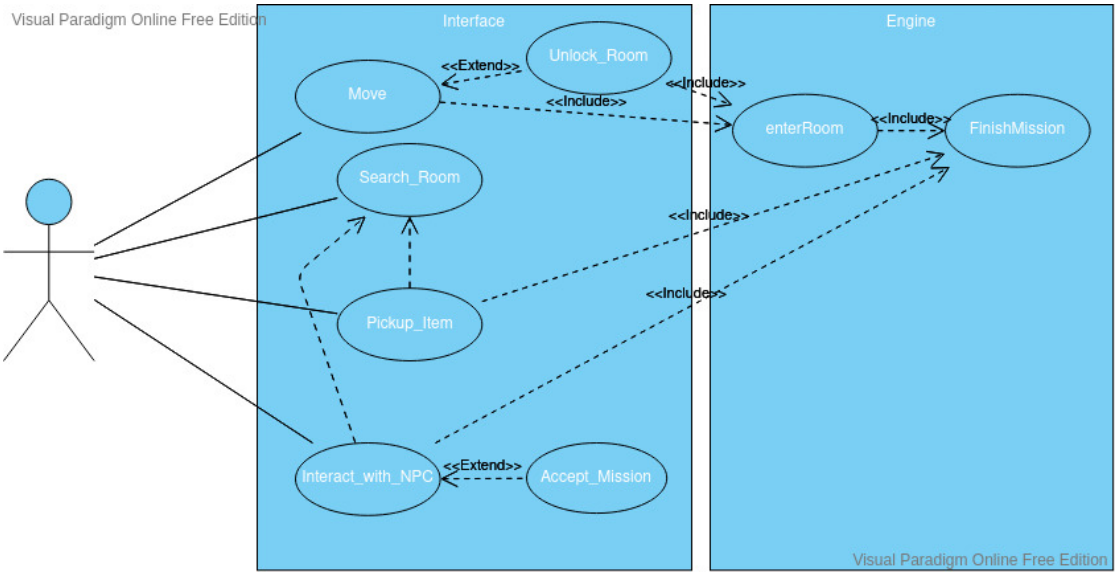

Figure 1: Functional Use Cases

5 Struktura

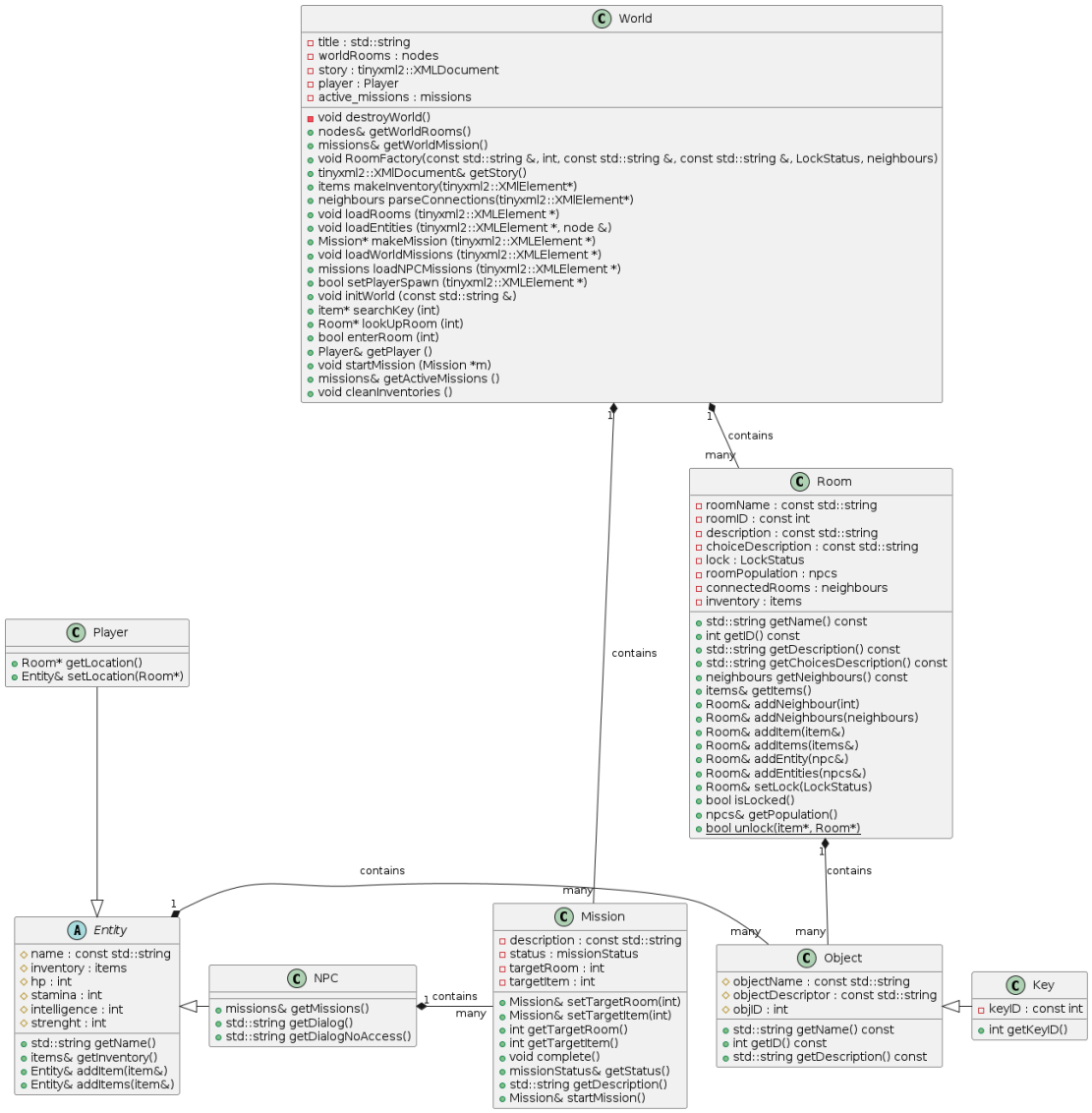

Figure 2: Class Architecture

5.1 Classok

class World A játék menetét kontrolláló class, a világ környezeti jellemzőit, szobáit, küldetéseit, sztoriát és magát a játékost kezeli. Ez a class fele az világ inicializálásáért, ezt az xml file parsolásával teszi meg.

class Room Játék világának fő építőeleme, a szobák kezelik a bennük lévő NPC-ket és tárgyakat.

class Entity A világ élőlényeit reprezentáló class, általában a szobákban tárolódnak el, kivéve a játékost reprezentáló Entity objectumot, amit a class World tartalmazza. Mindezek mellett a tárgyak inventoryba mozgazásához szükséges funkciókkal is rendelkezik.

class Object Tárgyak reprezentálására használt class.

class Key Tárgy class-ból származtatott osztály, ami a szobák nyitási mechanikáját valósítják meg.

class Mission A sztori haladásáért felelős osztály, a fő küldetéseket a játék inicializásakor már az aktív küldetések közé helyezzük, az opcionális küldetéseket a szobákban található NPC-től lehet felvenni. A küldetés célját tárolja, a státuszát kezeli és adja vissza és az automatikus befejezését kezeli.

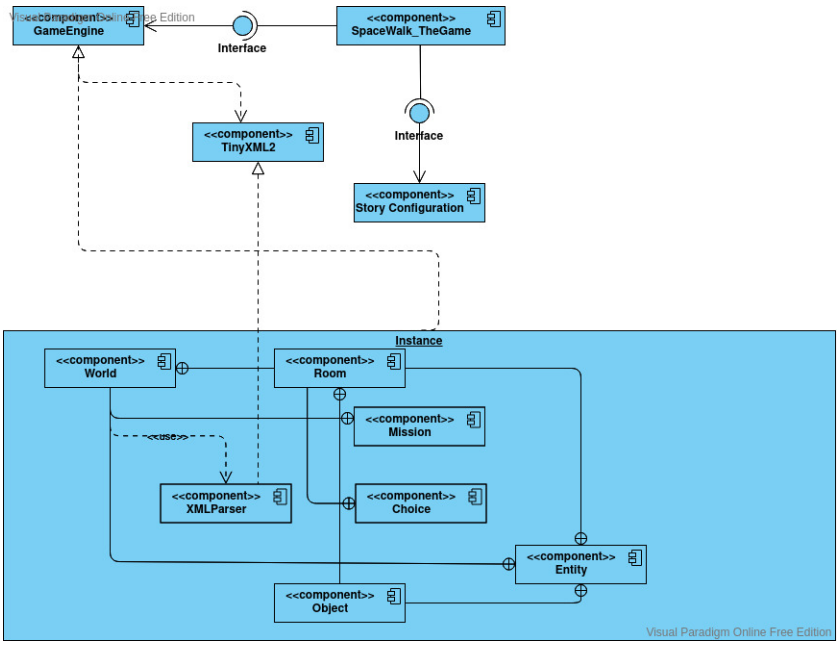

Figure 3: Component connection

Engine felépítése A játékvilág szobákból tevődik össze, amikben lehet tárgyakkal és NPC-kel interakcióba lépni. Vannak olyan szobák amiket csak kulcsok segítségével lehet kinyitni. A játékvilág felfedezése közben különböző küldetéseket kell végrehajtani, amik a sztori vonalát követik.

5.2 Világ inicializása

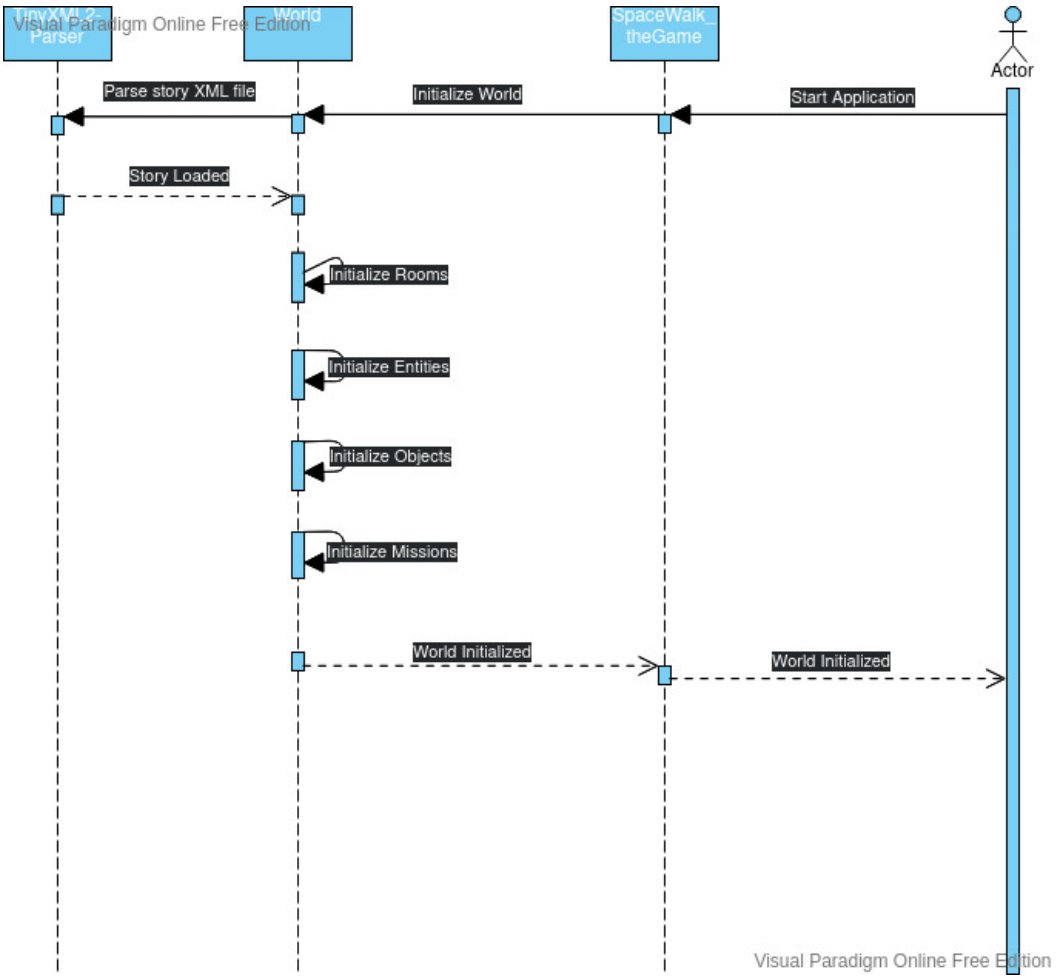

Figure 4: World Initialization squence

6 Viselkedes

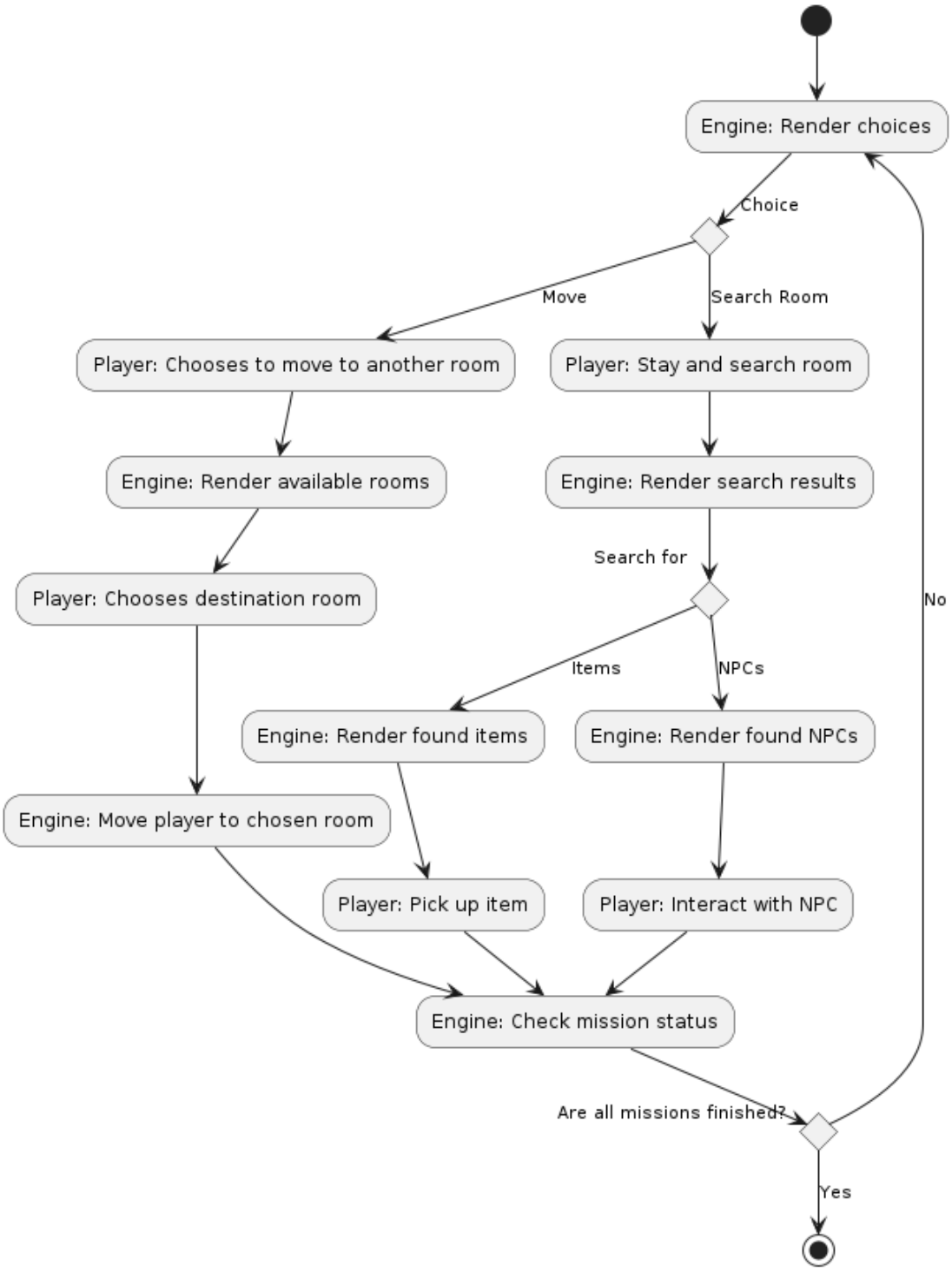

Figure 5: Gameloop Activity Diagram

6.1 Aktivitás diagram leírás

A játékosnak a játék 2 opciót ad amikor belép egy szobába: menjen tovább a következő szobába, kutassa át a szobát. Ha azt választja, hogy tovább menjen a következő szobába, akkor a játék kilistázza a lehetséges szobákat ahová lehet menni, amiután kiválasztotta a szobát ahová tovább akar menni, a játék átlépteti a játékos karakterét a szobába. A játék automatikusan tesztel, hogy teljesítve lettek-e küldetések. Ha a játékos a szoba átkutatását választja, akkor a játék kilistázza a talált tárgyakat és felfedezett NPC-ket, ezután választani kell, hogy a tárgyakat vizsgálja meg, vagy interakcióba lépjen-e az NPC-vel. Ha a tárgyak vizsgálatát választja akkor, kilistázza a játék a talált tárgyakat, amik közül ki lehet választani melyiket akarja felvenni. Ha az NCP-vel való interakciót választja, akkor megjelenik az NPC dialogja és elfogadhatjuk a felajánlott küldetést. A játék ezek után a tevékenységek után is tesztel, kész lett-e valamelyik küldetés.

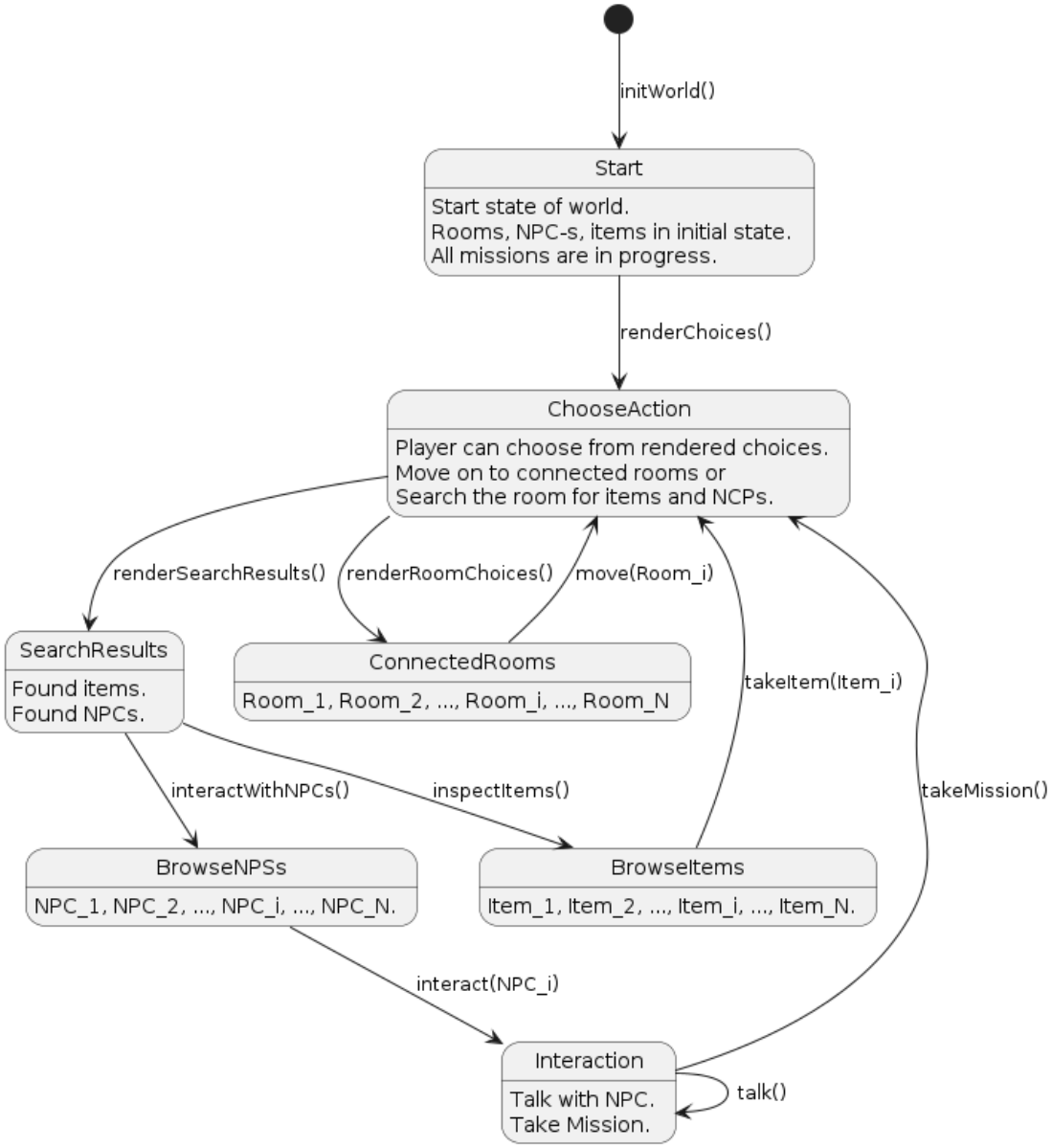

Figure 6: State Machine

6.2 Állapot diagram leírás

Amikor a játékos elindítja a játékot, a játék egy kezdő állapotban van, ahol a játék világ érintetlen, semelyik küldtés sincs befejezve. Ezután a játék egyből kilistázza a lehetséges opciókat, ez a tevékenység választási állapot. Ebből az állapotból átkerülhet a játék a csatolt szobák kilistázása állapotba és a szoba átkeresési állapotba. A szoba átkeresési állapotból 2 opció van, a talált tárgyak és a felfedezett NPC-k kilistázási állapot. Az összes állapot során valamilyen döntést kell hoznia a játékosnak, milyen cslekvést választ következőnek.

6.3 Játék menete

A játék loop jellegűen ajánlja fel a lehetséges tevékenységeket a játékos számára, a tevékenységek ismételten való elvégzésével lehet a küldetések céljait elérni és a sztori vonalán haladni. Lehetséges tevékenységek lehetnek: másik szobába mozgás, szoba átkutatása, tárgy megvizsgálása és felvétele, NPC-vel való interakció, küldetés elfogadása. A tevékenység sorban és egymásból következhetnek. Pl. ha azt választjuk hogy mozogjunk át egy másik szobába, akkor a játék a következő szobában újra felsorolja a lehetséges tevékenységeket, de ha azt választjuk, hogy kutassuk át a szobát, akkor ebből következhenek a tárgyak megvizsgálása vagy NPC-vel való kommunikáció tevékenységek. A küldetések státuszát a játék automatikusan frissíti a játékos tevékenységeinek és döntéseinek alapján.

7 Történet tárolási formátuma

A történetet xml fájlformátumban tároljuk és a TinyXML2 könyvtár segítségével olvassuk be. Az xml fájlformátum lehetővé teszi, hogy a játékvilágot és hozzá tartozó történelem részleteket modulárisan építhessük fel, így szabadon bővíthető lesz, vagy akár egy teljesen más történetet is betölthetünk.

7.1 XML file felépítése

7.1.1 <world>

A <world> tag-ből szigorúan csak egyet szabad használni, mivel ez fogja tartalmazni a játékvilág összes elemét.

7.1.2 <title>

<title> tag-ből megint csak egyet szabad használni, ez tartalmazza a játék címét. Ez a tag a <world> tagnek egy gyereke.

7.1.3 <room>

A <room> tag segítségével tölthetjük meg szobákkal a világot, ebből a tagből akármennyit lehet használi, de szigorúan a <world> tag-en belül.

7.1.4 <name>

A <name> tag a <room> nevét adja meg.

7.1.5 <description>

A <description> tag-el lehet a szobának a leírását, szobához tartozó storyt megadni.

7.1.6 <id>

Az <id> tag egyértelműen a szoba azonosítója, ami segít az egyszerű azonosításban, a kulcsokkal való összekötéssel és a szomszéd szobák megadásában.

7.1.7 <inventory>

A szobában megtalálható tárgyakat foglalja magába az <inventory> tag, amiből szobánként egy van.

7.1.8 <object>

Az <object> tageket az <inventory> tageken belülre kell helyezni, mivel ezzel tudjuk konkrétan megadni egy szobában található tárgynak a tulajdonságait.

7.1.9 <name>

Egy tárgynak is van megnevezése, ezért az <object>-en belül is kell használni a name taget.

7.1.10 <description>

A tárgyak kinézetét, felhasználási módját a leírás - description - tag-el lehet megadni.

7.1.11 <id>

A tárgyakat nehéz lenne csak név és leírás alapján beazonosítani programozási szemszögből így az id tagben megadott id-vel könnyen be lehet azonosítani.

7.1.12 <entity>

A szobákban létező NPC-ket az entity tag-el lehet megadni. Az entity-knek is van nevük, a név megadásához az eddig már többször elhangzott <name> tag-el lehet megadni. Az entity-knek is van <inventory>-juk, amiben ugyan úgy lehet megadni a tárgyakat - <object> -, mint a szobáknak.

7.1.13 <missions>

Ez az új tag a <world>-tag gyereke, a storyhoz tartózó küldetések megadására szolgál.

7.1.14 <mission>

A mission tag-en belül meg kell adni a küldetés leírását <description>-nel adható meg a küldetés leírása. A küldtés célját a <targetRoom> és a <targetItem> tag-el adhatjuk meg, ezeknek a tageknek a szoba vagy tárgy azonosítóját kell tartalmazniuk.

7.2 Példa az xml story fájl elkészítéséhez

```
<?xml version="1.0" encoding="UTF-8"?>
<world>
    <title>SpaceWalk</title>
    <spawn>
        1
    </spawn>
    <room>
        <name>TestRoom1</name>
        <description>This is the first test room</description>
        <choice>Choice Description of room 1</choice>
        <id>1</id>
        <lock>0</lock>
        <inventory>
            <object>
                <name>TestObject1</name>
                <description>This is test object 1</description>
                <id>1</id>
            </object>
        </inventory>
        <connections>
            <id>2</id>
            <id>3</id>
        </connections>
        <entity>
            <name>TestEntity1</name>
            <dialog>This is a test dialog.</dialog>
            <inventory>
                <key>
                    <name>TestKey1</name>
                    <id>7</id>
                    <description>This is test key 1</description>
                    <keyID>2</keyID>
                </key>
            </inventory>
            <missions>
                <mission>
```

```
                    <description>
                        This is a NPC test mission.
                    </description>
                    <targetItem>
                        1
                    </targetItem>
                </mission>
            </missions>
        </entity>
    </room>
    <room>
        <name>TestRoom2</name>
        <description>This is the second test room</description>
        <choice>Choice Description of room 2</choice>
        <id>2</id>
        <lock>1</lock>
        <connections>
            <id>1</id>
            <id>3</id>
        </connections>
    </room>
    <room>
        <name>TestRoom3</name>
        <description>This is the third test room</description>
        <choice>Choice Description of room 3</choice>
        <id>3</id>
        <lock>0</lock>
        <inventory>
            <object>
                <name>TestObject3</name>
                <id>3</id>
                <description>This is test object 3</description>
            </object>
        </inventory>
        <connections>
            <id>1</id>
            <id>2</id>
        </connections>
    </room>
    <missions>
        <mission>
            <description>
                This is a test mission.
            </description>
```

```
            <targetRoom>
                1
            </targetRoom>
            <targetItem>
                2
            </targetItem>
        </mission>
    </missions>
</world>
```

8 Tesztelés

8.1 GoogleTest

A GoogleTest könyvtár felhasználásával írtunk teszteket amik a tests mappában találhatók. A tesztek magukat a class-okat, a class-ok egymással való komunikációját és a use case-ket fedik le.

8.2 Példák

8.2.1 Room class konstruktorának tesztje

```
TEST_F(RoomTest, constructor_test) {
    node Room1(
        new Room("Room1", 1,
                 "This room is created to test the constructor."
        )
    );
    EXPECT_EQ("Room1", Room1->getName());
    EXPECT_EQ(1, Room1->getID());
    EXPECT_EQ("This room is created to test the constructor.",
              Room1->getDescription()
    );
}
```

8.2.2 Room class item-ekkel való operációja

```
TEST_F(RoomTest, test_Item_operation_methods) {
    // testItems should have 10 items now
    ASSERT_EQ(10, testItems.size());
    testRooms[0]->addItem(testItems[0]).addItem(testItems[1]);
    // 2 items from testItems were added to testRoom[0]'s
    // inventory, so now testItems should have 8 items
    ASSERT_EQ(2, testRooms[0]->getItems().size());
    // This unique_ptr should contain a nullptr,
    // because addItem moved it to the Room's inventory
    EXPECT_EQ(nullptr, testItems[0].get());
    EXPECT_EQ(nullptr, testItems[1].get()); // Same as above
    EXPECT_STREQ(std::string("Item No. 0").c_str(),
                 testRooms[0]->getItems()[0]->getName().c_str());
    EXPECT_STREQ(std::string("Item No. 1").c_str(),
                 testRooms[0]->getItems()[1]->getName().c_str());
    }
```

8.3 Github Actions

A githubnak egy nagyon hasznos funkciója az Actions amivel automatizált Workflow-okat lehet lefuttatni. Mi ennek a funkciónak a segítségével futtatjuk le a megírt teszteket. Ez egy remek újítás, mivel ez megakadályozza a hibás kódok master branchre való feltöltését. Ezt a funkciót a repository beállításainál a Branches fülben lehet megtatlálni.

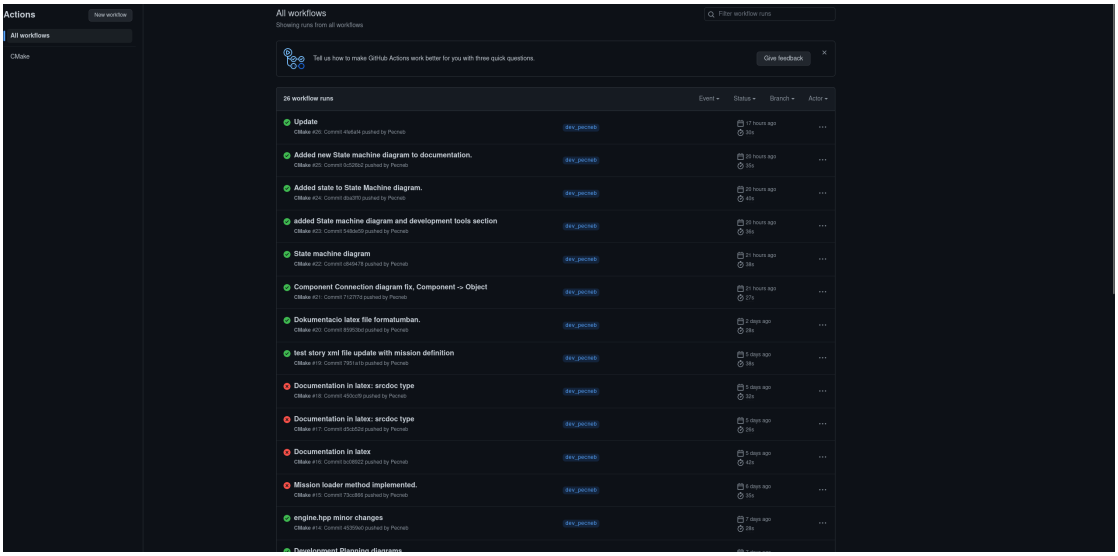

Figure 7: Workflows

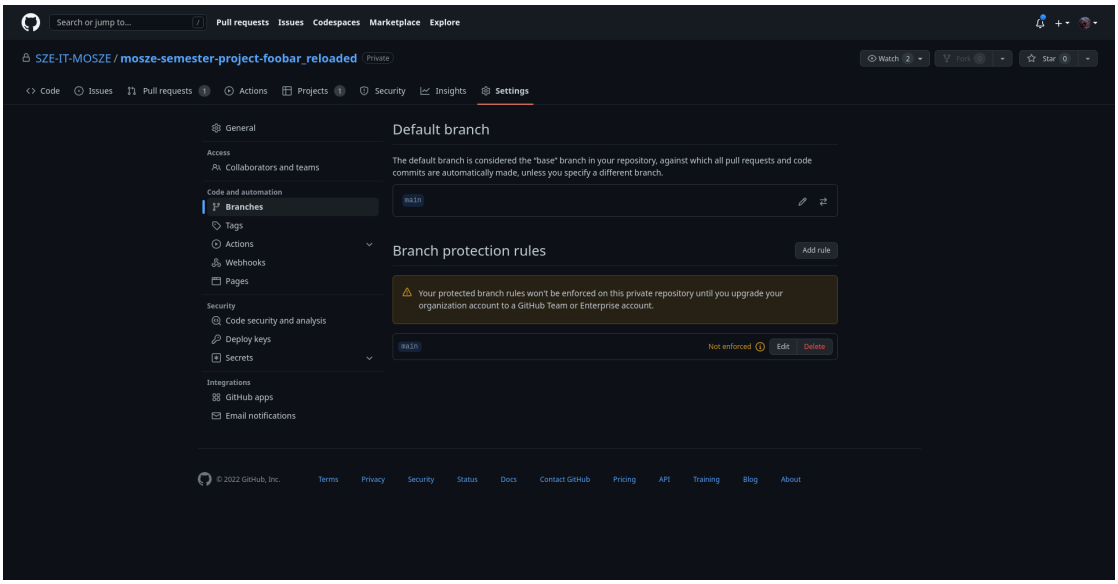

Figure 8: BranchProtection

9 Fejlesztés menete

9.1 Szoftverfejlesztési módszertanok

A játék fejlesztéséhez scrum és kanban módszertant törekszünk követni. Ehhez a github "Projektek" felülete nyújt segítséget, ahol egy kanban táblán vesszük fel a tevékenységeket amik alkotják a backlogot.

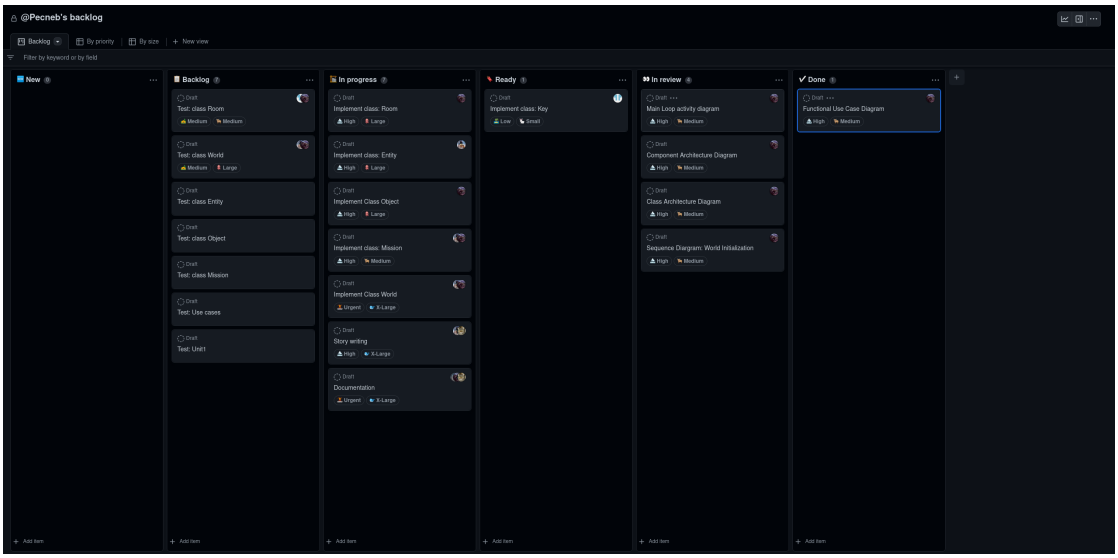

Figure 9: Kanban tábla

9.2 Dokumentáció

9.2.1 Doxygen

A Doxygen nagy segítségnek bizonyult a fejlesztés során, a kódunk megfelelő kommentelésével hasznos dokumentációt készít számunka, hogy megértsük és fel tudjuk használni a másik kódját.

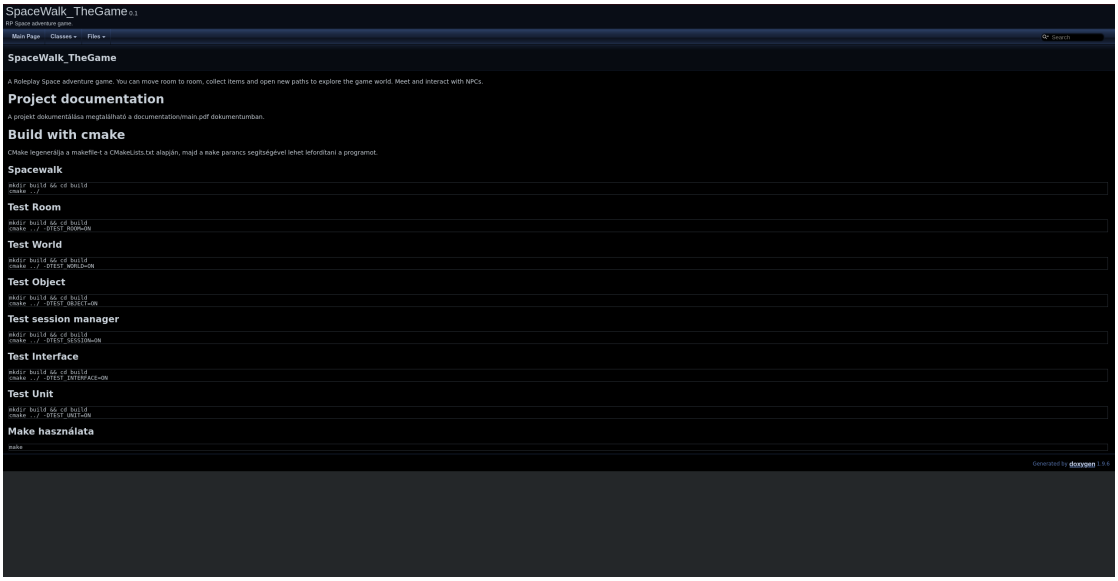

Figure 10: Doxygen

10 Fejlesztési eszközök

- Verziókezelés: Git, Szolgáltató: Github
- Fejlesztési környezet: VSCode
- Fejlesztési környezet kiegészítők: CMake Tools, LaTeX LaTeX Utilites, LaTeX Workshop, Makefile Tools, PlantUML, PlantUMLViewer, Vim, XML Tools
- Diagramok elkészítéséhez használt eszközök: VisualParadigm, PlantUML
- Operációs rendszerek: Pop! OS (Linux), Windows 10, Windows 11
- Dokumentáláshoz használt eszközök: Markdown, LaTeX, Doxygen
- Külső könyvtárak: GoogleTest, TinyXML2
- Debugger eszközök: GDB, Valgrind
- Szövegszerkesztő programok: Word
- Kommunikációs platformok: Discord, Messenger

11 Glossary

- Űrkalandjáték: Olyan játék, melynek fő helyszíne a világűr.
- Világűr: Az egyes égitestek között jelenlévő légüres tér (vákuum).
- Mesterséges intelligencia (Artificial Intelligence - AI): Egy gép, program vagy mesterségesen létrehozott tudat által megnyilvánuló intelligencia.
- NPC (Non-Playable Character): Olyan karaktert jelöl a játékban, melyet a játékos nem tud irányítani.
- Utópia: Olyan, általában jövőbeli társadalom, mely tökéletes vagy közel tökéletes. (Pl.: Platón - Állam)
- Disztópia: Az utópia ellentéte, azaz olyan társadalom, mely szétesőben van vagy az összeomlás felé tart, vagy már el is érte azt. Gyakori témái a háború, az erőszak, az elnyomás és a diktatúra. (Pl.: George Orwell - 1984)
- Sci-Fi (Science-Fiction): Olyan alkotás, mely létező vagy kitalált tudományoknak a társadalomra gyakorolt hatását mutatja be.
- Jégkorszak: Olyan időszak, amikor az adott égitesten állandó sarki jégtakaró van jelen.
- Űrséta: Olyan speciális tevékenység, melynek során az űrhajós teste, vagy testének egy része az űrhajón kívülre kerül.
- Aszteroida: 1 méterestől akár 1000 kilométeres átmérőig terjedő, szabálytalan alakú, szilárd anyagú égitest, amely nem rendelkezik atmoszférával, és csillag körül kering.
- Szabotázs: Romboló tevékenység, melynek célja valamilyen folyamat késleltetése.
- Speifikáció: Részletes ismertető, a tulajdonságok felsorolása és leírása.
- Monoton: Egyhangú, változatosság nélküli tevékenység.
- Űrállomás: Életfenntartó rendszerrel berendezett bolygó vagy hold körüli pályán keringő űreszköz.
- Svédasztal: Ételek felszolgálási módja, mely során egy külön asztalon mutatják be az ételeket és a vendégek igényeiknek megfelelően válogathatnak az asztalról.
- Diszkó: Szórakozhely, népszerű zenékkel és tánccal.
- Történetvázlat: Rövid, nagy vonalakban történő leírás a történetről és eseményekről.
- Használati eset / use case: Leírja a modellezendő redszer és a környezete kapcsolatát. Célja a követelményrögzítés.

- Tesztelés: Olyan folyamat, amely a szoftverfejlesztési életciklushoz kapcsolódik és különböző tevékenységekből tevődik össze. Célja hogy megtalálja a hibákat illetve megnézze, hogy a szotvertermék teljesíti-e a meghatározott követelményeket.
- Módszertan: Alkalmazási mód, különböző eljárások és módszerek összessége amelyeket tudatosan felhasználunk a cél elérése érdekében.
- XML: Extensible Markup Language azaz kiterjeszthető jelölőnyelv. Lehetővé teszi egyéni információs formátumok létrehozását és használatát.
- TinyXML2: C++ XML-elemző, amely könnyen integrálható más programokba mert egyszerű és hatékony.
- GoogleTest: Egy tesztelési keretrendszer, amely platformfüggetlen, nem csak az unit teszteket támogatja, gyors és könnyen tanulható.
- Workflow: Munkafolyamat, egy kitűzött cél elérése érdekében végrehajtott tevékenységek ésszerű sorrendje.
- Scrum: Agilis módszertan szigorú szabályokkal és tág felelősségi körökkel.
- Kanban: Emberek csoportos munkájának menedzselésére és fejlesztésére kialakított ütemezési rendszer.
- Git: Gráfalapú nyílt forráskódú verziókezelő rendszer.
- Github: A Microsoft által üzemeltetett felhőalapú git protokollt használó verziókezelő tárhely.
- VSCode: Ingyenes, nyílt forráskódú kódszerkesztő a modern webes és felhőalapú alkalmazások építéséhez és hibakereséséhez.
- Word: A Microsoft által készített dokumetumszerkesztő program.
- Discord: Ingyenes VoIP alkalmazás, amely lehetőséget nyújt a felhasználók közötti kommunikációra.
- Messenger: Üzenetküldő platform, amit a Facebook fejleszett ki.