Odisee Gent

Gebroeders de Smetstraat 1, 9000 Gent

MineColonies Automation

MineColonies Automation

Pjotr Brunain, Thibe Provost

Luca Vandenweghe, Jonas Van Kerkhove

Professionele bachelor Elektronica-ICT / Fase 2

2023-2024

Technische mentor: ma. Evert-Jan Jacobs

Taalmentor: ma. Sabine Martens

MineColonies Automation

Inhoud

Codefragmentenlijst

Figurenlijst

Tabellenlijst

Geen gegevens voor lijst met afbeeldingen gevonden.

Afkortingenlijst

MOD	Modificatie

Inleiding

Dit project bestaat erin om het beginproces van Minecolonies te vereenvoudigen en meer gebruiksvriendelijk te maken.

MineColonies is een MOD die een gemeenschap probeert na te bootsen in Minecraft. Bij deze MOD is de speler het hoofd van de gemeenschap en zorgt die ervoor dat alles in goede banen loopt. In het begin van dit proces is dit heel omslachtig omdat er veel tussen de verschillende gebouwen moet gelopen worden en er weinig overzicht is.
Om dit proces vlotter te laten verlopen is er een webapplicatie gemaakt waarin alle taken en inventaris van zowel de speler als de gemeenschap duidelijk zichtbaar zijn. Ook is het mogelijk om te beslissen of taken automatisch vervuld worden of dat deze door de speler geselecteerd moeten worden om vervuld te worden.

Om dit doel te verwezenlijken moet er een diepgaand onderzoek gebeuren rond de verschillende gebruikte MODS, de verschillende frameworks voor de website en de programmeertaal LUA.
Hierna moeten er verschillende programma's gemaakt worden, namelijk: de website frontend, de website backend, het C#-programma dat de link vormt tussen Minecraft en de website en de LUA-scripts die de nodige data uit Minecraft extraheert.

Vanaf hier gaat het rapport eerst over Minecraft en de gebruikte MODS en hoe deze gebruikt zullen worden om de nodige data te extraheren. Hierna gaan uitbreiden hoe het C# programma werkt waarna er verder uitgebreid wordt over de website.

1 Minecraft en MODS

1.1 Overzicht van de gebruikte Minecraft MODS

Voor dit project worden er een aantal MODS gebruikt bovenop Minecraft. Deze MODS zijn MineColonies, Applied Energistics 2, CC: Tweaked en Advanced Peripherals.

1.1.1 MineColonies

Deze MOD zorgt voor een simulatie waarbij de speler het stadshoofd is van een gemeenschap en die gemeenschap in goede banen moet leiden. [1]

Dit is de MOD die wordt gestroomlijnd.

1.1.2 Applied Energistics 2

Applied Energistics 2 is een MOD waarmee spelers items kunnen opslaan in een digitale opslag en deze items kunnen visualiseren op een mooie en begrijpelijke manier. Ook zorgt deze MOD voor het automatisch maken van items eens dat de speler het systeem dit heeft aangeleerd. [2]

1.1.3 CC: Tweaked

Deze MOD maakt het mogelijk om te programmeren in Minecraft. Dit gebeurt aan de hand van een speciale blok genaamd computers (Figuur 2) in het spel waarop er scripts geschreven kunnen worden in de LUA-taal. Deze blok kan communiceren met verschillende randapparaten om de basiswerking uit te breiden. [3]

Figuur 1: Advanced Computer [8]

1.1.4 Advanced Peripherals

Dit is een MOD die CC: Tweaked uitbreidt met verschillende extra randapparaten waaronder de Colony Integrator (Figuur 3) en de ME Bridge (Figuur 4). Respectievelijk zorgen deze voor communicatie met MineColonies en Applied Energistics 2. [4]

Figuur 2: Colony Integrator [9]

Figuur 3: ME Bridge [9]

1.2 LUA en Minecraft

CC: Tweaked gebruikt LUA om te interageren met Minecraft.

1.2.1 LUA

LUA is een scripttaal die bekend staat voor zijn snelheid en robuustheid. LUA wordt gebruikt omdat dit de scripttaal is die geïmplementeerd is door CC: Tweaked en Advanced Peripherals. [5]

1.2.2 LUA in CC: Tweaked

LUA in CC: Tweaked is een aangepaste versie van LUA waar een aantal van de functies van de base LUA niet in aanwezig zijn. Een lijst met welke features geïmplementeerd zijn en welke niet kan gevonden worden op https://tweaked.cc/reference/feature_compat.html [6]

Als basis voor alles wat met randapparaten te maken heeft, moeten deze randapparaten ingepakt worden zodat er toegang mogelijk is vanuit de code. Hiervoor is er een wrapPeripherals (Codefragment 1) script geschreven dat geïmplementeerd kan worden in andere scripts.

```
-- Initialize everything to detect the different peripherals

local function Initialize(monitorWriter)
    MonitorWriter = monitorWriter

    -- wrap all peripherals on each side
    local topPeripheral = peripheral.wrap("top")
    local bottomPeripheral = peripheral.wrap("bottom")
    local leftPeripheral = peripheral.wrap("left")
    local rightPeripheral = peripheral.wrap("right")
    local frontPeripheral = peripheral.wrap("front")
    local backPeripheral = peripheral.wrap("back")

    -- Check each peripheral against the peripheralTable
    local func = nil
    if not (topPeripheral == nil) then func = peripheralTable[peripheral.getType(topPeripheral)] end
    if not (func == nil) then func(topPeripheral, "top") end
    func = nil
    if not (bottomPeripheral == nil) then func = peripheralTable[peripheral.getType(bottomPeripheral)]
end
    if not (func == nil) then func(bottomPeripheral, "bottom") end
    func = nil
    if not (leftPeripheral == nil) then func = peripheralTable[peripheral.getType(leftPeripheral)] end
    if not (func == nil) then func(leftPeripheral, "left") end
    func = nil
    if not (rightPeripheral == nil) then func = peripheralTable[peripheral.getType(rightPeripheral)] end
    if not (func == nil) then func(rightPeripheral, "right") end
    func = nil
    if not (frontPeripheral == nil) then func = peripheralTable[peripheral.getType(frontPeripheral)] end
    if not (func == nil) then func(frontPeripheral, "front") end
    func = nil
    if not (backPeripheral == nil) then func = peripheralTable[peripheral.getType(backPeripheral)] end
    if not (func == nil) then func(backPeripheral, "back") end
    func = nil

    -- Check if the colonyIntegrator is in a colony, if not exit the program
    if not ColonyIntegrator.isInColony() then
        MonitorWriter.WriteLine("Block is not in a colony", Monitor)
        os.exit()
    end
```

Codefragment 1: Inpakken van randapparaten

1.3 Extraheren van takenlijst uit Minecolonies

De taken worden opgesplitst in 2 soorten taken. De bouwertaken en de algemene taken die niet van de bouwers komen.

1.3.1 Extraheren algemene taken

Voor het extraheren van algemene taken worden eerst alle taken geëxtraheerd, hierna worden de taken van de bouwers eruit gefilterd. (Codefragment 2)

```
local requestData = {}

    -- Add the colony name in the data
    requestData["Name"] = peripherals.GetColonyIntegrator().getColonyName()
    -- Add all requests into the data
    requestData["Requests"] = peripherals.GetColonyIntegrator().getRequests()
    -- Remove all requests that are comming from Builders from the requests table
    for i=#requestData["Requests"],1,-1 do
        if not (string.find(requestData["Requests"][i]["target"], "Builder") == nil) then
            table.remove(requestData["Requests"], i)
        end
    end
```

Codefragment 2: Extractie, algemene requests

Dit wordt gedaan door in de taak naar de target te kijken en als er “Builder” in staat deze uit de lijst te verwijderen.

1.3.2 Extraheren bouwertaken

Voor het extraheren van de bouwertaken moeten eerst de verschillende gebouwen opgehaald worden. Hierna wordt er over deze lijst gegaan en als het gebouw een bouwer is dan worden de specifieke taken van de builder geëxtraheerd. Ook wordt de naam van de builder meegegeven. (Codefragment 3)

```
    local buildings = peripherals.GetColonyIntegrator().getBuildings()

    -- For each building check if it is a builder. if so, get the resources from that builder (this is
more then the normal requests above)
    for i, building in ipairs(buildings) do
        if not (string.find(building["name"], "builder") == nil) then
            local builder = {}
            -- Add builder name if it has one
            if not (building["citizens"][1] == nil) then
                builder["name"] = building["citizens"][1]["name"]
                monitorWriter.WriteLine("Extracting requests from builder: " .. builder["name"],
peripherals.GetMonitor())
            else
                monitorWriter.WriteLine("Extracting requests from unknown builder.",
peripherals.GetMonitor())
            end
            -- Add builder location
            builder["location"] = building["location"]
            builder["Requests"] =
peripherals.GetColonyIntegrator().getBuilderResources(building["location"])
            table.insert(builderRequests, builder)
        end
    end
```

Codefragment 3: Extractie, bouwertaken

1.4 Inlezen en uitvoeren van gegeven commando's

Commando's worden door het C# programma gegenereerd en uitgeschreven naar een JSON-formaat. (Codefragment 4)

```
[
 {
  "Item": "minecraft:cobblestone",
  "Amount": 11,
  "NeedsCrafting": false
 },
 {
  "Item": "minecraft:cobblestone_slab",
  "Amount": 16,
  "NeedsCrafting": true
 }
]
```

Codefragment 4: Commando voorbeeld

Deze commando's worden hierna ingelezen door het LUA-programma en uitgevoerd.

Om deze commando's in te lezen wordt er eerst gecheckt of de NeedsCrafting variabele true is. Als dit zo is dan wordt het gevraagde item automatisch gemaakt, anders wordt het item geëxporteerd van het systeem aan de speler kant naar het systeem aan de colony kant.

Hierna wordt elk commando dat uitgevoerd werd verwijderd zodat deze commando's niet meerdere keren na elkaar uitgevoerd worden.

2 C#-programma

2.1 Materiaalcalculator

2.2 Extractie van recepten uit Minecraftbestanden

2.3 Integratie met JSON-data

3 Websiteontwikkeling

3.1 Ontwerp met Figma

3.2 Frontendontwikkeling

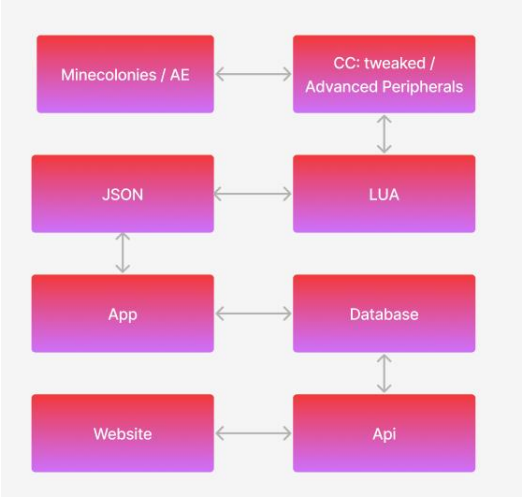
3.3 Backendontwikkeling

4 Samenhang tussen web en app

4.1 Webintegratie van de app

4.2 Samenhang en communicatie tussen API en C#

Om de gegevens van Minecraft efficiënt te kunnen verwerken wordt er gebruik gemaakt van een Web API waarnaar er request worden gestuurd vanuit de C# applicatie. Er word een onderscheid gemaakt tussen postrequests en getrequests (**Error! Reference source not found.**).



Figuur 4: Dataflowchart

4.2.1 POST requests

4.2.1.1 Werelden:

Werelden worden aangemaakt op het endpoint: “***/api/worlds”***. Om een wereld aan te maken dient in de requestbody de naam van de werled meegegeven te worden (Codefragment 5).

```
{
    "name": "C:-Users-Testuser-curseforge-minecraft-Instances-MineColonies
integration-saves-test world"
}
```

Codefragment 5: Postworld, requestbody

Na een succesvolle postrequest en nadat de wereld wordt aangemaakt, wordt er statuscode 201 teruggestuurd. In het geval dat de wereld al bestond wordt een statuscode van 400 teruggestuurd.

4.2.1.2 Colonies:

Nadat de werelden zijn aangemaakt kunnen er colonies op het endpoint: “***/api/colonies”*** aangemaakt worden. Om een colonie te maken dient in de requestbody de naam en de wereld id meegegeven te worden (Codefragment 6).

```
{
    "name": "ikecolonie",
    "world_id": 2
}
```

Codefragment 6: Postcolonie, requestbody

Na een succesvolle postrequest en nadat de colonie wordt aangemaakt, wordt er een statuscode 201 teruggestuurd. In het geval dat de colonie al bestond wordt een statuscode van 400 teruggestuurd.

4.2.1.3 Requests:

Nadat er een colonie aangemaakt is kunnen er request op het endpoint, “***/api/colonies/requests***”, aangemaakt worden. De endpoint verwacht een lijst van requestobjecten. Een requestobject heeft naast de data van een request ook de naam van de colonie en de wereld id om deze requests toe te voegen aan de database (Codefragment 7).

```
[

    {
        "colonies_id": 4,
        "fingerprint": "47e648c7-09b6-4f24-b497-0ea07d19e398",
        "count": 1,
        "name": "Chestplate",
        "desc": "Chestplate with minimal level: Leather and with maximal level:
Gold",
        "id": "47e648c7-09b6-4f24-b497-0ea07d19e398",
        "minCount": 1,
        "state": "IN_PROGRESS",
        "target": "Archer Case B. McGee",
        "items": [
            {
                "count": 1,
                "displayName": "[Golden Chestplate]",
                "fingerprint": "E969C47F2E35EC0378D7D0D55EE412C4",
                "maxStackSize": 1,
                "name": "minecraft:golden_chestplate",
                "nbt": {
                    "Damage": {
                        "ValueKind": 4
                    }
                },
                "tags": [
                    "minecraft:item/forge:armors/chestplates",
                    "minecraft:item/forge:armors",
                    "minecraft:item/minecraft:trimmable_armor",
                    "minecraft:item/minecraft:piglin_loved"
                ]
```


Codefragment 7: Postrequests, requestbody

4.2.1.4 Builderrequests:

Nadat er een colonie aangemaakt is kunnen er builderrequests op het endpoint, "***/api/colonies/builderrequests***", aangemaakt worden. De endpoint verwacht een lijst van builderrequestsobjecten. Een builderrequestsobject heeft naast de data van een builderrequests een colonie-id om deze builderrequests toe te voegen aan de database (Codefragment 8).

```
[
    {
        "colonies_id": 5,
        "location": {
            "x": 359,
            "y": 67,
            "z": -191
        },
        "name": "Frankie T. Claybrook",
        "Requests": [
            {
                "available": 0,
                "delivering": 0,
                "displayName": "Cobblestone",
                "item": {
                    "count": 1,
                    "displayName": "[Cobblestone]",
                    "fingerprint": "34C0EF2C2CD360FB88A0AD2798E4354D",
                    "maxStackSize": 64,
                    "name": "minecraft:cobblestone",
                    "nbt": {},
                    "tags": [
                        "minecraft:item/minecolonies:reduceable_product_excluded"
,
                        "minecraft:item/minecolonies:blacksmith_product_excluded"
                    ]
                },
                "needed": 11,
                "status": "DONT_HAVE"
            }
        ]
    }
]
```


Codefragment 8: Postbuilderrequests, requestbody

4.2.1.5 Storage items:

(bestaat nog niet)

4.2.2 GET requests

4.2.2.1 Werelden

Om te beslissen of een wereld moet toegevoegd worden, wordt er eerst gecheckt of deze wereld al bestaat in de database. Als returnbody krijgt het programma een JSON-object terug (Codefragment 9).

```
{
    "data": {
        "id": 4,
        "name": "C:-Users-Jonas-curseforge-minecraft-Instances-MineColonies
integration-saves-test world v2",
        "created_at": "2024-05-07T09:10:10.000000Z",
        "updated_at": "2024-05-07T09:10:10.000000Z",
        "colonies": [
            {
                "id": 4,
                "name": "test",
                "autocomplete": false,
                "autoArmor": false,
                "autoTools": false
            }
        ]
    }
}
```


Codefragment 9: Getwereld, responsebody

4.2.2.2 Colonies

Om te beslissen of een colonie moet toegevoegd worden, wordt er eerst gecheckt of deze colonie al bestaat in de database. Als returnbody krijgt het programma een JSON-object terug (Codefragment 10).


```
{

    "data": {
        "id": 4,
        "name": "test",
        "autocomplete": false,
        "autoArmor": false,
        "autoTools": false,
        "builderRequests": [
            {
                "id": 1,
                "name": "jeff",
                "autocomplete": true,
                "location": {
                    "x": 1,
                    "y": 2,
                    "z": 54
                },
                "colonies_id": 4,
                "created_at": "time"
            }
        ]
    }
```


Codefragment 10: Getcolonie, responsebody

Conclusie

Handleiding

Literatuurlijst

[1] W. Mark Fisher, „wiki.minecolonies.ldtteam,” MineColonies, 1 1 2021. [Online]. Available: https://wiki.minecolonies.ldtteam.com/. [Geopend 25 03 2024].

[2] „https://guide.appliedenergistics.org/1.20.1/index,” Applied Energistics 2, 1 1 2024. [Online]. Available: https://guide.appliedenergistics.org/1.20.1/index. [Geopend 25 03 2024].

[3] „tweaked.cc,” tweaked.cc, 25 03 2024. [Online]. Available: https://tweaked.cc/. [Geopend 24 03 2024].

[4] „advanced-peripherals.de,” SyntheticDev, [Online]. Available: https://docs.advanced-peripherals.de/. [Geopend 25 03 2024].

[5] D. o. C. S. o. PUC-Rio, „lua.org,” Lua.org, 18 02 2024. [Online]. Available: https://www.lua.org/about.html. [Geopend 25 03 2024].

[6] cc: Tweaked, [Online]. Available: https://tweaked.cc/reference. [Geopend 5 March 2024].

[7] „json.org,” json.org, 12 1999. [Online]. Available: https://www.json.org/json-en.html. [Geopend 25 03 2024].

[8] Feed The Beast, „ftbwiki.org,” [Online]. Available: https://ftbwiki.org/File:Block_Advanced_Computer.png. [Geopend 19 March 2024].

[9] SirEndii, „Advanced Peripherals Wiki,” [Online]. Available: https://docs.advanced-peripherals.de. [Geopend 19 March 2024].

Bijlagenoverzicht

Bijlage 1: Logboek

Week1:

Tijdens de eerste week hebben wij ons bezig gehouden met het opstarten en voorbereiden van de vergadering met onze technische mentor. We kwamen samen om onze Gitlab (https://gitlab.com/ikdoeict/thibe.provost/project-mincraft)omgeving op te zetten en iedereen de nodige rechten te geven. Zo hebben we ons probleem opgedeeld in verschillende kleinere deelproblemen.

We hebben tijdens deze 1ste week ook een moment vastgelegd om te vergaderen met onze mentor. De vergadering zal doorgaan 28/02/2024 om 9u, het kanaal dat we gebruiken voor deze vergadering is Microsoft Teams. Thibe Provost zal verantwoordelijk zijn voor het maken van het verslag.

Week2:

In het begin van deze week zijn we begonnen met het logboek en rapport op te maken. Thibe Provost begon samen met Jonas Van Kerkhove de koppen en kaft van het rapport op te maken. Terwijl Luca Vandenweghe samen met Pjotr Brunain aan de inleiding begon. Thibe heeft thuis nog het logboek aangezet en de layout van het rapport afgewerkt alsook de mail gemaakt.

Week3-4:

Tijdens de 3e en 4e week hebben niet aan dit logboek gewerkt , maar hebben we ons gefocust op het uitwerken van kleine demo’s en manieren om het project te realiseren.

Week5:

In de 5e week heeft Thibe Provost alle fouten uit het document gehaald aan de hand van de gegeven feedback behalve bij de inleiding. De inleiding hebben Pjotr Brunain, Thibe Provost en Jonas Van Kerkhove volledig herschreven. Hierna hebben Pjotr Brunain en Jonas Van Kerkhove gewerkt aan het eerste hoofdstuk, LUA en Minecraft, van het rapport. Luca Vandenweghe heeft alles nagelezen, fouten uitgehaald en bepaalde stukken veranderd waar nodig.

Week6-9:

Tijdens deze periode heeft Thibe Provost de overbodige koppen uit de inhoudstafel gehaald en het logboek aangevuld. Hij heeft ook de verslagen van de vergaderingen toegevoegd aan het rapport. Pjotr Brunain heeft alle feedback tijdens het gesprek met de taalmentor genoteerd en heeft deze samen met Luca Vandenweghe verbeterd. Jonas Van Kerkhove heeft samen met Thibe Provost en Luca Vandeweghe een extra hoofdstuk geschreven.

Bijlage 2: Verslag1

28/02/2024

Aanwezigen: Pjotr Brunain, Luca Vandeweghe, Jonas Van Kerkhove, Evert-Jan Jacobs, Thibe Provost.

Afwezigen: /

Wat hebben we allemaal al gedaan?

We hebben ter voorbereiding op deze vergaderingen een subgroup op gitlab.com gemaakt. Hierin hebben we alvast 3 projecten aangemaakt: één voor alle documentatie, één voor de website en één voor de applicatie. In ieder project maakten we ook meerdere issues aan zodat wij een goed overzicht konden krijgen van alle openstaande opdrachten.
Het logboek en het rapport werden aangemaakt voor de vergadering en zal verder worden aangevuld naarmate de opdracht word uitgevoerd.

Wat kan beter?

Waar we zeker op moeten letten in de toekomst is het nalezen documenten op schrijffouten en zinsstructuur. Het issue board op gitlab.com is ook nog steeds niet 100% volledig. Dit zouden moeten proberen zo volledig mogelijk te maken zodat we een goed overzicht krijgen over de openstaande problemen. Ook de communicatie tussen de mentor en teamleden om een datum vast te leggen zou beter kunnen. Als groep zouden we beter een paar datums afspreken dat wij kunnen en die voorleggen aan de mentor. De voorbereiding op de vergadering zelf kan vlotter verlopen. Voor de volgende keer kunnen we best al eens op voorhand samenzitten en nadenken wat er de volgende 2 weken moet gebeuren.

Wat moet er nog gebeuren?

- Een schema maken die visueel de samenwerking tussen de verschillende delen van ons project voorstelt.
- Het logboek moet eens nagelezen worden en alle fouten moeten eruit gehaald worden.
- Logboek aanvullen over week2.
- Verslag lay-out maken voor volgende vergadering.
- Kleine demo maken voor volgende vergadering .
- Issue board vervolledigen.
- Een extra label extra’s maken om features bij te houden die tof zijn als uitbreiding. Bv: Prioriteiten aanpassen van de verschillende taken, kijken om het project in een server omgeving werkend te krijgen.
- Brainstormen over front-end.

Wat moet zeker gebeuren tijdens deze sprint?

- Het logboek aanvullen.
- Nadenken over een demo.
- Brainstormen over front-end.

- Nadenken over wat we de volgende 2 weken gaan doen.
- Uitnodiging naar mentor sturen voor volgende vergadering.

Tips van Evert-Jan

- Probeer het zo simpel mogelijk uit te leggen.
- Documenteer alles zodat iemand jullie project kan overnemen of als er een vervolgproject zou volgen jullie verder kunnen werken.
- Maak al een voorbereiden verslag zodat jullie weten wat jullie tijdens de meeting willen tonen.

Bijlage 3: Verslag2

13/03/2024

Aanwezigen: Pjotr Brunain, Luca Vandeweghe, Jonas Van Kerkhove, Evert-Jan Jacobs, Thibe Provost.

Afwezigen: /

Wat hebben we gedaan?

Thibe had een eerste ruw databank design gemaakt en ondertussen alweer wat verfijnd door Pjotr en Jonas. Ook heeft Jonas al een werkende demo gemaakt waarin we de data uit een voorbeeld json bestand halen. Pjotr, Jonas en Luca maakten een schema die visueel de samenwerking tussen de verschillende delen van ons project voorstelt. Ook langs de Minecraft kant van ons project heeft Pjotr al wat werk verricht. Zo wordt er al data geëxporteerd aan de hand van een paar LUA-scripts. Ook voor de website heeft Luca er al eens nagedacht over een design. Het issue bord werd uitgebreid met een “Need Feedback” kolom door Thibe en extra issues. Thibe heeft ook wat probleempjes in Gitlab opgelost.

Demo’s

- Korte demo van het uitlezen van een Json bestand (Jonas)
- Korte demo LUA (pjotr)
- Kort voorstellen van schema

Wat kan beter?

- Demo’s zijn goed gekeurd maar structureer het schema net iets beter
- Alle velden met kleine letters zetten in databank (Fk-velden)
- Probleem id dat hoog oploopt → Misschien een andere unieke key (misschien de id uit de Json file)
- Commentaar bij LUA

- Markdown bestandje maken op git

Wat doen we tegen volgende sprint?

- Get Items
- Db connectie
- Patterns aka recipies
- Extracten crafting recipies
- Final db shema
- Finish Figma design
- First step of website
    - Homepage en login

Tips van Evert-Jan

- Maak een kalender/ planning voor wie wat doe tot het einde van het project → via roadmap op git
- Wat gaan we tonen op het einde van de opdracht? Wat is ons eindproduct

Bijlage 4: Verslag3

27/03/2024

Aanwezigen: Pjotr Brunain, Luca Vandeweghe, Jonas Van Kerkhove, Evert-Jan Jacobs, Thibe Provost.

Afwezigen: /

Wat hebben we gedaan?

Tijdens deze sprint hebben Jonas en Pjotr samen met een beetje hulp van Thibe de inleiding herschreven. Thibe heeft aan de hand van de gegeven feedback het rapport verbeterd. Pjotr heeft een hoofdstuk geschreven met veel feedback van Jonas. Jonas heeft ook nog verder aan de wpf-applicatie, hij heeft de items van een request uit een Json file gehaald. Ook heeft hij dit gelinkt aan de correcte colonie en het in een mooie Gui gepresenteerd. Pjotr heeft geluisterd naar Evert-Jan en commentaar bij zijn LUA code geschreven. Hij hielp ook jonas met de wpf-applicatie door te beginnen aan de file selection . Hij heeft ook in Lua een script geschreven om de items uit te lezen en een epic en roadmap op git gemaakt. Thibe heeft dezelfde Json files omgezet in een Laravel project. Hij heeft migrations en seeders gemaakt om de databank te maken. Door gebrek aan tijd is het hem niet gelukt om de apiroutes te maken. Luca heeft de volledige website gemaakt en figma afgewerkt. Hij heeft zowel een mobile als desktopversie gemaakt en meerdere css animaties.

Demo’s

Website → Luca

Wpf demo → Pjotr

Laravel → Thibe

Wat kan beter?

- 2 nav bars website
- Complexiteit website
- Automatish genereren seeders?
- Mooi maken wpf?

Wat doen we tegen volgende sprint?

- Zie Roadmap Git

Tips van Evert-Jan

- Overzicht waar we willen eindigen
- Wanneer klaar?
- Dashboard
- Denk na over verhaal → maak filmpje/ demo / iets
- Pjotr dacht om een aparte computer de demo te doen een eigenlijk live demo

Bijlage 5: Verslag4

17/04/2024

Aanwezigen: Pjotr Brunain, Luca Vandeweghe, Jonas Van Kerkhove, Evert-Jan Jacobs, Thibe Provost.

Afwezigen: /

Wat hebben we gedaan?

Tijdens deze sprint heeft Thibe gewerkt aan de api. Hij heeft al een post en get endpoint voor users, world, colonies en een get voor builderrequests met zijn location en specifiedrequest. Hij was ook beginnen met een delete maar dat bleek niet nodig. Door de sprintlabo's van web en andere verplichtingen heeft hij nog niet verder kunnen werken. Pjotr heeft commands in Lua werkend gekregen en de wrap peripherals script herwerkt hiervoor. Ook heeft hij een main loop voor het lua programma geimplementeerd en een savestate voor lua geimplementeerd. Jonas heeft Thibe erop gewezen dat de databank moest aangepast worden en hij heeft requests omgezet naar commands die aan lua worden overgedragen. Luca is nuxt beginnen leren. Zijn eerste indruk was : "wtf da is ". Ook heeft hij een nuxt project aangemaakt en is hij begonnen aan de transfer van onze website naar nuxt framework.

Demo's

Lua → Pjotr

Laravel api → Thibe

Wat kan beter?

- Verslag van wetenschappelijk rapporteren (toch meer in detail)
- Location <-> builderRequest een op een maken

Wat doen we tegen volgende sprint?

- Zie Roadmap Git
- Api af en al wat request op de website krijgen
- Het hapbaar maken van ons idee/project

Tips van Evert-Jan

- Eens samen met evert-jan samenzitten met verslag
    - Code fragmenten uitleggen
- Zorg voor iets werkend op den opendeurdag
- Maak het hapbaar

Bijlage 6: Verslag5

8/05/2024