

BLÁTHY



SZAKDOLGOZAT

MCraftingTree

2022/23

Konzulens: Szegedi Barnabás

Készítette: Tóth Dániel Mátyás 5/13.S

Szakdolgozat adatlap

Szakdolgozat készítőjének neve:

Tóth Dániel Mátyás

Elérhetősége(i):

dani.toth2003@gmail.com

+36-30-603-1890

A szakdolgozat témája:

Egy C#-ban fejlesztett program, aminek a feladata hogy tároljon a Minecraft nevű játékból tárgyakat és az elkészítési módjukat, és annak az adatait kezelje.

Szakdolgozat címe:

MCraftingTree

Konzulens tanár:

Szegedi Barnabás

Kelt: Budapest, 2023. 04. 03.

.....
a szakdolgozat készítőjének aláírása

.....
a konzulens tanár aláírása

Hallgatói nyilatkozat

Alulírott, ezúton kijelentem, hogy a szakdolgozat saját, önálló munkám, és korábban még sehol nem került publikálásra.

tanuló neve

Konzultációs lap

Vizsgázó neve: Tóth Dániel Mátyás 5/13.S

Szakdolgozat címe: MCraftingTree

Program nyújtotta szolgáltatások:

- Minecraft tárgyak és receptek kezelése
- könnyed kezelhetőség

Sorszám	A konzultáció időpontja	A konzulens aláírása
1.	2022.10.26	
2.	2022.11.30	
3.	2022.12.21	
4.	2023.01.09	
5.	2023.02.22	
6.	2023.03.22	

A szakdolgozat beadható:

A szakdolgozatot átvettem:

Budapest, 2023. Budapest, 2023.

.....

Konzulens
tanfolyamfelelő

A szakképzést folytató intézmény

Tartalom

1. Köszönetnyilvánítás	6
2. Bevezetés	7
3. Feladat-specifikáció	8
4. Fejlesztői dokumentáció	8
4.1. Alapfogalmak	8
4.2. Fejlesztői környezet	11
4.3. Visual Studio 2022 Community Edition	11
4.4. MonoDevelop	12
4.5. Entity Framework 6	12
4.6. SQL Server Express LocalDB	13
4.7. Az Adatbázis	17
4.7.1. Items tábla	18
4.7.2. Types tábla	19
4.7.3. MobDrops tábla	20
4.7.4. Furnace tábla	21
4.7.5. Brewing tábla	22
4.7.6. CraftingTable tábla	23
4.8. Osztályok, változók	26
4.8.1. Globális változók	26
4.8.2. MainWindow	26
4.8.3. CrafigTree	31
4.9. Továbbfejlesztési lehetőségek	32
5. Felhasználói dokumentáció	33
5.1. Mi az a MCraftintTree?	33
5.2. Rendszerkövetelmények	34

5.3.	Telepítés	34
5.4.	Alkalmazás használata	36
5.4.1.	A felhasználói felület.....	36
5.4.2.	A funkciók helyes használata	39
6.	Irodalomjegyzék	41

1. Köszönetnyilvánítás

Köszönöm Deli Bencének, Szigeti Péternek, A Gay Motherland-ből Dmxk-nak és Noel-nek hogy tesztelték a programomat, és javaslatokat tettek, hogy hogyan lehet használhatóbb.

2. Bevezetés

A programom készítéséhez az inspiráció akkor jött, mikor játszottam a Minecraft¹nevű játékkal, és egy barátomnak teszteltem egy modifikált verzióját. Sokkal több tárgy, ellenség és barkácsolási recept volt benne, mint az alap játékban, ezért rengeteg mindent nem tudtam, hogy hogyan kell megcsinálni, és hogy milyen tárgyakba kerülne. Annyit tudtam, hogy melyik tárgyat kell megszereznem ahhoz, hogy „győzzek”, amihez egy nagyon komplex recept tartozott. Ekkor volt egy gondolatom, hogy: „Hmm, de jó lenne hogyha lenne egy program vagy egy modifikáció ahhoz, hogy megmutassa nekem a szükséges alapanyagokat ehhez a tárgyhoz...”, és úgy döntöttem, hogy megcsinálom.

Először is, ki kellett találnom, hogy hogyan és mivel hozom létre, és mi kell ahhoz, hogy úgy működjön, ahogy azt én akarom. Ideálisan Java-ban lett volna jó megírni, hiszen a játék is Java 8, majd Java 17-ben íródott, és könnyen tudtam volna importálni az osztályfájlokat magából a .jar fájlból, de helyette a C# mellett döntöttem, mert ahhoz jobban értek, mint a Java-hoz. Megterveztem az adatbázis alapjait draw.io-ban, hogy könnyen tudjak változtatni az oszlopokon. A grafikai felülethez WPF-et használtam, mert gyorsabb, modernebb és több funkcionalitást nyújt a Windows Forms-hoz képest. Emellé még Material Design-t is használtam, ami a Bootstrap-hez hasonlóképpen egy felhasználói felület keretrendszer, hogy professzionálisan nézzen ki a program. Kitaláltam, hogy akarom, hogy a program kinézzen, milyen funkciói legyenek, és nekikezdttem megírni a programot, az adatbázissal kezdve. Entity Framework 6-ot használva megcsináltam a terveim alapján, és nagyon alap funkciókkal teszteltem, hogy be tud-e olvasni és lehet-e rá írni. Ezután megírtam, hogy lehessen hozzáadni tárgyakat az adatbázishoz, hogy dobja-e a megadott tárgyat egy ellenség, ha igen melyik és mekkora eséllyel, a tárgyakkal asszociált típus, tárgyakból készített receptek, legyen az kemencében kisütés, alkímiában készült főzetek, vagy csak egyszerűen barkácsolások. Ezután kidolgoztam, hogy megjelenjenek a tárgyak egy listában, a velük asszociált képekkel együtt, megcsináltam, hogy be lehessen tölteni recepteket azzal, hogy behúzol egy képet a

¹ <https://www.minecraft.net/en-us>

barkácsasztal menüjébe a program közepén, és hogy kiszámolja hogy milyen tárgyak kellenek a megadott receptnek az elkészítéséhez, és mennyi belőlük. Végül hozzáadtam alap adatokat az adatbázishoz, tárgyak, recepteket és típusok amik megjelennek az alap játékban is, modifikációk nélkül.

Végül csak a tesztelés és ellenőrzés maradt, amik sikeresek voltak.

3. Feladatspecifikáció

A szoftver lényege a Minecraft játékból tárgyak és receptek felvétele és kezelése, hogy tudjuk, hogy a játékban mennyibe kerülnek megadott tárgyak.

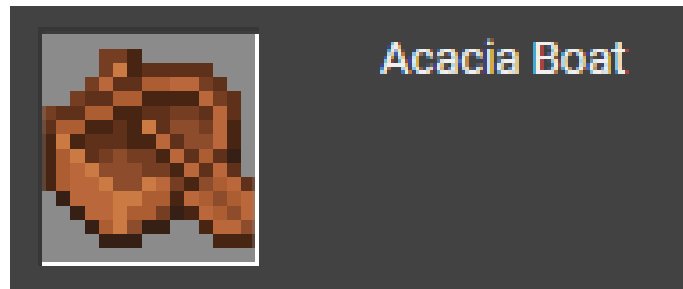
Célok:

- Tárgyak kezelése és felvétele, típusokkal és dobási eséllyel
- Receptek felvétele
- Alap tárgyak megkeresése egy megadott tárgy receptjéhez

4. Fejlesztői dokumentáció

4.1. Alapfogalmak

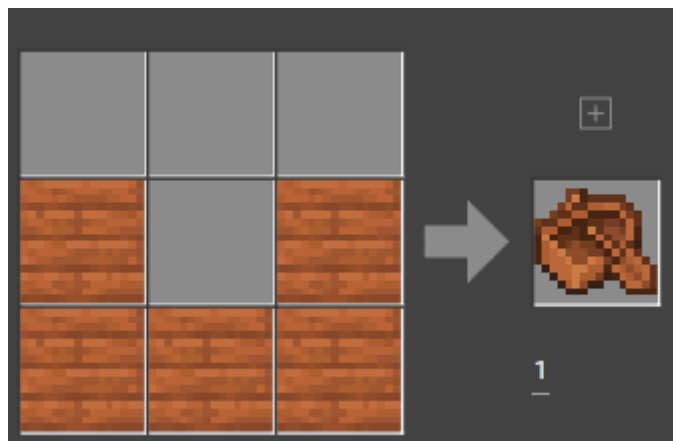
- Modifikáció: Játékosok által készített változtatások a játékban, például extra receptek és/vagy tárgyak, legtöbbször CurseForge-ról vagy Modrinth-ról szerezhetők be, de saját magunk is tudunk csinálni, ha ismerjük a Java-t.
- Tárgy: egy Minecraft-ból származó elem, amivel tudjuk befolyásolni a világot azzal, hogy letesszük a világba, felveszünk velünk valamit a világból, vagy csinálunk belőlük másik tárgyakat, melyek képesek az korábban felsoroltakra.



1. ábra

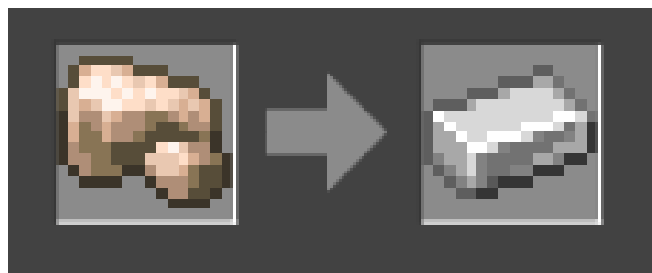
- Recept: Egy megadott felhasználói felületen milyen tárgyakat kell felhasználni ahhoz, hogy más tárgyakat lehessen létrehozni belőlük. Ezeknek van 3 fajtája:

- Barkácsoló asztal: egy háromszor hármass felületen lehet azonnal más tárgyakat létrehozni recepttől függően, vagy egy előre megadott alakzatban, vagy alakzat nélkül. Van egy megadott mennyisége, hogy mennyi tárgyat kapsz egy megadott receptből.



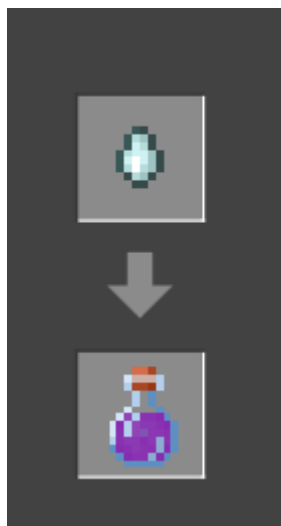
2. ábra

- Kemence: Egy az egyhez átalakítás egy tárgyból a másikba, nem azonnali, de ezt nem reprezentálom a programban. Használható ha valamit ki kell sütni vagy valamit beolvasztani.



3. ábra

- Alkímia asztal: Egy alkímiai hozzávalót belerakunk egytől háromig terjedő vízzel vagy főzettel teli üvegbe, és belefőzzük. Leginkább főzetek készítéséhez használatos.



4. ábra

Ezt az alkalmazást túlnyomó többségben a Minecraft játékosai fogják használni, azon belül azok, akik egyedi és/vagy másik játékosok által terjesztett modifikációkat használnak a játékhoz. Leginkább ahhoz fogják használni, hogy megnézzék milyen nyersanyagokba, azaz annál jobban le nem bonthatóbb tárgyakba, kerül egy megadott tárgy a hozzá asszociált receptet használva. Ezért észben kell tartani hogy mi a legfontosabb miközben változtatásokat hozunk létre a programhoz: a tárgyak és a receptek vannak az előnyben, minden más extra; Ha valami blokkolja a tárgyak és/vagy receptek rendes működését, azt vagy törölni vagy változtatni kell kiadás előtt.

4.2. Fejlesztői környezet

Az általam használt fejlesztői környezet a Visual Studio 2022 Community Edition. Egy IDE (*Integrált Fejlesztői Környezet*) a C#-hoz és a .NET Framework-ös fejlesztéshez. Ideális a beépített funkciói, például az IntelliCode, az integrált Git hozzáférés és a NuGet bővítmények beszerzése miatt, de csak Windows-on vagy Linux-on Wine-ban használható, de az utóbbit gyengén működik.

Ha egy Linux disztribúción szeretné fejleszteni, akkor a MonoDevelop-ot tudom ajánlani, de tesztelni a programot Wine-ban kell, mert a WPF nem támogatott Linux-on.

A használt adatbázis az MSSQL alapú LocalDB, amit Entity Framework 6-al kezelünk.

4.3. Visual Studio 2022 Community Edition

A Visual Studio Community Edition² az egy Microsoft által fejlesztett ingyenes IDE diákok, nyílt forráskódú és egyéni fejlesztők számára. Alapból nem képes adatbázisokat kezelni, de be lehet szerezni hozzá csomagokat, amik képesek rá. Több szolgáltatást is nyújt, ami miatt előnyös a használatra:

- NuGet: Egy csomagkezelő, amit elsősorban olyan szoftverek csomagolására és megosztására van, amik .NET Framework-el lettek megírva.
- IntelliCode: Egy mesterséges intelligencia, amely képes kiegészíteni, átalakítani és befejezni kódjaidat egy sorban. Jelentősen meggyorsítja a kódírást.
- Git: Egy elosztott verziókezelő szoftver, amit arra használunk hogy több ember is tudja fejleszteni a programot több gépen, és hogy bárhol el lehessen érni a forráskódot. Ez a funkció bele van építve a Visual Studio-ba, hogy gyorsan és könnyen fel lehessen nyomni a legutóbbi változtatásokat a Git repository-ba.

² <https://visualstudio.microsoft.com/vs/community/>

4.4. MonoDevelop

A MonoDevelop³ az egy nyílt forráskódú IDE amit lehet használni C#-hoz. Hasonló módon a Visual Studio-hoz, nem tud kezelni magától adatbázist. Csak akkor javaslom használni ezt, ha a Visual Studio nem működik, vagy Linux-on akarja fejleszteni a programot.

Előnyei:

- Gyorsabb és kevesebb tárhelyet fogyaszt a Visual Studio-hoz képest.
- Képes kódokat kiegészíteni automatikusan.

Hátrányai:

- Kevésbé stabilabb, mint a Visual Studio.
- Nincsenek bővítmények a fejlesztő számára (Ez alatt nem NuGet csomagokat kell érteni).

4.5. Entity Framework 6

Az Entity Framework (EF) egy adatbázis-kezelő keretrendszer a Microsofttól, amely lehetővé teszi az alkalmazások számára, hogy egyszerűen kezeljék az adatbázisokkal való kommunikációt az objektumorientált programozás (OOP) szemléletmódjában.

Az Entity Framework 6 az EF előző verziója, amely 2013-ban jelent meg. Az EF 6 lehetővé teszi a .NET keretrendszer használatával történő adatbázis-kezelést, és támogatja a Code First, Database First és Model First megközelítéseket. A Code First megközelítés lehetővé teszi, hogy a fejlesztők az alkalmazás osztályaiból automatikusan létrehozzák az adatbázis sémáját. A Database First megközelítés hasonlóan működik, de az adatbázis sémáját már létrehozzák, és az EF generálja a megfelelő osztályokat az adatbázisból. A Model First megközelítés lehetővé teszi az osztályok és az adatbázis sémájának egyszerre történő létrehozását.

³ <https://www.monodevelop.com/>

Az EF 6 támogatja a LINQ (Language Integrated Query) nyelv használatát az adatok lekérdezéséhez, a lazán összekapcsolt objektumgráfok kezelését és az adatok módosítását az objektumok által. Az EF 6 a többi .NET keretrendszerrel is jól integrálódik, és lehetővé teszi az alkalmazások egyszerű telepítését és használatát.

Az Entity Framework 6 az EF utolsó verziója, mivel a Microsoft azóta az EF Core-ra fókuszálta a figyelmét, amely a .NET Core-hoz és az ASP.NET Core-hoz készült. Az EF Core ugyanazokat a funkciókat kínálja, mint az EF 6, de optimalizáltabb és gyorsabb adatbázis-kezelést biztosít.

4.6. SQL Server Express LocalDB

A LocalDB az egy Microsoft által fejlesztett és fenntartott SQL alapú adatbázis szerver. Mi ezt használjuk a programunk adatbázisaként, és ezen kezeli az Entity Framework 6 az adatainkat.

Mivel ez egy függősége a programnak, ezért installálni kell, amit a következőképpen lehet elvégezni:

- Rákeresünk arra, hogy „LocalDB”, és az első Microsoft oldalt megnyitjuk, vagy beírjuk a keresőbe a következő linket:

<https://learn.microsoft.com/en-us/sql/database-engine/configure-windows/sql-server-express-localdb?view=sql-server-ver16>

- Meg kell jelennie a következő oldalnak, ahol láthatunk egy „Installation media” menüpontot, az alatt egy „SQL Server Express 2019” nevű linket, és arra rányomunk.

Learn / SQL / SQL Server /

SQL Server Express LocalDB

Article • 12/01/2022 • 9 minutes to read • 18 contributors [Feedback](#)

Applies to: SQL Server

Microsoft SQL Server Express LocalDB is a feature of SQL Server Express targeted to developers. It is available on SQL Server Express with Advanced Services.

LocalDB installation copies a minimal set of files necessary to start the SQL Server Database Engine. Once LocalDB is installed, you can initiate a connection using a special connection string. When connecting, the necessary SQL Server infrastructure is automatically created and started, enabling the application to use the database without complex configuration tasks. Developer Tools can provide developers with a SQL Server Database Engine that lets them write and test Transact-SQL code without having to manage a full server instance of SQL Server.

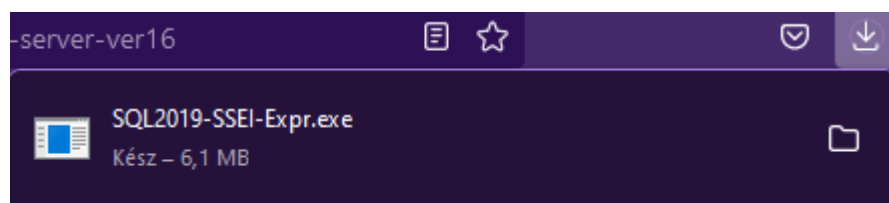
Installation media

LocalDB is a feature you select during SQL Server Express installation, and is available when you download the media. If you download the media, either choose **Express Advanced** or the **LocalDB** package.

- **SQL Server Express 2019**
- SQL Server Express 2017
- SQL Server Express 2016

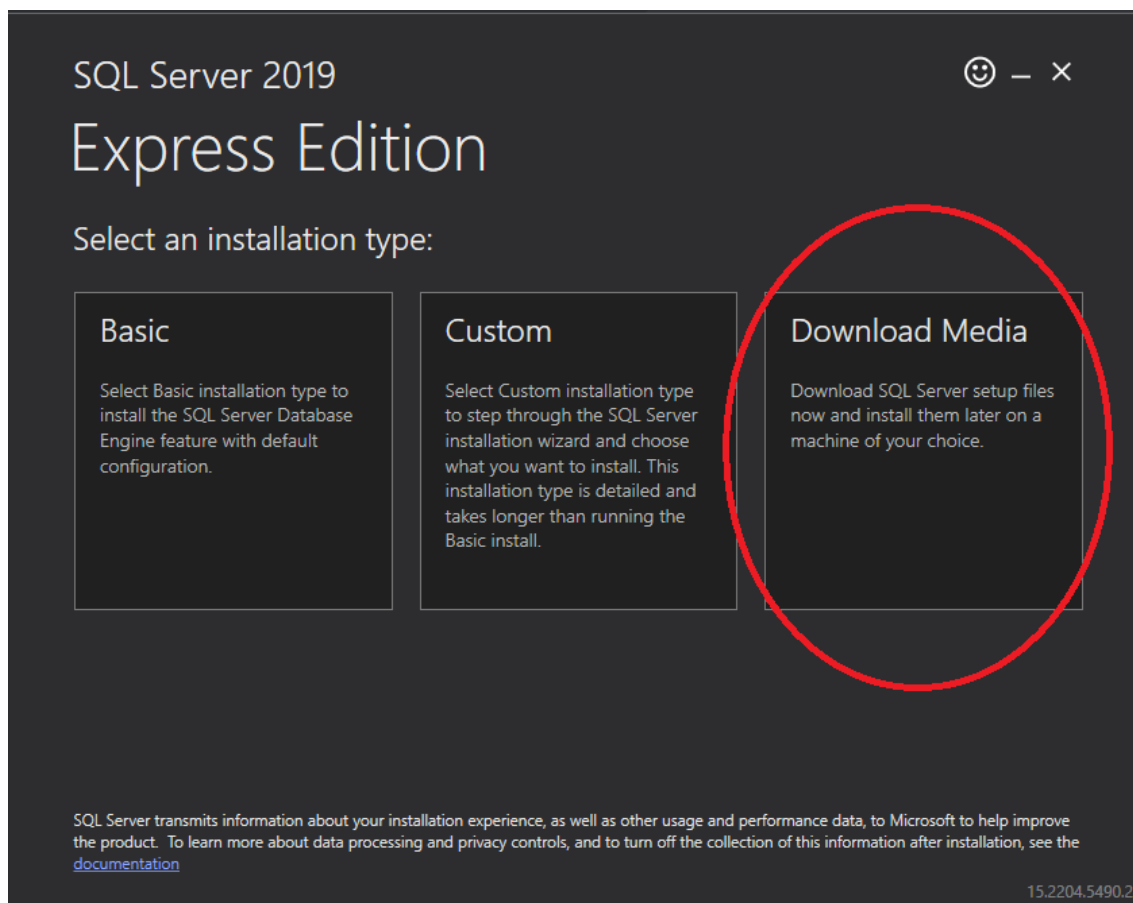
4

- Ez elkezdi nekünk letölteni egy .exe fájlt, amire majd ha kész van rákattintunk vagy a böngésző jobb felső sarkában való menüből, vagy a „Letöltések” mappából.

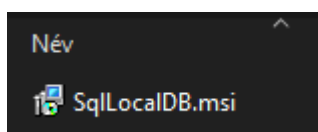


- Ez feldob nekünk egy telepítőt, ahol rá kell kattintanunk a „Download Media” gombra, majd ki kell választanunk a „LocalDB” lehetőséget, és végül rákattintunk a „Download” gombra.

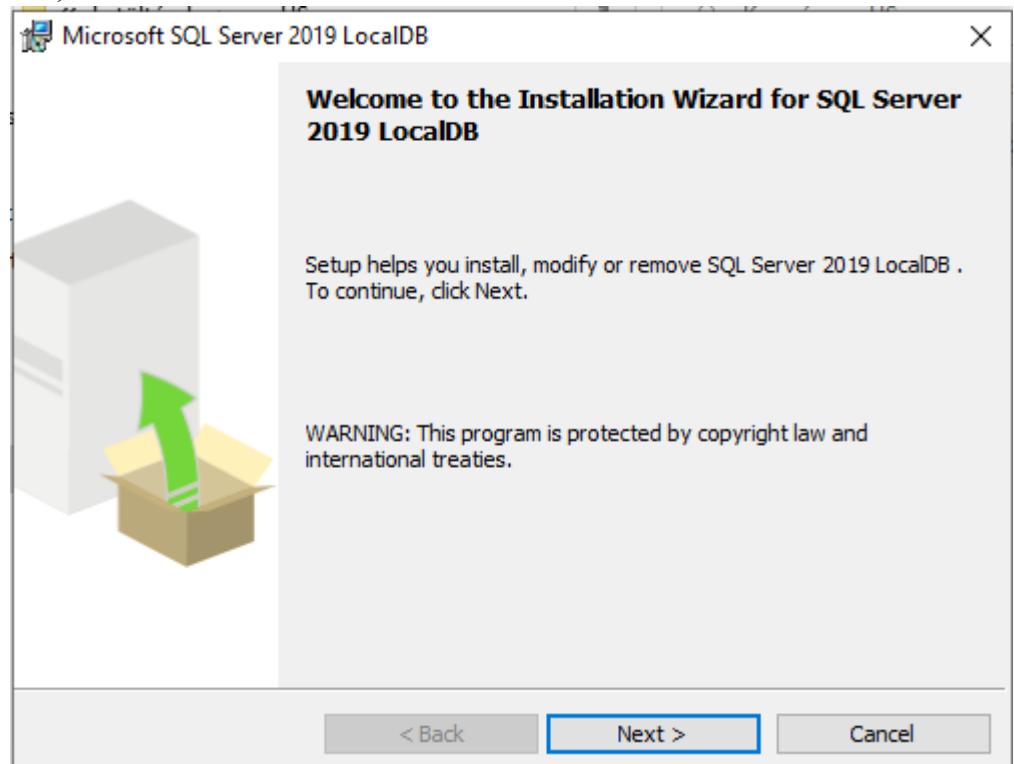
⁴ Forrás: <https://learn.microsoft.com/en-us/sql/database-engine/configure-windows/sql-server-express-localdb?view=sql-server-ver16>



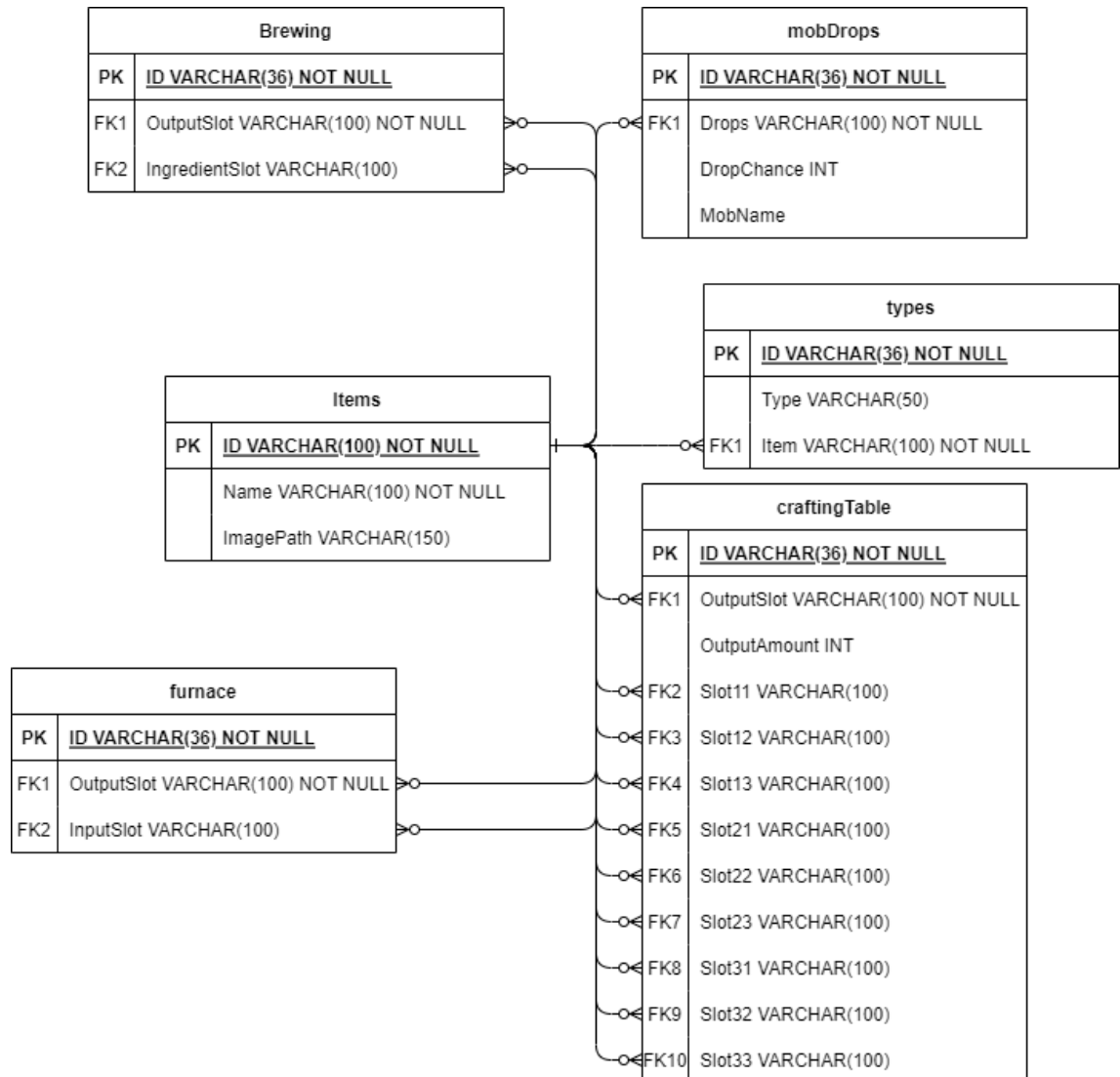
- Miután befejezte a letöltést, rákattintunk arra, hogy „Open Folder”, és ezután bezárhatjuk.
- Az előbbi gomb megnyitott nekünk egy mappát, amiben van egy „SqlLocalDB.msi” nevű fájl. Ezt megnyitjuk, és feldob nekünk egy telepítőt.



- Végig megyünk a telepítő lépésein, megvárjuk amíg befejezi, és készen van, már lehet használni.



4.7. Az Adatbázis



4.7.1. Items tábla

Az Items tábla tartalmazza a következő oszlopokat:

- ID: A játékban szereplő ID, pl.: „minecraft:acacia_boat”
 - 100 karakter maximum hossz
 - Nem lehet null
- Name: A játékban megjelenő neve a tárgynak, pl.: „Acacia Boat”
 - 100 karakter maximum hossz
 - Nem lehet null
- ImagePath: egy relatív útvonal a program .exe fájlától számítva a tárgygal asszociált képhez, pl.: „/ImageResources/Items/acacia_boat.png”
 - 150 karakter maximum hossz
 - Null lehet, ha nincs hozzá kép
 - Nem egy képet tartalmaz

Az Items tábla a játékban található tárgyak eltárolására van kitalálva, ezeket használja a program majdnem mindegy funkciója.

A következőképpen lehet felvenni új elemeket:

- C#, Entity Framework 6:

```
Items item = new Items() { ID = "minecraft:item",  
Name = "Item", ImagePath = "/ImageResources/Items/item.png"  
};
```

- SQL:

```
INSERT INTO Items (ID, Name, ImagePath)  
  
VALUES ("minecraft:item", "Item",  
"/ImageResources/Items/item.png");
```

4.7.2. Types tábla

A Types tábla tartalmazza a következő oszlopokat:

- ID: Egy automatikusan generált GUID (Globálisan Egyedi Identifikáció) token, nem a felhasználó által beírt adat.
 - 36 karakter hossz összesen
 - Nem lehet null
 - Gép által generált
- Type: A típus neve, ami szerint csoportosítva vannak tárgyak
 - 150 karakter maximum hossz
 - Nem lehet null
- Item: A tárgy, ami kapcsolódik a típushoz
 - Egy Items típus, lásd az [Items táblát](#)

A Types tábla a játékban megtalálható típusokat tárolja, és a majd később továbbfejlesztett verzióban fontos szerepet fog játszani a CraftingTable típus barkácsolási receptjeiben.

A következőképpen lehet felvenni új elemeket:

- C#, Entity Framework 6:

```
Items item = new Items() { ID = "minecraft:item",  
Name = "Item", ImagePath =  
"/ImageResources/Items/item.png" };  
  
Types type = new Types() { ID =  
Guid.NewGuid().ToString(), Item = item, Type = "type" };
```

- SQL:

```
INSERT INTO Types(ID, Item, Type)  
  
VALUES("{generálj le egy GUID-ot és rakd ide}",  
"minecraft:item", "type");
```

4.7.3. MobDrops tábla

A MobDrops tábla tartalmazza a következő oszlopokat:

- ID: Egy automatikusan generált GUID (Globálisan Egyedi Identifikáció) token, nem a felhasználó által beírt adat.
 - 36 karakter hossz összesen
 - Nem lehet null
 - Gép által generált
- MobName: A játékban szereplő lény neve, ami dobhatja a megadott tárgyat
 - 50 karakter maximum hossz
- DropChance: Mekkora az esély, hogy a megadott lény dobni fogja a tárgyat
 - 0-100 skálájú egész szám
 - Egész számok százalékká válnak, pl.: 1 -> 1%
- Drops: Milyen tárgyat dob a megadott lény
 - Egy Items típus, lásd az Items táblát

A MobDrops tábla a játékban megjelenő lényeknek azon tárgyait tárolja, amit esélyesen dobhatnak halálukkor.

A következőképpen lehet felvenni új elemeket:

- C#, Entity Framework 6:

```
Items item = new Items() { ID =  
    "minecraft:porkchop", Name = "Porkchop",  
    ImagePath = "/ImageResources/Items/porkchop.png"  
};
```

```
MobDrops mobDrop = new MobDrops() { ID =  
    Guid.NewGuid().ToString(), MobName = "Pig", DropChance =  
    100, Drops = item };
```

- SQL:

```
INSERT INTO Types(ID, MobName, DropChance, Drops)

VALUES("{generálj le egy GUID-ot és rakd ide}",

"Pig", 100, "minecraft:porkchop");
```

4.7.4. Furnace tábla

A Furnace tábla tartalmazza a következő oszlopokat:

- ID: Egy automatikusan generált GUID (Globálisan Egyedi Identifikáció) token, nem a felhasználó által beírt adat.
 - 36 karakter hossz összesen
 - Nem lehet null
 - Gép által generált
- OutputSlot: Milyen tárgy az eredménye a megadott receptnek
 - Egy Items típus, lásd az Items táblát
- InputSlot: Milyen tárgyból jön létre a kiadott tárgy
 - Egy Items típus, lásd az Items táblát

A Furnace tábla a játékban lévő kemencében tett recepteket tárolja. Egy az egyhez átalakítás történik, szóval egy tárgyból csak egy tárgy készülhet.

A következőképpen lehet felvenni új elemeket:

- C#, Entity Framework 6:

```
Items inputItem = new Items() { ID =
"minecraft:porkchop", Name = "Porkchop",
ImagePath = "/ImageResources/Items/porkchop.png"
};

Items outputItem = new Items() { ID =
"minecraft:cooked_porkchop", Name = "Cooked
Porkchop", ImagePath =
"/ImageResources/Items/cooked_porkchop.png" };
```

```
Furnace recipe = new Furnace() { ID =
Guid.NewGuid().ToString(), OutputSlot = outputItem,
InputSlot = inputItem };
```

- SQL:

```
INSERT INTO Furnace(ID, OutputSlot, InputSlot)

VALUES("{generálj le egy GUID-ot és rakd ide}",
"minecraft:cooked_porkchop", "minecraft:porkchop");
```

4.7.5. Brewing tábla

A Brewing tábla tartalmazza a következő oszlopokat:

- ID: Egy automatikusan generált GUID (Globálisan Egyedi Identifikáció) token, nem a felhasználó által beírt adat.
 - 36 karakter hossz összesen
 - Nem lehet null
 - Gép által generált
- OutputSlot: Milyen tárgy az eredménye a megadott receptnek
 - Egy Items típus, lásd az Items táblát
- IngredientSlot: Milyen hozzávalóból jön létre a kiadott tárgy
 - Egy Items típus, lásd az Items táblát

A Brewing tábla a játékban lévő alkímiai asztalban tett összes receptet tartalmazza. A célja, hogy főzeteket hozzon létre vizes üvegekbe. A következőképpen lehet felvenni új elemeket:

- C#, Entity Framework 6:

```
Items ingredientItem = new Items() { ID =
"minecraft:ghast_tear", Name = "Ghast Tear",
ImagePath =
"/ImageResources/Items/ghast_tear.png" };
```

```

Items outputItem = new Items() { ID =
"minecraft:regeneration_potion ", Name =
"Regeneration Potion", ImagePath =
"/ImageResources/Items/potion.png" };

Brewing recipe = new Brewing() { ID =
Guid.NewGuid().ToString(), OutputSlot = outputItem,
IngredientSlot = ingredientItem };

```

- SQL:

```

INSERT INTO Brewing(ID, OutputSlot,
IngredientSlot)

VALUES("{generálj le egy GUID-ot és rakd ide}",
"minecraft:cooked_porkchop", "minecraft:porkchop");

```

4.7.6. CraftingTable tábla

A CraftingTable tábla tartalmazza a következő oszlopokat:

- ID: Egy automatikusan generált GUID (Globálisan Egyedi Identifikáció) token, nem a felhasználó által beírt adat.
 - 36 karakter hossz összesen
 - Nem lehet null
 - Gép által generált
- Slot11: Egy tárgyat tárol, a barkácsasztal bal-felső részéhez tartozik
 - Egy Items típus, lásd az [Items táblát](#)
- Slot12: Egy tárgyat tárol, a barkácsasztal közép-felső részéhez tartozik
 - Egy Items típus, lásd az [Items táblát](#)
- Slot13: Egy tárgyat tárol, a barkácsasztal jobb-felső részéhez tartozik
 - Egy Items típus, lásd az [Items táblát](#)
- Slot21: Egy tárgyat tárol, a barkácsasztal bal-közép részéhez tartozik
 - Egy Items típus, lásd az [Items táblát](#)
- Slot22: Egy tárgyat tárol, a barkácsasztal középső részéhez tartozik

- Egy Items típus, lásd az Items táblát
- Slot23: Egy tárgyat tárol, a barkácsasztal jobb-közép részéhez tartozik
 - Egy Items típus, lásd az Items táblát
- Slot31: Egy tárgyat tárol, a barkácsasztal bal-alsó részéhez tartozik
 - Egy Items típus, lásd az Items táblát
- Slot32: Egy tárgyat tárol, a barkácsasztal közép-alsó részéhez tartozik
 - Egy Items típus, lásd az Items táblát
- Slot33: Egy tárgyat tárol, a barkácsasztal jobb-alsó részéhez tartozik
 - Egy Items típus, lásd az Items táblát
- OutputSlot: Milyen tárgy az eredménye a megadott receptnek
 - Egy Items típus, lásd az Items táblát
- OutputAmount: Mennyi tárgyat ad a recept
 - Egy természetes szám, felső határa 4 294 967 295

A CraftingTable tábla a játékban szereplő összes barkácsoló receptet tárolja. Ez a fő lényege a programnak, hisz ezeknek az alap tárgyait lehet megkeresni.

A következőképpen lehet felvenni új elemeket:

- C#, Entity Framework 6:

```
Items component1 = new Items() { ID =
"minecraft:flint", Name = "Flint", ImagePath =
"/ImageResources/Items/flint.png" };
```

```
Items component2 = new Items() { ID =
"minecraft:iron_ingot ", Name = "Iron Ingot",
ImagePath =
"/ImageResources/Items/iron_ingot.png" };
```

```
Items outputItem = new Items() { ID =
"minecraft:flint_and_steel ", Name = "Flint and
```



```
Steel", ImagePath =  
"/ImageResources/Items/flint_and_steel.png" };
```

```
CraftingTable recipe = new CraftingTable() { ID =  
Guid.NewGuid().ToString(), Slot11 = component1, Slot12 =  
component2, Slot13 = nullItem, Slot21 = nullItem, Slot22 =  
nullItem, Slot23 = nullItem, Slot31 = nullItem, Slot32 =  
nullItem, Slot33 = nullItem, OutputSlot = outputItem,  
OutputAmount = 1};
```

- SQL:

```
INSERT INTO Furnace(ID, Slot11, Slot12, Slot13,  
Slot21, Slot22, Slot23, Slot31, Slot32, Slot33, OutputSlot,  
OutputAmount)
```

```
VALUES("{generálj le egy GUID-ot és rakd ide}",  
"minecraft:flint", "minecraft:iron_ingot", "-1", "-1", "-  
1", "-1", "-1", "-1", "minecraft:flint_and_steel", 1);
```

4.8. Osztályok, változók

4.8.1. Globális változók

- `ctx`: Egy Context változó, amin keresztül el lehet érni az adatbázist
- `nullItem`: Egy Items típus, ami arra van használva hogy jelölje az üres részeket a recept típusokban.
- `switchScreen`: Egy string típusú változó, arra van használva, hogy lementse melyik recept ablakon vagyunk.
- `ImagePath`: Egy string típusú változó, arra van használva hogy el lehessen érni a legutóbbi lementett tárgy képének a helyét.
- `DialogHostKey`: Egy string típusú változó, arra van kitalálva, hogy lementse melyik tárgy funkció gombjára kattintottunk.
- `RecipeID`: Egy string típusú változó, arra van használva, hogy lementse a legutóbbi betöltött recept ID-jét.

4.8.2. MainWindow

MainWindow osztály tartalmazza az összes funkciót és gombot, ami a fő ablakon szerepel.

Funkciók:

- `NewBitmapImage`:
 - Egy, a programtól relatív, útvonalat fogad be egy képhez string formátumban.
 - Ezt az útvonalat befogadja, megkeresi a képet, és csinál belőle egy új BitmapImage típusú képet, amit visszaad.
 - A lényege ennek a funkciónak, hogy megakadályozza az `Image.Load()` metódust, ami megakadályozza, hogy lehessen törölni tárgyakat.

- Biztosra kell menni hogy nem egy teljes útvonalat adunk meg (pl.: C:/users/pista/MCraftingTree/ImageResources/Items/acacia_boat.png) hanem egy relatív útvonalat
- példa:

```
string path = "/ImageResources/Items/acacia_boat.png";
```

```
BitmapImage bmImage = NewBitmapImage(path);
```

Ez nekünk visszaadja a következő képet:



- LoadItems:
 - Feltölti a bal oldali adatrácsot (ItemsDG) tárgyakkal
 - Két változata van, az első:
 - Nem fogad be semmilyen adatot, lekéri magának az adatbázisból
 - Aszinkron, hogy a program indításánál ne lassítja indításnál
 - A második:
 - Befogad egy string típusú változót, amit megnézz, hogy szerepel-e az adatbázisban.
 - Nem aszinkron, nem szükséges a jelenlegi mennyiségű adattal.
 - ImageUidSearch:
 - Add_Recipe gombban van használva, megkeresi egy megadott résznek az Uid-jét, ahol a tárgynak az ID-je van tárolva

- A beviteli adat egy Border típus, mert a részek legkülsőbb eleme, amit lát a program az egy Border
- CheckForRecipes:
 - A feladata, hogy megnézzze, hogy egy tárgynak vannak-e vele asszociált barkácsoló receptjei
 - Befogad egy Items típusú változót
 - Visszaad egy Items típusú listát
- CheckForSmelting:
 - A feladata, hogy megnézzze, hogy egy tárgynak vannak-e vele asszociált kemencés receptjei
 - Befogad egy Items típusú változót
 - Visszaad egy Items típusú változót

Gombok:

- SwitchScreen:
 - Váltogatja a recept ablakokat
 - switchScreen változót beállítja a megfelelő ablak ID-jére
- Add_Recipe:
 - Hozzáad az adatbázishoz egy receptet, attól függően hogy milyen ablakon vagyunk
 - Használja a switchScreen változót, hogy eldöntse melyik táblához adja hozzá a receptet
 - Ha a kiadandó tárgy része üres, nem menti le
 - Ha valamelyik rész a beviteli oldalon üres, hozzáadja a nullItem változót a helyére
- Delete_Recipe:
 - Töröl egy jelenleg betöltött receptet
 - Használja a switchScreen változót, hogy eldöntse melyik táblából törölje
 - Használja a RecipeID változót, hogy eldöntse melyik receptet kell törölni
 - Újratölti a tárgyak adatrácsát, hogy frissüljenek a receptek.

- NumberOnly
 - Nem egy gomb, hanem a Crafting ablakban lévő beviteli mezőben történő minden gomb lenyomáskor ellenőrzi, hogy át lehet-e alakítani a megadott szöveget egy számmá.
 - Ha üres, akkor nem történik semmi.
- SwitchItem:
 - Jobboldalon váltja a menüket az alapanyagok és a tárgykezelő között.
- DialogHost_Button_Prep:
 - Beállítja a DialogHostKey változót, hogy melyik gombra kattintottunk („Add” vagy „Alter”).
 - Ha megnyomjuk az „Alter” gombot, akkor előkészíti a felugró ablakot azzal, hogy belerakja a megváltozandó adatnak a tulajdonságait
- MobDropEnable:
 - A felugró ablakban láthatóvá teszi a MobDrop részt.
- Add_Or_Alter_Item:
 - Használja a DialogHostKey változót, hogy eldöntse melyik gombot nyomtuk meg.
 - Ellenőrzi, hogy az „ItemName” meg az „ItemID” üres-e, ha igen akkor hibád dob.
 - Kettő fő része van, az első:
 - Létrehoz egy Items típusú változót az ItemName és ItemID adataiból
 - Ha a type nem üres, létrehoz egy Types változót
 - Ha a MobName nem üres, létrehoz egy MobDrops változót
 - Ha nem üres az ImagePath változó, hozzáadja az Items típusú változóhoz
 - Hozzáadja az adatbázishoz a létrehozott változókat a megfelelő táblákhoz
 - A második:

- Frissíti a DialogHost_Button_Prep-ben betöltött adatokat
- Delete_Item:
 - Az tárgyak adatrácsból egy kiválasztott tárgyat töröl az adatbázisból
 - Ha egy másik tárgy nem használja a képet amit ez használ, törli a képet az ImageResources-ból
- OpenFile:
 - Megnyitja a Fájlkezelőt, és engedi hogy kiválassz egy képet vagy gif-et, amit akarsz hogy asszociálva legyen a feltöltendő tárggyal
 - Feltölti a megadott képet az ImageResources/Items útvonalú mappába
 - Beállítja az ImagePath változót a relatív útvonalra a képhez (pl.: ImageResources/Items/acacia_boat.png)
- Search_Items:
 - Beolvassa a tárgy adatlista feletti mező értékét
 - Ha üres, akkor meghívja a LoadItems-t adat nélkül
 - Ha nem üres, meghívja a LoadItems-t a megadott szöveggel
 - A Search_Items_Click kattintásra, a SearchBar_KeyDown Enter lenyomására ezt hívja meg
- Item_Source:
 - A tárgy adatrácsból ha elkezdünk húzni egy képet, akkor ez hívodik meg, és be tudjuk dobni egy Item_Destination-al rendelkező részbe
- Item_Destination:
 - A húzott tárgyat be lehet dobni ide, lemásolja a képet és a tárgy ID-jét
- Remove_Image:
 - Ha egy tárggyal rendelkező részre jobb klikkelünk, eltünteti a tárgyat onnan.
- Load_Recipe:

- Ha egy tárgyra kétszer kattintunk a tárgy adatrácsban, három dolog történhet:
 - Ha nincs vele asszociált recept, nem csinál semmit
 - Ha csak egy recept van asszociálva vele, betölti a megadott recept ablakba a receptet
 - Ha több recept is van asszociálva vele, feldob a recept ablak fölé egy adatrácsot, ami tartalmazza az összes vele asszociált receptet, ezek közül lehet rákattintani egyre, ami meghívja a RecipeDG_Chosen funkciót
- Használja a switchScreen változót, hogy eldöntse melyik recept ablakon vagyunk és mit tud betölteni
- RecipeDG_Chosen:
 - A recept adatrácsból kiválasztott receptet betölti a megadott recept ablakba
 - Használja a switchScreen változót, hogy eldöntse milyen ablakon van
- Add_To_Tree:
 - Megnézi egy betöltött receptnek az alapanyagait, ezt átalakítja egy CraftingTree típusúvá, és ezt betölti egy adatrácsba a BaseMaterials menüpontban

4.8.3. CrafterTree

A MainWindow osztályban szerepel, az egyetlen lényege, hogy fel lehessen használni az Add_To_Tree funkcióban az adatrács feltöltésére.

4.9. Továbbfejlesztési lehetőségek

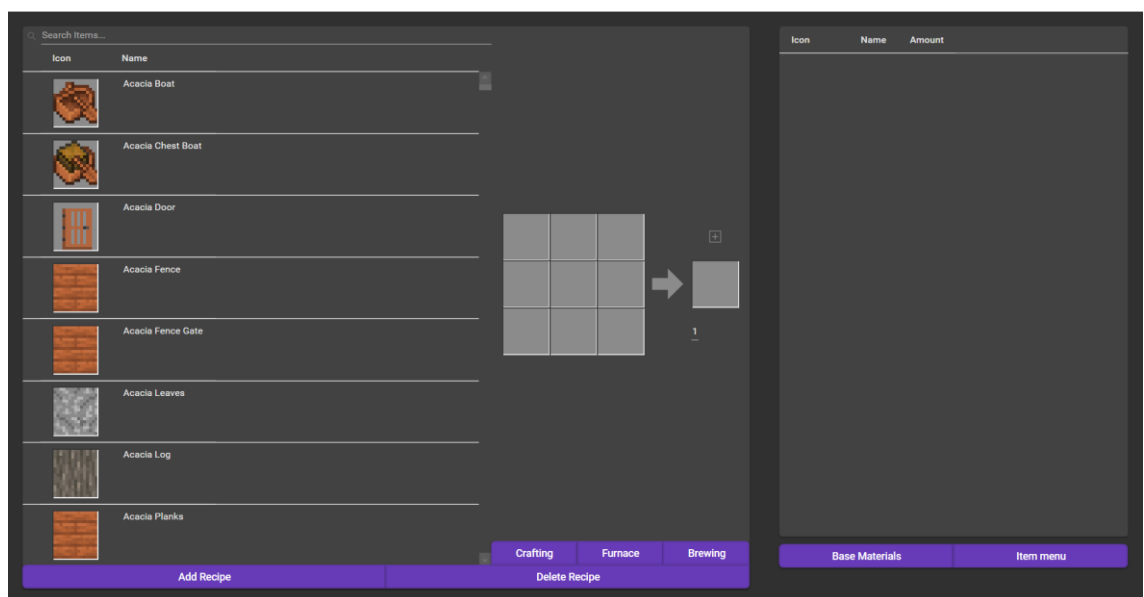
A nagyobb továbbfejlesztési lehetőségek a következők:

- A receptek használják a típusokat tárgyak helyett ahol megfelelő, és váltogasson a különböző tárgyak között a betöltött recept.
- Lehessen beolvasni egy JSON fájlt, ami tartalmazza a recepteket és típusokat.
- Lehessen beolvasni Java osztályfájlokat, amik a tárgyakat tartalmazzák.
- Csináljon egy képet a Add_To_Tree funkció, ami mutatja az útvonalat hogy hogyan lehet megcsinálni egy megadott tárgyat.
- A MobDrops tábla is használatban legyen a Add_To_Tree funkcióban, és mutassa a lénynek a fejét a képen.
- Úgy jelenjenek meg a tárgyak és a lények, mint a játékban, azaz egy 3D-s objektumként.

5. Felhasználói dokumentáció

5.1. Mi az a MCraftintTree?

Az MCraftingTree az egy magánálló program a Minecraft nevű játékhoz kitalálva. A lényege az az, hogy meg lehessen keresni egy tárgynak az alapanyagait. Az alkalmazást lehet használni Minecraft tárgyak és receptek felvételére és kezelésére. A program szolgáltatja a felhasználót az alap, meg nem változtatott játék tárgyaival és receptjeivel, és ha a felhasználó egy felhasználók által készített modifikációnak tárgyait és receptjeit szeretné használni, azt magának kell felvennie az adatázisba.



5.2. Rendszerkövetelmények

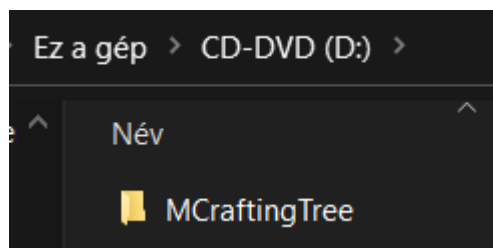
A minimum gépigény:

- CPU: Intel Core I5-2500 / AMD FX 6300
- Memória: 2 GB RAM
- Tárhely: 50 MB szabad
- DirectX: 9
- Grafikai kártya: Nvidia NVS 310 / AMD Radeon HD4670M

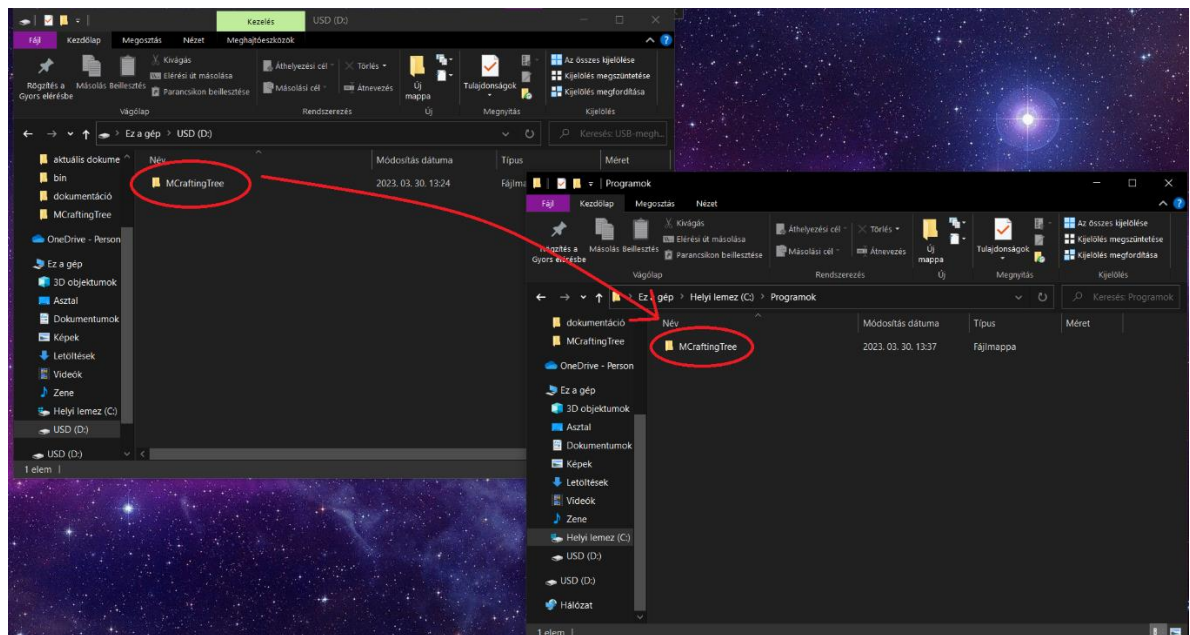
A program nem nagy gépigényű, és elfut minden, amin Windows 10 elfut, szóval a leírt minimum gépigények, a tárhely kivételével, mind a Windows 10 gépigényei.

5.3. Telepítés

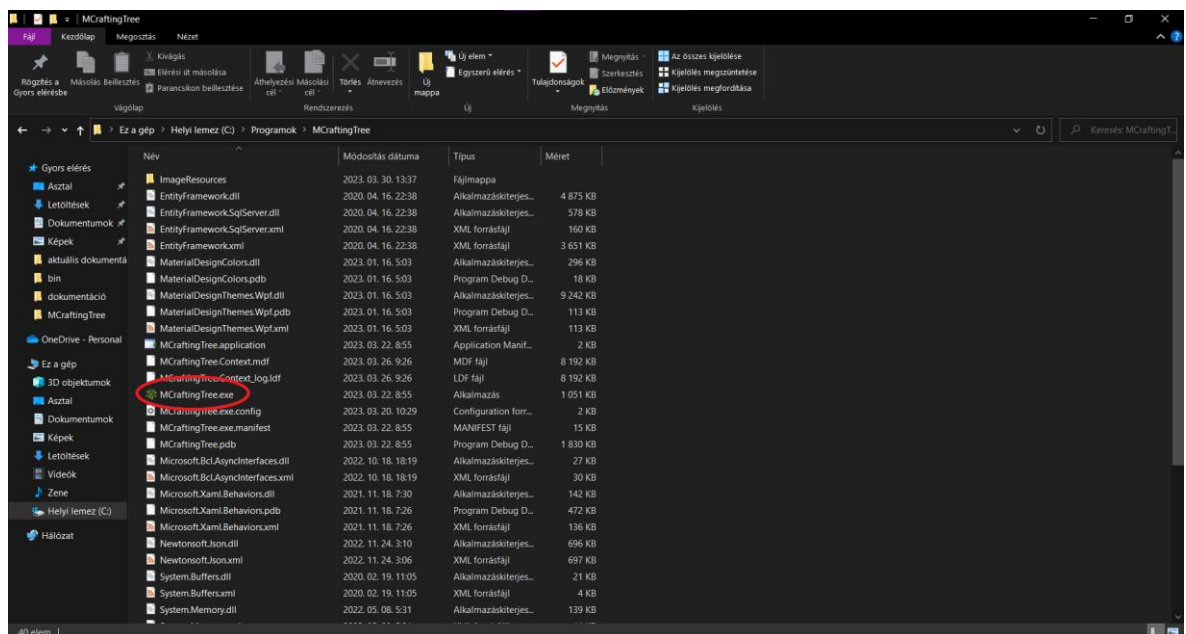
A dokumentáció leghátsóbb oldalán van egy cd, az azon szerepelnek egy „MinecraftTree” mappa.



Erre rákattintunk, és megnyomjuk a CTRL és C billentyűt, és egy ideális helyen megnyomjuk a CTRL és V gombot, vagy csak rákattintunk a mappára, és áthúzzuk a gépre.



Ezután el tudjuk indítani a programot a MCraftingTree.exe fájlal.

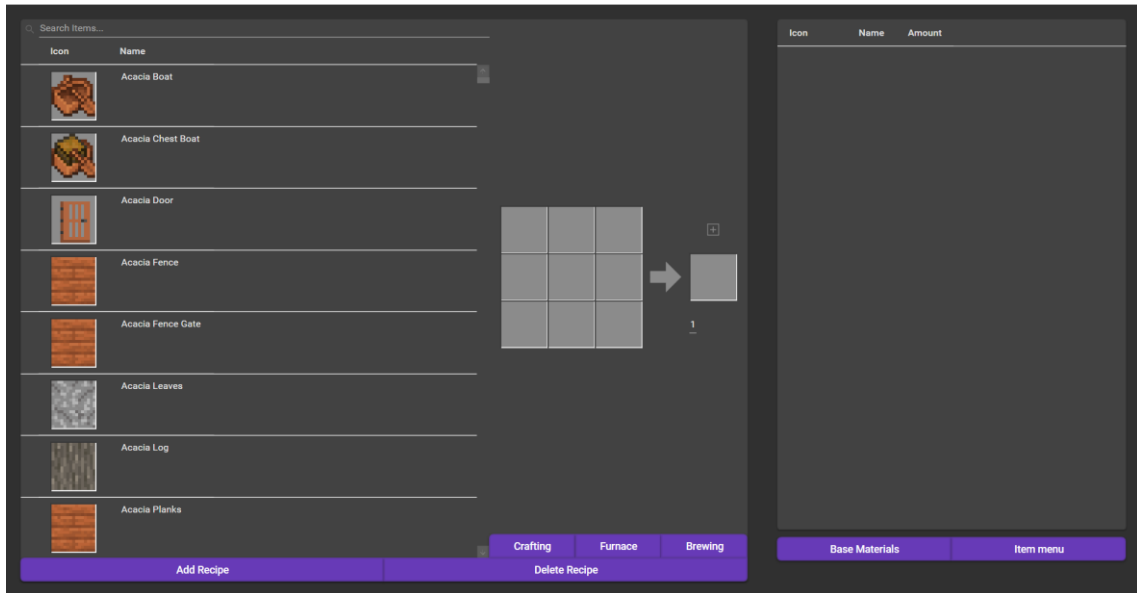


Ha a program nem indul el, telepíteni kell a LocalDB-t, amire a telepítési folyamat le van írva a Fejlesztői dokumentáció részben (Tartalomjegyzék 3.4.)

5.4. Alkalmazás használata

5.4.1. A felhasználói felület

A felhasználói felület a következőképpen nézz ki:

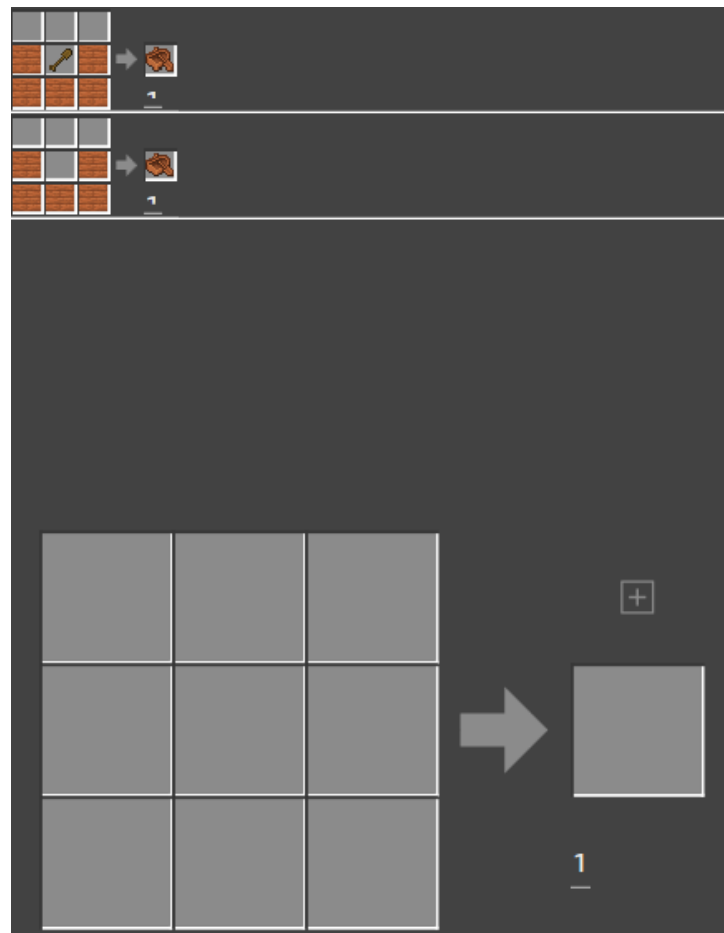


Az elrendezés a következő:

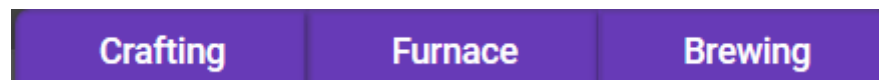
- A baloldali menü a recept kezelő
 - A menü baloldalán van a tárgy adatrács
 - Ha duplán kattintunk egy képre, be lehet tölteni a tárgyal asszociált receptet, ha van
 - Ha elkezdünk húzni egy képet, akkor betehetjük egy részbe a recept menükben
 - A tárgy adatrács felett van a tárgykereső
 - Jobb oldalt van a recept ablak
 - A recept ablaknak a következő alablakai vannak: Crafting/Barkács, Furnace/Kemence, Brewing/Alkímia



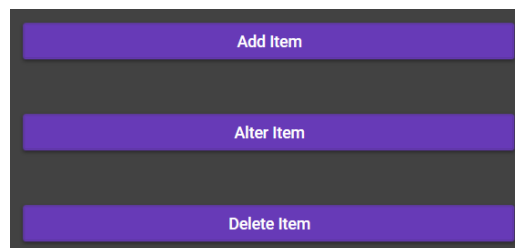
- Mindegyiken szerepelnek rések, amikbe bele lehet húzni tárgyakat az adatrácsból, és ha szerepel bennük egy tárgy amit nem akarunk, lehet rá jobb klikkelni hogy eltüntessük
- Ha egy receptet próbálunk betölteni, de egy tárgyal több recept van asszociálva, akkor feldob nekünk egy adatrácsot a recept ablak fölé, amiből egy szimpla kattintással kiválasztjuk a nekünk megfelelő receptet



- A barkács ablakon van egy gomb plusz jellel, ami kiszámolja az alapanyagokat, ami megtekinthető a Base Materials menüben
- Szerepel még egy TextBox, ami mutatja hogy mennyi tárgy jön létre ebből a receptből, bele lehet írni egy egész számot
- A menü alján vannak a funkció gombok
 - A Crafting, Furnace, Brewing gomb váltogatja a recept menüket



- Az Add Recipe gomb recepteket ad az adatbázishoz, amit a recept ablakokon lehet összerakni
- A Delete Recipe gomb töröl egy receptet, ami be van töltve a recept ablakban
- A jobboldali menü a tárgykezelő és az alapanyagok kijelző
 - A menü alján szerepelnek a menüváltó gombok, a Base Materials az alapanyagok menüt, az Item menu a tárgyak menüt mutatja
 - Az alapanyagok menüben megjelenik egy receptnek legalapabb anyagai
 - A tárgyak menü tartalmazza a tárgykezelő gombokat

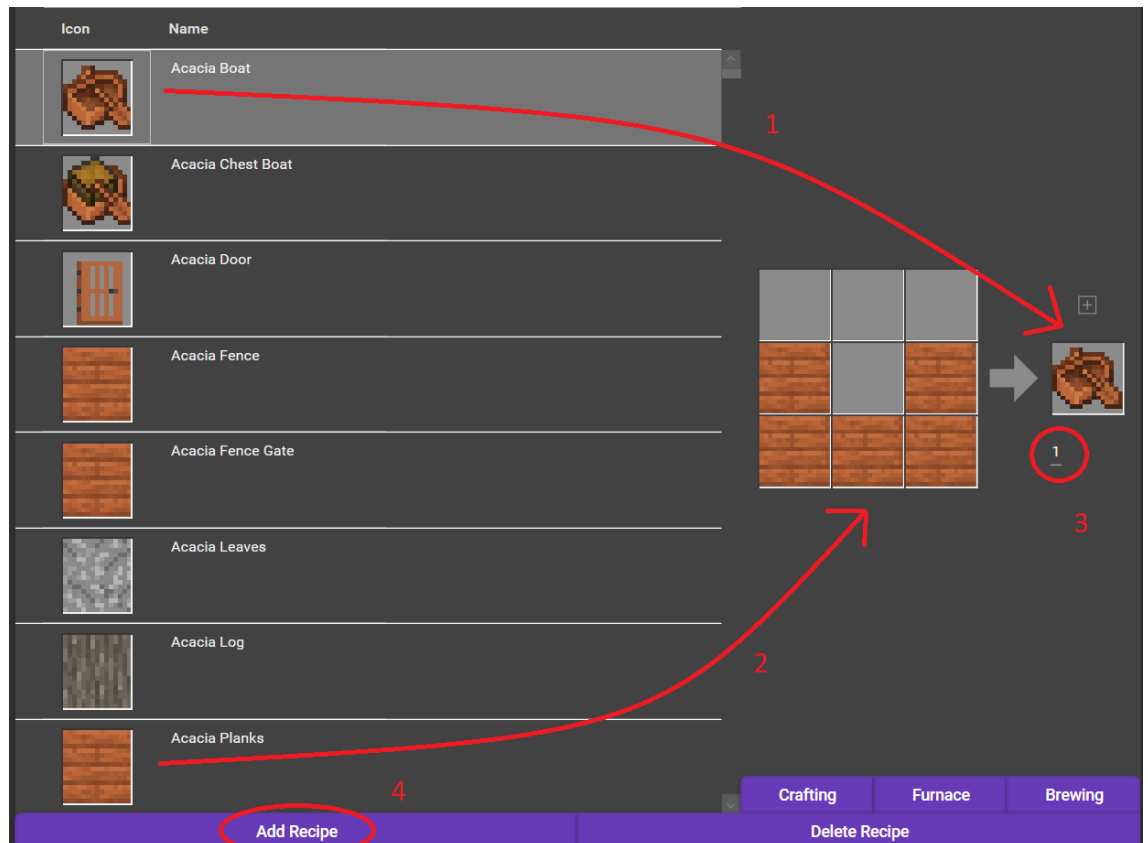


- Az Add Item gomb egy teljesen új tárgyat ad az adatbázishoz
- Az Alter Item megváltoztat egy létező tárgyat, amit kiválasztunk a tárgyak adatrácsban
- A Delete Item töröl egy kiválasztott tárgyat

5.4.2. A funkciók helyes használata

Egy Recept hozzáadása:

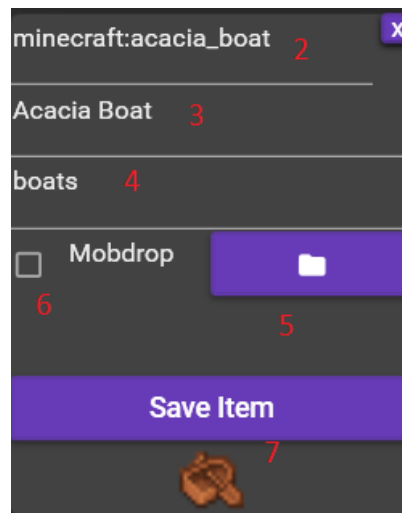
1. Megfogunk egy tárgyat a tárgyrácsból, amivel akarjuk asszociálni a receptet, és behúzzuk a résbe amire mutat a nyíl.
2. Megfogjuk és behúzzuk a tárgyakat a többi részbe, ahol akarjuk hogy legyen egy tárgy.
3. Beírjuk a mennyiséget, hogy mennyi tárgyat hozzon létre a megadott recept.
4. Megnyomjuk az „Add Recipe” gombot.



Egy tárgy helyes hozzáadása:

1. Amikor megnyomjuk az „Add Item” gombot, az feldob nekünk egy felugró ablakot.

2. Először beírjuk a tárgy ID-jét. Ez általában vagy a játék vagy a modifikáció nevével kezdődik, pl.: minecraft:acacia_boat vagy tconstruct:slime_boots.
3. Ezután megadjuk a hozzáadandó tárgy nevét.
4. Ha van egy vele asszociált típus, akkor ezt beírjuk ebbe a dobozba.
5. Hozzáadunk a tárgyhoz egy képet, a formátuma .png és a mérete 32*32 pixel.
6. Ha dobja egy lény, akkor ezzel hozzátudjuk írni a lénynek a nevét és a dobási esélyt.
7. Végül megnyomjuk a „Save Item” gombot.



6. Irodalomjegyzék

<https://www.minecraft.net/en-us>

<https://visualstudio.microsoft.com/vs/community/>

<https://www.monodevelop.com/>

<https://learn.microsoft.com/en-us/sql/database-engine/configure-windows/sql-server-express-localdb?view=sql-server-ver16>