

SpaceWalk The Game

Péter Bence, Vasánszki Milán, Székely Katalin, Tóth Balázs
A Foobar Reloaded csapata

October 2022

Contents

1	Alapinformációk	3
2	Történetvázlat	3
3	Használati esetek	4
4	Struktura	5
5	Viselkedés	6
6	Történet tárolási formátuma	8
6.1	XML file felépítése	8
6.1.1	<world>	8
6.1.2	<title>	8
6.1.3	<room>	8
6.1.4	<name>	8
6.1.5	<description>	8
6.1.6	<id>	8
6.1.7	<inventory>	8
6.1.8	<object>	8
6.1.9	<name>	9
6.1.10	<description>	9
6.1.11	<id>	9
6.1.12	<entity>	9
6.1.13	<missions>	9
6.1.14	<mission>	9
6.2	Példa az xml story fájl elkészítéséhez	9
7	Tesztelés	12
7.1	GoogleTest	12
7.2	Példák	12
7.2.1	Room class konstruktorának tesztje	12
7.2.2	Room class item-ekkel való operációja	12
7.3	Github Actions	13
8	Fejlesztés menete	14
8.1	Szoftverfejlesztési módszertanok	14
8.2	Dokumentáció	14
8.2.1	Doxygen	14
9	Fejlesztési eszközök	15
10	Glossary	15

1 Alapinformációk

Ez a specifikáció egy szöveges, szerepjátékos úrkalandjátékhoz készült. A játékban mozoghatsz szobáról szobára, gyűjthetsz tárgyakat és új utakat nyithatsz meg a játék világának felfedezéséhez. Találkozhatasz 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 automatikus 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™? Kezdsnek 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 Használati esetek

Table 1: Használati esetek

<i>Eset</i>	<i>Leiras</i>
Move	Mozgas szobarol szobara
Search room	Kutasd at a szobat, itemek es NPC-k utan
Pick up item	Vegyed fel a talalt targyat
Interact with NPC	Beszelj a talalt NPC-vel
Accept mission	Fogadd el az NPC kuldeteset

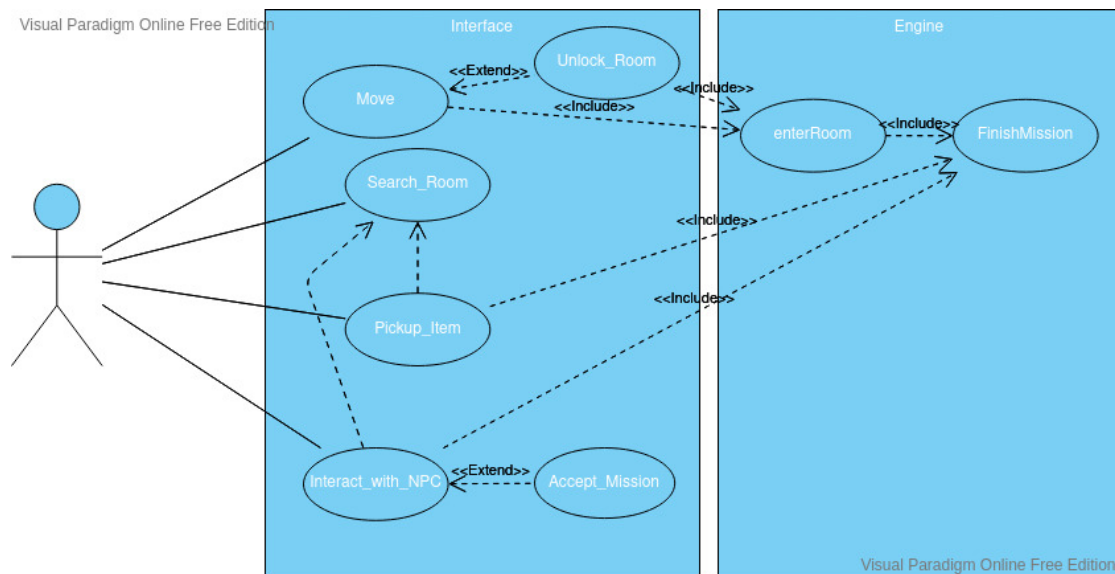


Figure 1: Functional Use Cases

4 Struktura

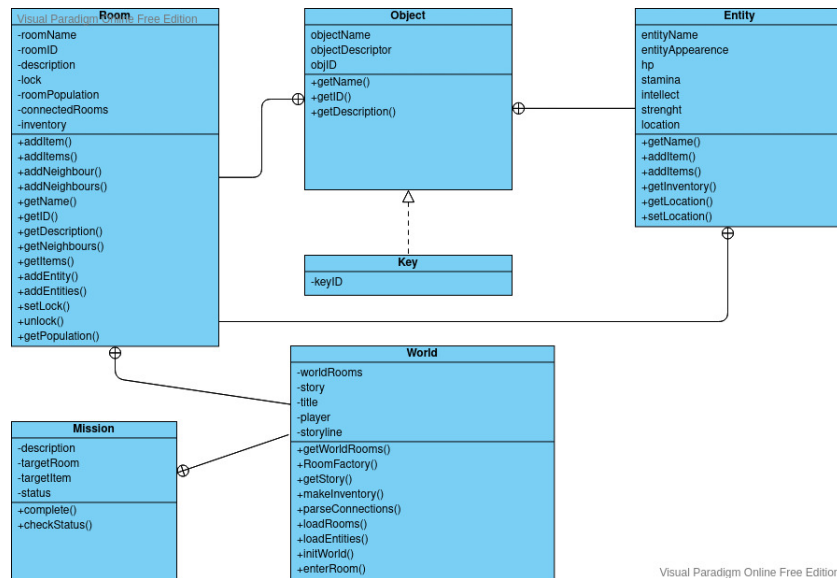


Figure 2: Class Architecture

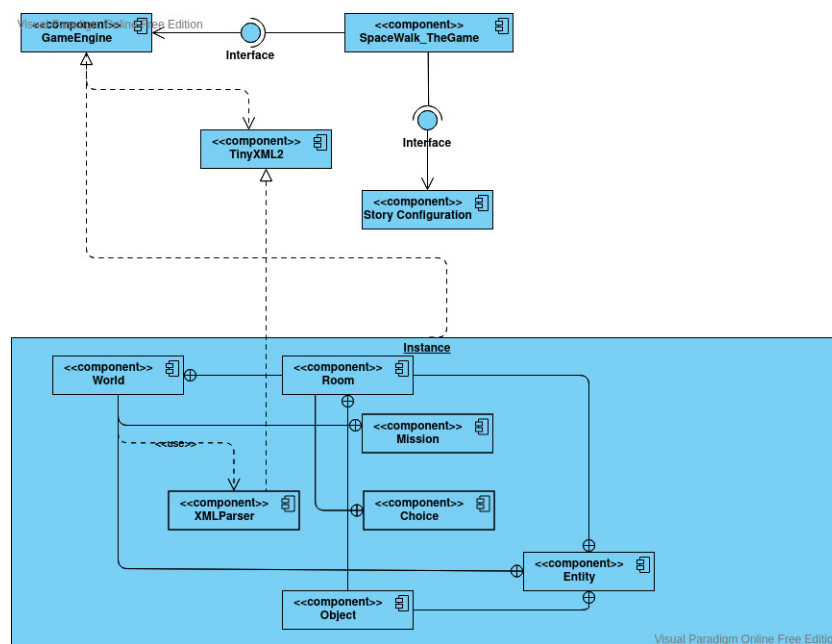


Figure 3: Component connection

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 Viselkedés

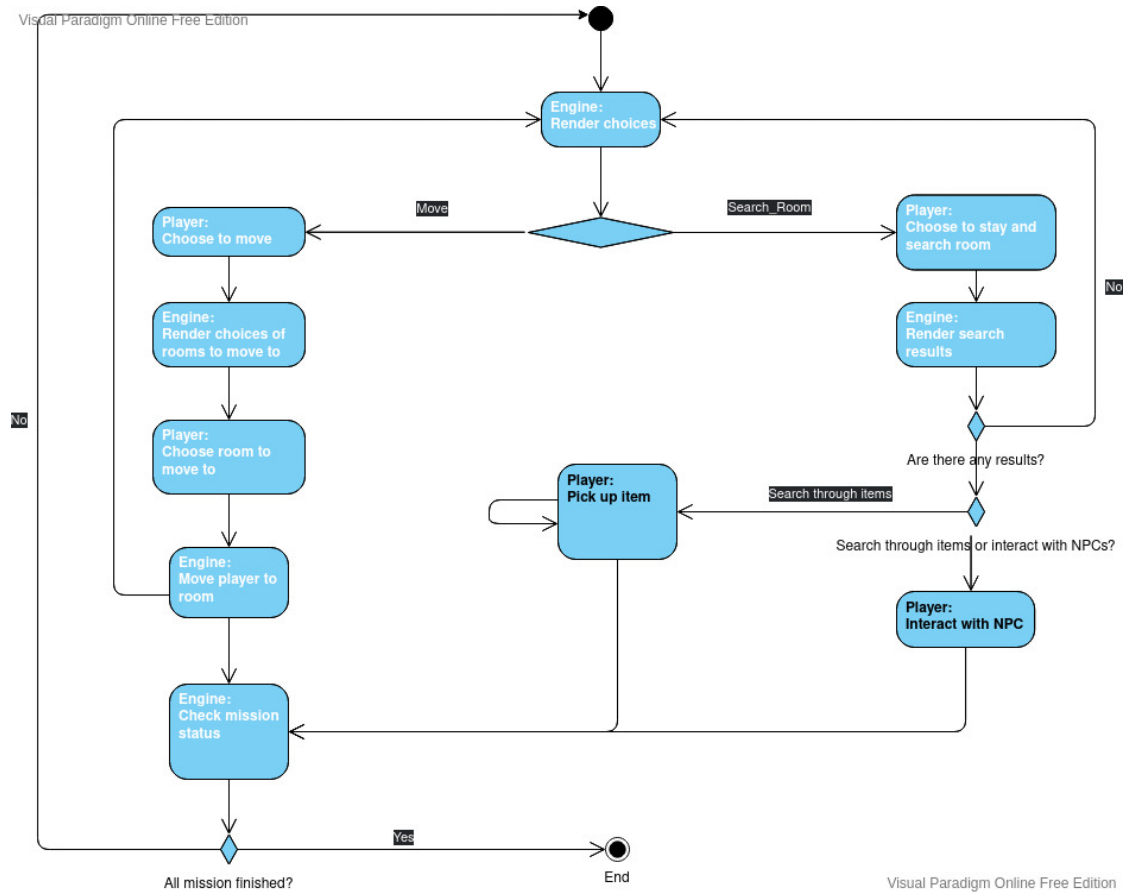


Figure 4: Gameloop Activity Diagram

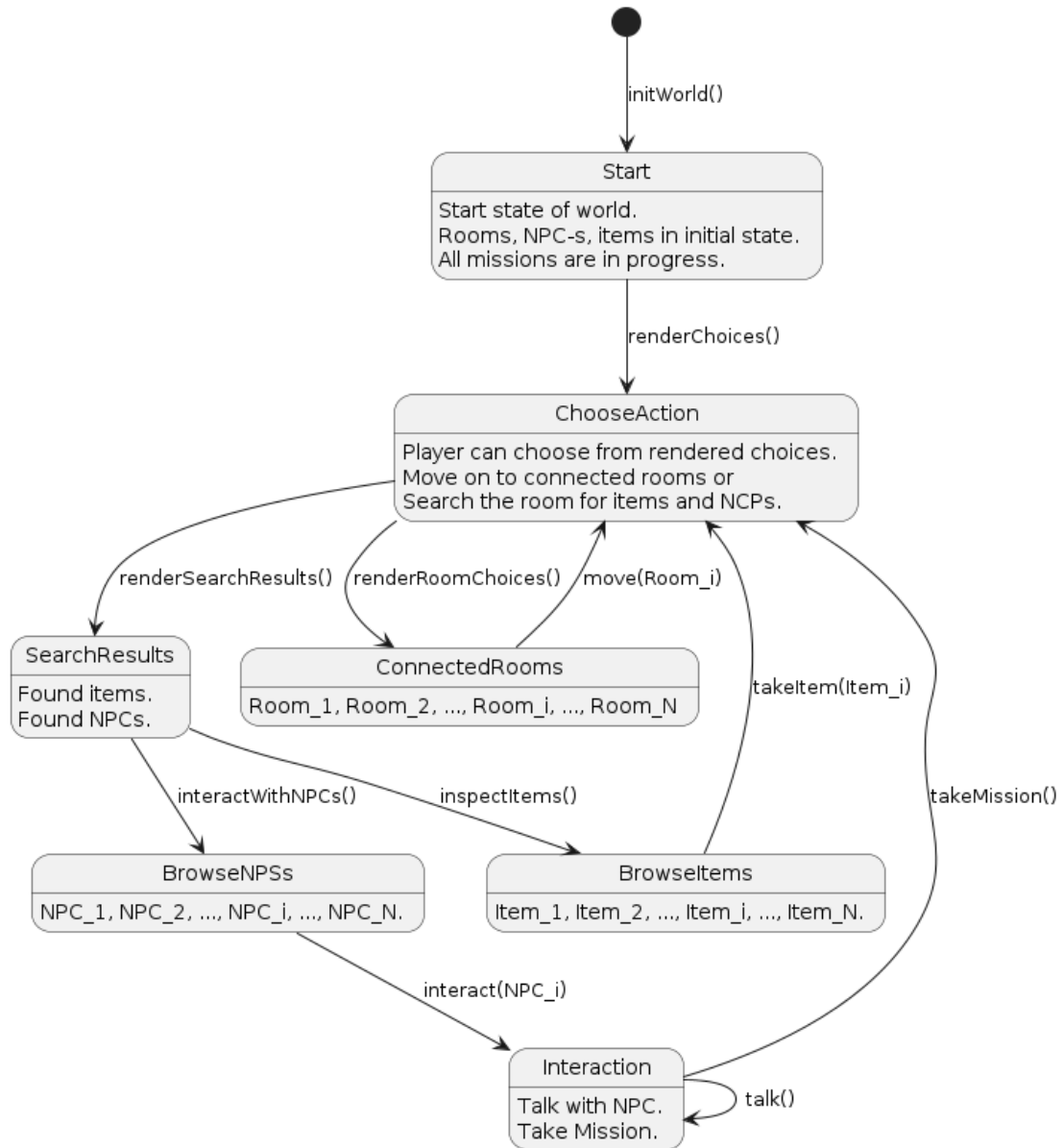


Figure 5: State Machine

6 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.

6.1 XML file felépítése

6.1.1 <world>

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

6.1.2 <title>

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

6.1.3 <room>

A <room> tag segítségével tölthetjük meg szobákkal a világot, ebből a tagból akármenyit lehet használni, de szigorúan a <world> tag-en belül.

6.1.4 <name>

A <name> tag a <room> nevét adja meg.

6.1.5 <description>

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

6.1.6 <id>

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

6.1.7 <inventory>

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

6.1.8 <object>

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

6.1.9 <name>

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

6.1.10 <description>

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

6.1.11 <id>

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

6.1.12 <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.

6.1.13 <missions>

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

6.1.14 <mission>

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

6.2 Példa az xml story fájl elkészítéséhez

```
<?xml version="1.0" encoding="UTF-8"?>
<world>
  <title>SpaceWalk</title>
  <room>
    <name>TestRoom1</name>
    <description>This is the first test room</description>
    <id>1</id>
    <inventory>
      <object>
        <name>TestObject1</name>
        <description>This is test object 1</description>
        <id>1</id>
      </object>
    </inventory>
  </room>
</world>
```

```

    </inventory>
    <connections>
        <id>2</id>
        <id>3</id>
    </connections>
    <entity>
        <name>TestEntity1</name>
        <inventory>
            <object>
                <name>TestObject4</name>
                <description>This is an object in TestEntity1's inv.</description>
                <id>4</id>
            </object>
        </inventory>
    </entity>
</room>
<room>
    <name>TestRoom2</name>
    <description>This is the second test room</description>
    <id>2</id>
    <inventory>
        <object>
            <name>TestObject2</name>
            <id>2</id>
            <description>This is test object 2</description>
        </object>
    </inventory>
    <connections>
        <id>1</id>
        <id>3</id>
    </connections>
</room>
<room>
    <name>TestRoom3</name>
    <description>This is the third test room</description>
    <id>3</id>
    <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>
    </mission>
</missions>
</world>

```

7 Tesztelés

7.1 GoogleTest

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

7.2 Példák

7.2.1 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()
    );
}
```

7.2.2 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_EQ("Item No. 0",
        testRooms[0]->getItems()[0]->getName()
    );
    EXPECT_EQ("Item No. 1",
        testRooms[0]->getItems()[1]->getName()
    );
};
```

7.3 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.

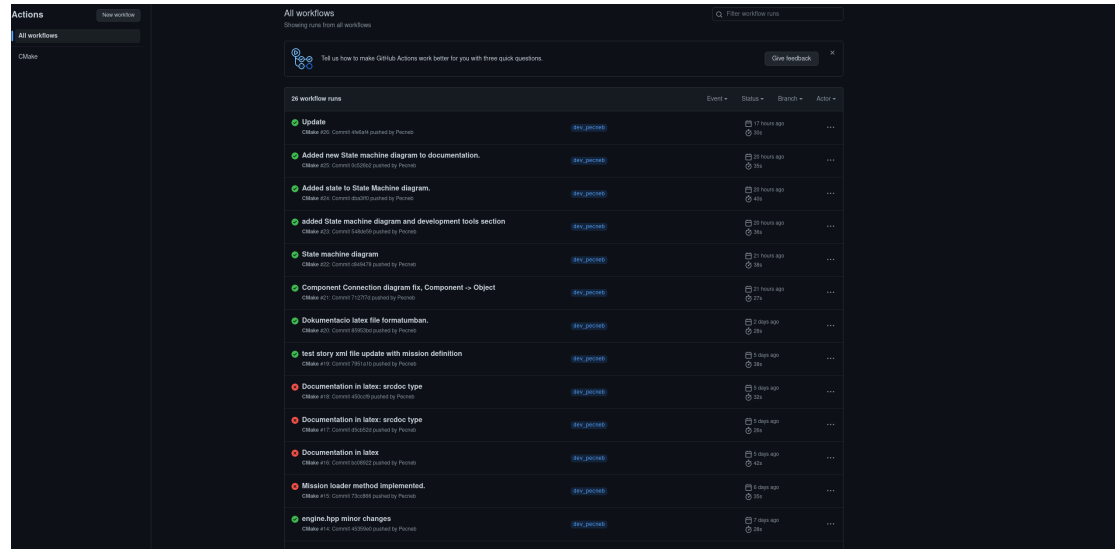


Figure 6: Workflows

8 Fejlesztés menete

8.1 Szoftverfejlesztési módszertanok

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

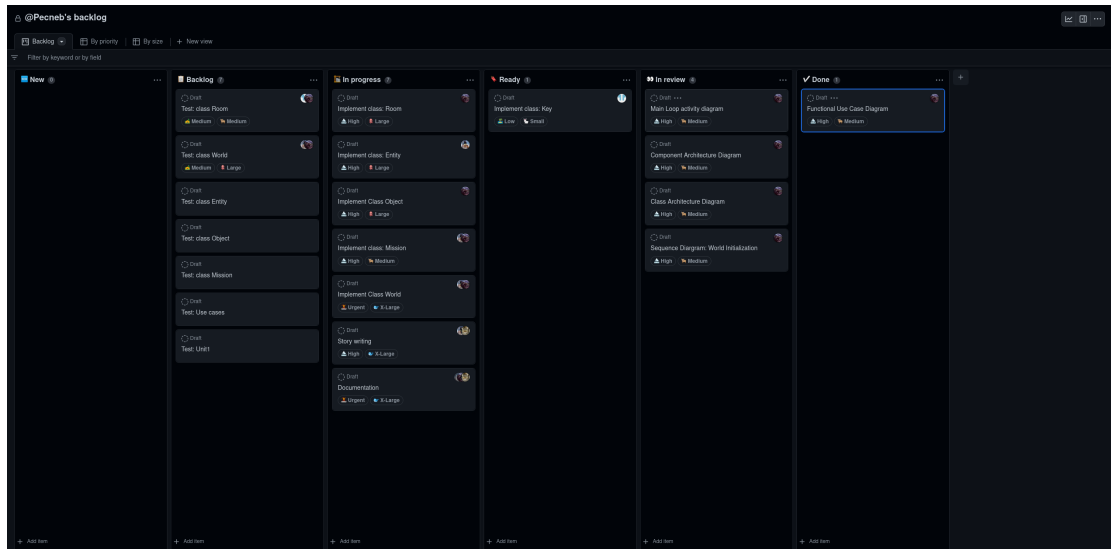


Figure 7: Kanban tábla

8.2 Dokumentáció

8.2.1 Doxygen

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

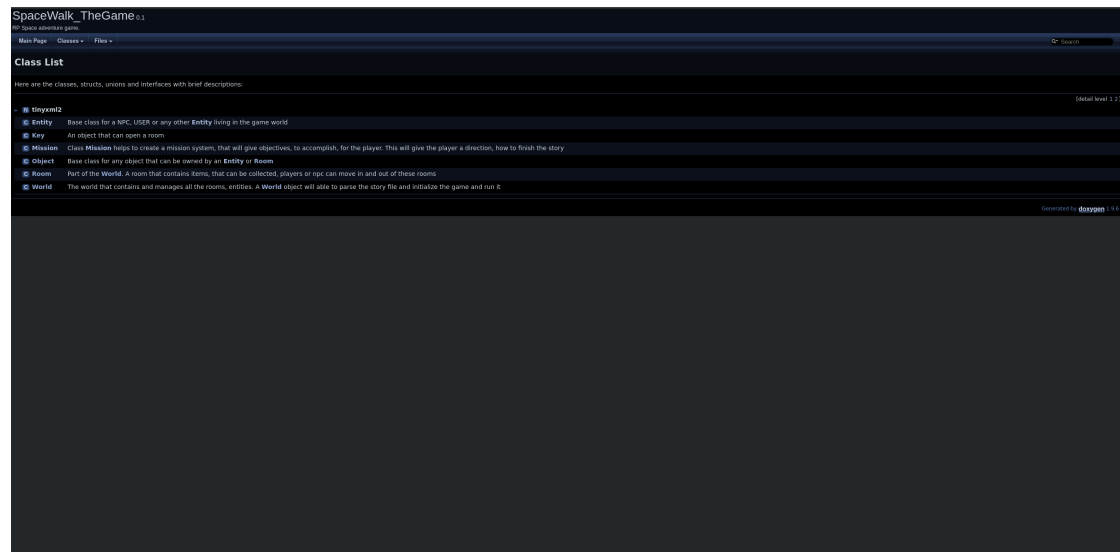


Figure 8: Doxygen

9 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

10 Glossary

- úrkalandjáték:
- NPC Non - Playable Character: A gép által vezérelt entitás.

- utópia:
- sci-fi:
- mesterséges intelligencia: