



Odisee Gent

Gebroeders de Smetstraat 1, 9000 Gent

MineColonies Automation

MineColonies Automation

Pjotr Brunain, Thibe Provost

Luca Vandenweghe, Jonas Van Kerkhove

MineColonies Automation

Inhoud

Codefragmentenlijst.....	7
Figurenlijst.....	8
Tabellenlijst	9
Afkortingenlijst.....	10
Inleiding.....	11
1 Minecraft en mods	12
1.1 Overzicht van de gebruikte Minecraft mods.....	12
1.1.1 MineColonies.....	12
1.1.2 Applied Energistics 2.....	12
1.1.3 CC: Tweaked	12
1.1.4 Advanced Peripherals	12
1.2 LUA en Minecraft	13
1.2.1 LUA	13
1.2.2 LUA in CC: Tweaked	13
1.3 Extraheren van takenlijst uit Minecolonies	14
1.3.1 Extraheren algemene taken	14
1.3.2 Extraheren bouwer taken.....	14
1.4 Inlezen en uitvoeren van gegeven commando's.....	15
2 C#-programma	15
2.1 Materiaalcalculator.....	15
2.2 Extractie van recepten uit Minecraftbestanden	15
2.3 Integratie met JSON-data	15
3 Websiteontwikkeling	15
3.1 Ontwerp met Figma	15
3.2 Frontendontwikkeling	15
3.3 Backendontwikkeling.....	15
4 Samenhang tussen web en app	15
4.1 Webintegratie van de app.....	15
4.2 Samenhang en communicatie tussen componenten.....	15
Conclusie	16
Handleiding	17
Literatuurlijst	18
Bijlagenoverzicht.....	19
Bijlage 1: Logboek	19

Week1:	19
Week2:	19
Week3-4:	19
Week5:	19
Bijlage 2: Verslag1	20
Wat hebben we allemaal al gedaan?	20
Wat kan beter?	20
Wat moet er nog gebeuren?	20
Wat moet zeker gebeuren tijdens deze sprint?	20
Tips van Evert-Jan	21
Bijlage 3: Verslag2	21
Wat hebben we gedaan?	21
Demo's	21
Wat kan beter?	21
Wat doen we tegen volgende sprint?	22
Tips van Evert-Jan	22

Codefragmentenlijst

Codefragment 1: Inpakken van randapparaten	13
Codefragment 2: Algemene request extractie.....	14
Codefragment 3: Bouwer taken extractie.....	14

Figurenlijst

Figuur 1: Dataflow Chart.....	11
Figuur 2: Advanced Computer.....	12
Figuur 3: Colony Integrator.....	12
Figuur 4: ME Bridge.....	12

Tabellenlijst

Geen gegevens voor lijst met afbeeldingen gevonden.

Afkortingenlijst

Inleiding

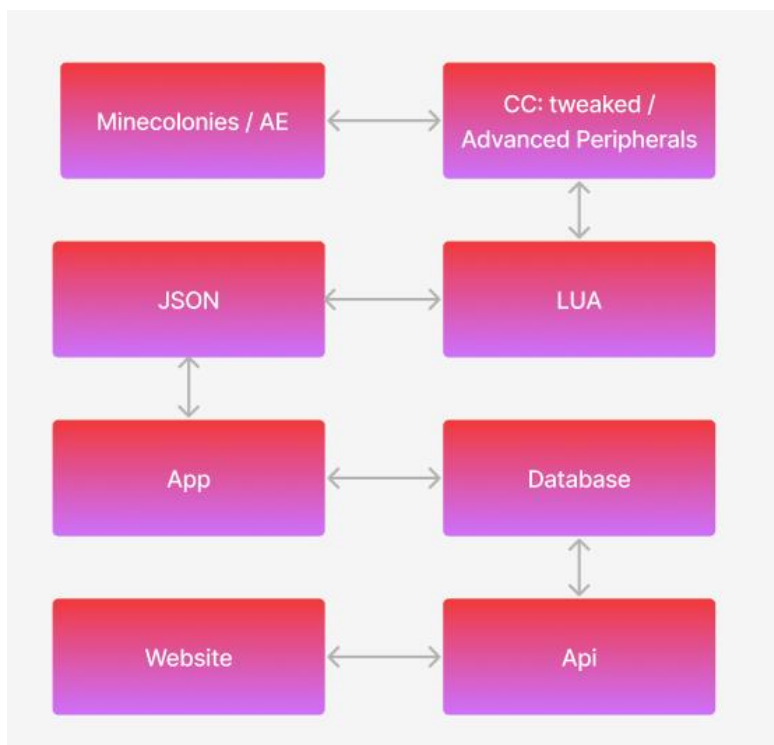
Dit project bestaat erin om het beginproces van Minecolonies te vereenvoudigen en meer gebruiksvriendelijk te maken.

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 zal er een webapplicatie gemaakt worden waarin alle taken en inventaris van zowel de speler als de gemeenschap duidelijk zichtbaar zijn. Ook zal het mogelijk zijn 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 zal de mod CC: Tweaked gebruikt worden om de taken en items in een JSON-formaat te extraheren. Hierna wordt deze informatie in een overzichtelijke webapplicatie gegoten om alles duidelijker te maken voor de speler. Hierbij wordt er ook verdere verwerking van deze data gedaan om de basiselementen uit elke taak te halen.

De communicatie tussen de webapplicatie en Minecraft zal over een aantal stappen gaan zoals getoond in (Figuur 1)



Figuur 1: Dataflow Chart

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 we wensen te stroomlijnen.

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 je scripts kunt schrijven in de LUA-taal. Deze blok kan communiceren met verschillende randapparaten om de basiswerking uit te breiden. [3]



Figuur 2: Advanced Computer

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 zullen deze zorgen voor communicatie met MineColonies en Applied Energistics 2. [4]



Figuur 3: Colony Integrator



Figuur 4: ME Bridge

1.2 LUA en Minecraft

CC: Tweaked gebruikt LUA om te interageren met Minecraft.

1.2.1 LUA

LUA is een scripttaal dat 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

Als basis voor alles dat met randapparaten te maken heeft, moeten deze randapparaten ingepakt worden zodat we er toegang toe hebben vanuit de code. Hiervoor is er een wrapPeripherals (Codefragment 1) script geschreven dat kan 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
end
```

Codefragment 1: Inpakken van randapparaten

1.3 Extraheren van takenlijst uit Minecolonies

De taken worden opgesplitst in 2 soorten taken. De bouwer taken 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 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 coming 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: Algemene request extractie

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

1.3.2 Extraheren bouwer taken

Voor het extraheren van de bouwer taken moeten eerst de verschillende gebouwen opgehaald worden. Hierna wordt er over deze lijst gegaan en als het gebouw een bouwer is dan extraheren we de specifieke taken dat deze builder heeft. 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(builder["location"])
        table.insert(builderRequests, builder)
    end
end
```

Codefragment 3: Bouwer taken extractie

1.4 Inlezen en uitvoeren van gegeven commando's

De commando's zullen in JSON-formaat gegenereerd worden door het C#-programma waarna dit ingelezen zal worden door het LUA-programma. Dit is momenteel nog niet geïmplementeerd

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 componenten

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] „json.org,” json.org, 12 1999. [Online]. Available: <https://www.json.org/json-en.html>. [Geopend 25 03 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 1^{ste} 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 kft 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 3^e en 4^e 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 5^e 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.

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

