

# Gamification van een opleiding

---

**Opleiding: Electronica-ICT**

**Academiejaar: 2018-2019**

Sam Van Bogaert

Promotor: Cindy Braspenning

Stagebegeleider: Serge Horsmans

# Voorwoord

Alvorens zou ik graag iedereen bedanken die mij geholpen heeft bij het verwezelijken van het uiteindelijke resultaat van dit project.

Mijn stagebegeleider Cindy Braspenning die mij begeleid heeft tijdens de ontwikkeling van dit project en die mij wekelijks constructieve feedback gaf dat ik telkens kon verwerken in het project.

Voor technische vragen over Unity kon ik steeds terecht bij Geert Verreet die mij steeds geholpen heeft om op de juiste manier te werk te gaan.

Voor andere kennis, technische vragen of extra begeleiding kon ik steeds terecht bij Thomas De Witte.

Ook zou ik graag de mensen van Ferranti bedanken om mij een kans te geven om het project tot een goed einde te brengen.

# 1 Samenvatting

MeCity is een spel dat ontworpen is als uitbreiding of vervanging van de opleiding die gegeven wordt aan de trainees van Mecoms. De originele opleiding bestond uit een korte, algemene samenvatting alsook een specifiekere opleidingsonderdeel over een bepaalde partij uit de energiesector (producent, TGO, DGO, leverancier en consument).

De huidige opleiding gaf vaak een saaie blik op de energiesector en bestaat enkel uit pdf bestanden dat de gebruiker moet doorlopen. Als oplossing kwam de Solution Architect, Mevr. Cindy Braspenning op het idee om al die informatie om te vormen van pdf's naar een spel dat ongeveer dezelfde informatie zou toedienen aan de speler. Dit spel zou dan als uitbreiding kunnen dienen op de huidige opleiding.



*Figuur 1 - logo Mecity*

## 2 Inhoudstafel

Gamification van een opleiding.....	1
Voorwoord .....	1
1 Samenvatting.....	2
2 Inhoudstafel .....	3
3 Lijst van figuren .....	6
4 Samenvatting project inhoud .....	7
4.1 Voor aanvang van de stage.....	7
4.1.1 Log-in scherm .....	7
4.1.2 Hoofd menu .....	7
4.1.3 Levels menu .....	7
4.1.4 Megan introductie .....	7
4.1.5 Spel introductie .....	8
4.1.6 Highscores menu.....	8
4.1.7 Leverancier level.....	8
4.2 Na afloop van de stage.....	8
5 Uitvoering .....	9
5.1 Project Management .....	9
5.2 Version Control.....	10
5.3 Game Development.....	10
5.3.1 Prioriteit.....	10
6 Ondervonden problemen en oplossingen .....	11
6.1 Data opslaan via XML .....	11
6.1.1 PlayerPrefs .....	11
6.1.2 JSON .....	11
6.1.3 Binary.....	11
6.1.4 XML .....	11
6.2 Converteren XML structuur van de vragen .....	12
6.3 Implementeren van encryptie en decryptie .....	12
6.4 Sturen van e-mail bij loggen van reports, suggestions ... ..	12
7 Handleiding.....	13
7.1 Algemeen.....	13

7.1.1	User interface .....	14
7.1.2	Pauze menu.....	14
7.1.3	Quiz & Score.....	14
7.2	Producer level.....	15
7.3	TGO level.....	16
7.3.1	Memory.....	16
7.3.2	Breakout .....	16
7.3.3	Simon says .....	17
7.3.4	Mastermind.....	17
7.3.5	Hangman .....	18
7.3.6	Catcher .....	18
7.4	DGO level .....	19
7.5	Supplier level .....	20
7.6	Consumer level.....	21
7.7	Mecom's level.....	22
8	Support pagina .....	23
8.1	Probleem rapporteren.....	23
8.1.1	Code: toevoegen en verwijderen van een report .....	24
8.2	Suggesties maken .....	25
8.3	Vragen toevoegen .....	26
8.4	Admin pagina.....	27
9	Project structuur .....	28
9.1	Doc .....	28
9.1.1	Nieuwe documentatie .....	28
9.1.2	Originele documentatie.....	28
9.2	Src .....	29
9.2.1	Mecity project.....	29
10	How to's .....	36
10.1	Vragen toevoegen of aanpassen.....	36
10.2	Variabelen aanpassen.....	36
10.3	Levels toevoegen .....	36
10.3.1	Introductie toevoegen .....	36
11	Uitbreidings mogelijkheden.....	37
11.1	Support pagina .....	37
11.2	Highscores.....	37

11.3	Encryptie & decryptie.....	37
11.4	Verschillende talen toevoegen .....	37
12	Bibliography .....	38

### 3 Lijst van figuren

Figuur 1 - logo Mecity .....	2
Figuur 2 - Megan .....	7
Figuur 3 - flow van het spel .....	13
Figuur 4 – user interface: probleem rapporteren .....	23
Figuur 5 - code: toevoegen en verwijderen van een report.....	24
Figuur 6 - code: report confirmeren.....	24
Figuur 7 – user interface: suggestie maken .....	25
Figuur 8 – user interface: kiezen van level.....	26
Figuur 9 – user interface: invullen van vraag & antwoorden .....	26

## 4 Samenvatting project inhoud

### 4.1 Voor aanvang van de stage

Doordat het project al reeds in ontwikkeling was door enkele jobstudenten, waren er al enkele zaken aanwezig voor aanvang van de stage periode. Hieronder staan deze zaken kort opgelijst met telkens een kleine descriptie over de inhoud van dat onderdeel.

- Log-in scherm
- Hoofdmenu
  - Play button
  - Levels button
  - Highscore button
  - Exit button
- Levels Menu
- Megan Introductie
- Spel Introductie
- Highscores menu
- Leverancier level

#### 4.1.1 Log-in scherm

Hierin werd het logo van het spel weergegeven en kon men een gebruikersnaam opgeven dat moest voldoen aan enkele voorwaarden. (bv. Inputfield mocht niet leeg zijn, geen speciale tekens ...)

#### 4.1.2 Hoofd menu

In dit menu had de speler verschillende opties. De speler had de optie om de 'Megan' introductie te starten, om het levels menu te bezoeken, om de highscores van andere spelers te bekijken in het highscores menu of om het spel af te sluiten.

#### 4.1.3 Levels menu

In dit menu kreeg de speler een overzicht van alle levels die in het spel aanwezig waren of nog moesten geïmplementeerd worden. De speler had wel enkel de keuze om de speelbare levels te bezoeken.

#### 4.1.4 Megan introductie

De 'Megan' Introductie is een korte introductie dat gegeven wordt door Megan, de mascotte van het spel. Megan geeft in deze introductie een korte uitleg over hoe men op het idee is gekomen om van MeCity een werkelijkheid te maken en wie er aan het project begonnen is. Ook werd er een korte uitleg gegeven over hoe men de camera kon besturen om door de stad van het spel te navigeren.



Figuur 2 - Megan



### 4.1.5 Spel introductie

In deze introductie level werden alle partijen uit de energiesector kort voorgesteld aan de speler. Zo kreeg de speler al een eerste indruk van wat hem te wachten stond in de rest van het spel. Nadat de speler de introductie had doorlopen, kreeg de speler in deze versie van het spel de optie om de introductie nog eens te doorlopen, om de leverancier level te starten of om naar het hoofd menu te gaan.

### 4.1.6 Highscores menu

In dit menu kreeg de speler een soort 'aquarium' te zien met verschillende vissen van verschillende groottes. Deze vissen stelden dan spelers voor en hoe hoger de score van de speler was, hoe groter de vis in het aquarium. Ook had de speler de optie om het aquarium om te vormen tot een lijst waarbij de speler met de hoogste score bovenaan de lijst staat en de speler met de laagste score onderaan de lijst staat.

### 4.1.7 Leverancier level

In deze level moest de speler in de rol van de leverancier kruipen en zoveel mogelijk klanten te winnen bij de concurrentie en ervoor zorgen dat de energie in balans blijft. (niet te veel onnodige energie aankopen, maar ook niet te weinig zodat de stad niet zonder elektriciteit valt.)

In deze level werd er al gebruik gemaakt van een stad dat werd ontwikkeld door de jobstudenten. Ook de user interface werd al reeds voorzien in deze level, samen met enkele scripts die instonden voor de functionaliteit van de andere schermen in dit level.

## 4.2 Na afloop van de stage

Aangezien er maar één van de 5 partijen origineel aan bod kwam in het spel, was uitbreiding noodzakelijk. Onderstaande lijst bevat alle zaken die doorheen de stage toegevoegd, verbeterd of verwijderd zijn.

- Menu om te navigeren naar andere spelniveau's
  - Oplijsting van alle levels.
    - Introduction
    - Producer level
    - TGO level
    - DGO level
    - Supplier level
    - Consumer level
    - Mecoms level
- Highscore menu
  - Het aquarium is niet meer aanwezig en is permanent vervangen door een lijst, om de duidelijkheid van de highscores te behouden.
- Support pagina

Bij alle levels is de stad rechtstreeks geïmporteerd van de originele Supplier level. Ook de user interface van alle levels is deels aangepast om dit meer overeen te laten stemmen met de werking van de level.

Nieuwe UI of scripts die de nieuwe UI elementen aansturen, zijn handmatig toegevoegd en geprogrammeerd.

## **5 Uitvoering**

### **5.1 Project Management**

Om een goed beeld te behouden over het project, werd er gebruik gemaakt van Azure Devops.

Azure Devops is een project management tool dat zeer vergelijkbaar is met Jira.

Er werd gekozen om met deze project management tool te werken, om een nieuwe ervaring op te doen en omdat deze tool ook wordt gebruikt bij projecten binnen in Ferranti.

Deze project management tool geeft de mogelijkheid om een project aan te pakken met een agile methodologie.

Per level werd er een sprint voorzien met een lengte van twee weken.

Elk onderdeel in deze level werd onderverdeeld in features. Deze features werden onderverdeeld in kleinere user stories die op hun beurt onderverdeeld werden in tasks.

Na elke sprint werd er een moment ingepland om de vooruitgang te bekijken van het project en om andere feedback of ideeën te bespreken.

Via de burndowncharts, die beschikbaar werden gesteld via het portaal op Azure Devops, kwam er al snel duidelijkheid over hoe het project vorderde per sprint.

Vaak konden er ook al user stories van andere levels opgenomen worden in de vorige sprint. Dit zorgde ervoor dat het project al snel afgewerkt kon worden en waardoor er ook nieuwe uitbreidingsmogelijkheden konden worden voorzien.

Ook tijdens het testen werd er gebruik gemaakt van de project management tool.

Telkens als er een bug gevonden werd, kon deze makkelijk toegevoegd worden aan de lopende sprint. Zo kwam er een overzicht van alle problemen die konden worden teruggevonden in het spel.

## **5.2 Version Control**

Voor version control werd er gebruik gemaakt van Git. Door Git te gebruiken als version control, kwam er een duidelijk beeld over elke verandering die in het project werd toegepast.

Telkens als er een onderdeel werd afgewerkt, werden alle veranderingen gecommited op de Developer branch van het project.

Telkens als er een sprint werd afgewerkt, werd de Developer branch gemerged met de Master branch van het project en werden de nieuwe wijzigingen na deze merge opnieuw gecommited op de Developer branch.

## **5.3 Game Development**

Aangezien het project al reeds in Unity in ontwikkeling was, was de keuze vrij logisch om het project hiermee verder te zetten. De scripts die reeds geschreven waren, werden allemaal geschreven in C# en ook hiermee werden alle nieuwe scripts mee verdergezet, vooral om consistentie te behouden doorheen het project.

### **5.3.1 Prioriteit**

De prioriteit van deze opdracht lag eerst vooral bij het toevoegen van nieuwe content en levels zodat alle partijen die voorkomen uit de energiesector voorkomen in het spel. Elke level wordt voorgesteld door een partij uit de energiesector. Hierna werd de focus gelegd op extra functionaliteit zoals de support pagina en de verbeterde hoge scores pagina.

## **6 Ondervonden problemen en oplossingen**

### **6.1 Data opslaan via XML**

Één van de functionaliteiten van het spel is het opslaan van de hogescores van spelers op een bestandslocatie op de computer van de gebruiker. Er moest dus een keuze gemaakt worden tussen verschillende manieren om data op te slaan. Doordat de prioriteit lag bij het toevoegen van nieuwe levels en nieuwe content, was deze functionaliteit pas laat van toepassing. Door het gebrek aan tijd en de ontbrekende kennis over SQL databanken en de samenwerking ervan met Unity, was deze optie al snel uitgesloten. Hiernaast zijn er nog enkele manieren om data lokaal op te slaan die zeer goed samenwerken met Unity.

#### **6.1.1 PlayerPrefs**

PlayerPrefs is één van de de makkelijkere manier om data op te slaan, maar niet de meest efficiënte manier om data op te slaan.

In de PlayerPrefs kan men telkens een string, float of int waarde toevoegen met de naam van de variabele. Dit is dus een goede manier om een bepaalde configuratie op te slaan of bepaalde data over de speler zelf, maar niet voor grote stukken tekst of grote lijsten van objecten.

Deze manier van data opslaan word in dit project wel gebruikt om te weten of het geluid in het spel aan of af moet staan.

#### **6.1.2 JSON**

Via de JSON methode kan men op een snelle manier veel data opslaan, omdat via de JSON methode de data snel kan worden gelezen en snel kan worden aangepast en ook omdat array objecten ondersteund worden door JSON.

De methode om data op te slaan via JSON data is momenteel nog niet geïntegreerd in het project maar zou ook een mogelijke uitbreidingsmogelijkheid zijn om later toe te voegen.

Zo zouden bijvoorbeeld de waardes van de gebouwen in de producer level opgeslagen kunnen worden in een JSON bestand.

#### **6.1.3 Binary**

Binary is ook een goede manier om data op te slaan dat zeer vergelijkbaar is met hoe men data kan opslaan via JSON, enkel wordt er bij JSON een string gegenereerd en bij Binary word er byte data gegenereerd. Ook zorgt deze manier van data opslaan al voor een eerste vorm van encryptie omdat men deze bestanden moeilijk kan lezen zonder deze data om te vormen.

Dit zorgt er dan wel voor dat het tijdens development ook moeilijker is om aanpassingen te maken aan het bestand.

#### **6.1.4 XML**

XML is zeer vergelijkbaar met de JSON methode, enkel is deze methode beter geschikt voor grote teksten of grote lijsten met tekst omdat XML de data op een meer leesbare manier weergeeft.

Hierdoor is het dus handig om de grote stukken tekst zoals de introducties van alle levels in een XML bestand op te slaan en ook de hogescores van de spelers.

Ook bepaalde teksten, zoals de introductie teksten van verschillende levels, werden reeds in XML bestanden opgeslagen om snelle aanpassingen te kunnen maken.

## **6.2 Converteren XML structuur van de vragen**

Door telkens dezelfde structuur te gebruiken bij het bedenken van nieuwe vragen, werd het aanpassen van het reeds bestaande 'quiz controller' script op de leverancier level al snel een mogelijkheid. Hierdoor kon er telkens één algemeen script gebonden worden aan één algemeen quiz canvas. De vragen van een bepaalde level zouden dan telkens automatisch terecht komen op het quiz canvas van desbetreffend level.

Door de implementatie van de functionaliteit om globaal items op te slaan (zoals reports, suggesties, vragen...), werd er door een XmlSerializer een XML structuur gegenereerd dat een andere XML structuur opleverde dan de XML structuur van de reeds bestaande vragen. De reeds bestaande vragen moesten dan opnieuw geconverteerd worden naar de nieuwe structuur die gegenereerd werd door de XmlSerializer.

Door een tijdelijk script te schrijven dat vergelijkbaar is met het 'quiz controller' script, konden alle vragen uit de oude lijst opgehaald worden en via de XmlSerializer opnieuw toegevoegd worden aan het nieuwe bestand met de nieuwe XML structuur.

## **6.3 Implementeren van encryptie en decryptie**

Bij het verwezelijken van het opslaan van highscores in een bestand dat terecht zou komen op een gedeelde locatie, kwamen er enkele vragen aan te pas. De highscores bevatten geen confidentiële informatie, maar indien dit wel het geval zou zijn, is er nood aan encryptie en decryptie.

Dit was al reeds een deel verwerkt in het project voor aanvang van de stage. Door deze manier van werken over te nemen, kwamen hier ook enkele problemen bij kijken.

Telkens het bestand opnieuw gedecrypteerd werd, was het bestand onbruikbaar door het spel. Door een gebrek aan tijd kon encryptie niet meer geïmplementeerd worden.

## **6.4 Sturen van e-mail bij loggen van reports, suggestions ...**

Notificaties bij het loggen van reports, suggesties of vragen waren ook noodzakelijk. Telkens als er een probleem, suggestie of nieuwe vraag gelogd moet worden, dan moet er een e-mail verstuurd worden naar een e-mail adres als notificatie.

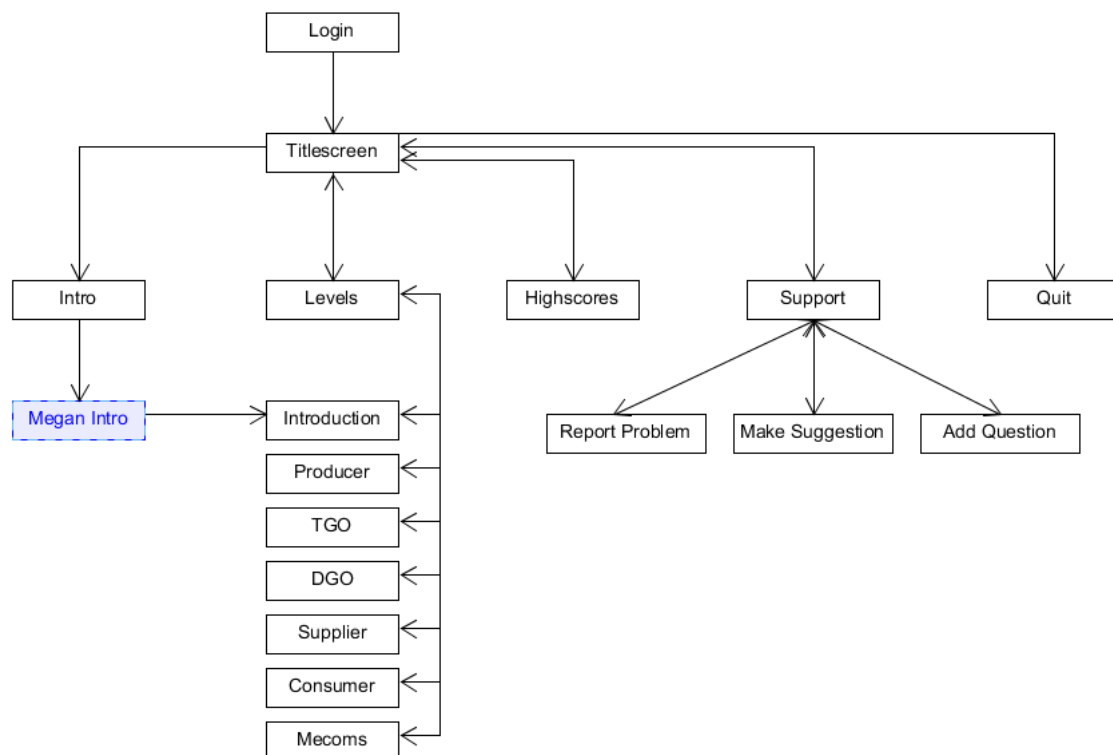
Dit werd opgelost door gebruik te maken van SMTP (simple mail transfer protocol), waarbij de datum van het loggen, type van gelogd item en titel van gelogd item in de titel van de mail staat en waarbij de uitgebreide beschrijving over het gelogde item in de beschrijving van de mail komt te staan.

## 7 Handleiding

### 7.1 Algemeen

In dit deel van de handleiding zullen vooral onderdelen aan bod komen dat in één of meerdere levels kan voorkomen.

Onderstaande afbeelding geeft een algemeen beeld over de flow van het spel en geeft weer hoe men kan wisselen tussen verschillende onderdelen van het spel.



Figuur 3 - flow van het spel

### 7.1.1 User interface

De user interface zal in alle levels verschillen van andere levels. Zaken zoals geld of energie verbruik dat wordt weergegeven op de user interface kan in sommige levels wel voorkomen en in andere levels niet.

Er zijn uiteraard zaken die wel in de user interface van elke level zullen terugkomen. Zaken die terugkomen, worden meestal bovenaan in de user interface weergegeven en zaken die niet terugkomen, worden meestal onderaan in de user interface weergegeven.

De onderdelen die in de user interface van elke level terugkomen zijn:

- Spelerdata: Dit onderdeel bevat de gebruikersnaam van de speler en de score dat speler tot nu toe behaald heeft. Deze score is de highscore dat de speler ook kan terugvinden in de Highscores menu.
- Pauzeknop: Deze knop kan de speler gebruiken om het spel te pauzeren.

De opmaak van elke user interface blijft consistent doorheen het spel. De kleuren die op alle user interfaces voorkomen zijn default kleuren of kleuren die voorkomen op een kleurenpalette dat gebruikt wordt binnen Mecoms.

### 7.1.2 Pauze menu

Als de speler het spel gepauzeerd heeft, zal het pauze menu verschijnen. Op dit menu kan de speler ervoor kiezen om het spel terug verder te zetten of om de level af te sluiten en terug te keren naar het hoofdmenu.

Ook staat er op het pauzemenue een kleine uitleg over hoe de speler de camera kan bewegen.

### 7.1.3 Quiz & Score

In sommige levels zal er na een bepaald interval een pop-up verschijnen dat aantoont dat de speler een vraag kan beantwoorden met als onderwerp de partij waar de huidige level om draait.

Deze vragen zullen bestaan uit multiple-choice vraag waarbij de speler één van de gegeven antwoorden moet selecteren. Indien er een antwoord geselecteerd wordt, worden de knoppen van de antwoorden uitgeschakeld en wordt het correcte antwoord weergegeven met een groene kleur en alle foute antwoorden worden weergegeven met een rode kleur.

Na afloop van een level heeft de speler de mogelijkheid om de resultaten van het huidige level te bekijken en om alle beantwoorde vragen in te kijken.

Hierbij krijgt de speler de vraag te zien die de speler beantwoord heeft, alsook het correcte antwoord op de vraag en het antwoord dat de speler heeft aangeduid.

## 7.2 Producer level

In de producer level is het de taak van de speler om zoveel mogelijk contracten aan de 'suppliers' of leveranciers te verkopen, en om genoeg energie producerende gebouwen aan te kopen. Sommige energie producerende gebouwen produceren ook vervuiling en dit wordt dan ook in rekening gebracht. Indien het vervuilingsgehalte van de stad te hoog is, zal de speler automatisch de level verliezen. Ook gebouwen met groene energie hebben hun eigen voor- en nadelen. Gebouwen die groene energie produceren zijn meestal afhankelijk van het weer of andere omstandigheden.

Doorheen de level, op een willekeurig interval, zal het weer dan ook veranderen.

Dit zorgt ervoor dat de getroffen gebouwen een stijging of daling in energie productie vertonen dat afhankelijk is van de weersomstandigheden.

De speler kan extra gebouwen aankopen via het 'Market' menu. In dit menu kan de speler zien hoeveel elk gebouw kost, produceert, wat voor type energie het gebouw produceert en de hoeveelheid vervuiling dat dit gebouw procentueel met zich meebrengt. Tevens kan de speler hier zien hoeveel gebouwen hij van elk type heeft en de bijbehorende statistieken hiervan. Deze statistieken bevatten de totale productie, vervuiling en het eventuele gedrag in productie afhankelijk van het weer.

Zoals eerder vermeld zal de speler contracten moeten verkopen aan de leveranciers.

Deze zullen aangeboden worden na een bepaald interval via een pop-up. Op elk contract zal te zien zijn hoeveel energie de speler aan de leverancier verkoopt en hoeveel winst hij maakt door dit contract te accepteren.

Elk contract zal terechtkomen op het 'Contract' menu. Op dit menu is te zien welke contracten de speler in zijn bezit heeft. Tevens is er een optie om een bepaald contract op te zeggen. Dit zal vooral nodig zijn als de speler te weinig energie produceert en daardoor een contract moet opzeggen. Indien de speler dit niet doet, zal de algemene blijheid van de stad dalen en ook hierdoor kan de speler dus de level verliezen.

Er zullen doorheen de level ook andere, gelijkaardige pop-ups verschijnen zoals de pop-ups van de contractaanbiedingen. Deze pop-ups duiden aan dat de speler een vraag kan beantwoorden in verband met de producer. De speler kan kiezen om deze vraag te weigeren of te accepteren. Om de speler toch aan te moedigen om tijdig een vraag te beantwoorden, zal de blijheid van de stad doorheen de level blijven dalen. Elk juist antwoord zal beloond worden met een bonus in score en blijheid.

### **Korte samenvatting van de voorwaarden om de level te verliezen:**

*Enkel één van deze voorwaarden is voldoende om het spel te verliezen.*

- Geld van de speler < 0
- Algemene blijheid < 0%
- Vervuiling van de stad > 100%

### **Korte samenvatting van de voorwaarden om de level te winnen:**

*Enkel één van deze voorwaarden is voldoende om het spel te winnen.*

- Geld van de speler > 1.000.000
- Aantal consistent correct beantwoorde vragen > 20
- Aantal aangenomen contracten > 14



## **7.3 TGO level**

Aangezien er niet veel informatie beschikbaar is over de TGO, werd deze level eerder ontwikkeld als een ontspannende level dat bestaan uit enkele bekende minigames.

### **7.3.1 Memory**

Het doel in de Memory minigame is om alle afbeeldingparen te zoeken.

Alle afbeelding paren zullen omgekeerd op een willekeurige plaats terug te vinden zijn op het spelbord.

De speler kan twee afbeeldingen kiezen om om te draaien.

Indien de afbeeldingen een afbeeldingenpaar vormen, blijven de afbeeldingen op het spelbord liggen en worden er punten toegevoegd aan de score van de speler. Indien de afbeeldingen geen afbeeldingenpaar vormen, worden de afbeeldingen terug omgedraaid en moet de speler opnieuw zoeken naar afbeeldingen die wel een afbeeldingenpaar vormen.

### **7.3.2 Breakout**

Bij de Breakout Minigame krijgt de speler enkele rijen van 'tegels' te zien die de speler moet vernietigen.

Dit kan de speler bereiken door de bal te besturen die door het speelveld zal vliegen.

Telkens dat de bal een tegel raakt, zal de speler punten krijgen en zal deze tegel verdwijnen en zal de bal teruggekaatst worden.

Om opnieuw de bal terug te kaatsen, moet de speler een platform bedienen met behulp van zijwaartse bewegingen die de speler moet maken met de computermuis of pijltoesten om het toetsenbord.

Als de bal de onderste grens van het speelveld raakt, dan verliest de speler één van zijn drie levens.

De bal zal dan opnieuw gereset worden om verder te kunnen spelen.

Als de speler geen levens meer heeft, dan reset heel het spel en moet de speler opnieuw proberen om alle tegels te vernietigen.

### 7.3.3 Simon says

Bij aanvang van de Simon says minigame wordt er één van de vier mogelijke kleuren opgelicht voor één seconde en zal er een geluidssignaal afspelen dat ook een onderscheid maakt over welke kleur het gaat.

Indien de speler op de opgelichte kleur drukt, zal deze kleur opnieuw opgelicht worden, samen met het geluidssignaal.

Hierna zal er een nieuwe willekeurige kleur oplichten en bijhorend geluidssignaal afspelen.

Deze kleur zal dan achteraan de sequentie worden toegevoegd.

Telkens zal de speler de hele sequentie moeten nabootsen en herhalen.

Indien de speler de sequentie correct herhaald, zal de speler punten krijgen afhankelijk van de lengte van de sequentie.

Indien de speler de sequentie incorrect herhaald, zal de sequentie resetten en zal er opnieuw een kleur gekozen worden.

### 7.3.4 Mastermind

Bij aanvang van de Mastermind minigame, zal er een willekeurige kleurencombinatie gegenereerd worden dat de speler moet raden.

De kleurencombinatie bestaat uit 4 kleuren die willekeurig gekozen worden uit 8 mogelijke kleuren. Ook positie is zeer belangrijk tijdens het raden van de kleurencombinatie.

Indien er op de eerste positie van de kleurencombinatie een bepaalde kleur voorkomt, dan zal de speler tijdens het raden ook deze bepaalde kleur moeten plaatsen op de eerste plaats. Het is dus niet voldoende om alle kleuren te raden, maar om ook de posities van de kleuren te raden.

Indien de speler een gok gewaagd heeft, dan zal deze worden toegevoegd aan een antwoordenlijst. Hierop krijgt de speler ook feedback over de kleurencombinatie dat de speler heeft ingediend.

Indien er een kleur aanwezig is, maar niet op de juiste positie staat, zal er een gele bolletje verschijnen naast het gegeven antwoord.

Indien er een kleur aanwezig is en ook op de juiste positie staat, zal er een groene bolletje verschijnen naast het gegeven antwoord.

De speler wint als er vier groene bolletjes naast het laatst gegeven antwoord staan.

### 7.3.5 Hangman

Bij aanvang van de Hangman minigame, zal er een willekeurig woord worden gekozen uit een woordendatabank.

Het doel van deze minigame is dat de speler het willekeurig gekozen woord moet raden. De speler wordt hierin al gedeeltelijk in geholpen doordat de lengte van het woord wordt weergegeven met underscores die elke letter in het woord representeren.

De speler kan dan kiezen om een letter te raden of om het hele woord te raden.

Indien de speler ervoor kiest om een letter te raden, dan kunnen er twee zaken plaatsvinden:

1. De letter maakt deel uit van het woord en alle plaatsen waar de letter voorkomt in het verborgen woord, zullen vervangen worden door de gekozen letter.
2. De letter maakt geen deel uit van het woord en wordt toegevoegd aan een lijst met foutive woorden die de speler ten allen tijde kan zien en raadplegen. Ook wordt er bij elke foutive poging van het raden van een letter een deel van de 'Hangman' afbeelding getoond.

Indien de afbeelding de volledige 'Hangman' toont (of na een zes pogingen), dan verliest de speler het spel en moet de speler het spel resetten.

Indien de speler ervoor kiest om een woord te raden, dan kunnen er twee zaken plaatsvinden:

1. Het geraden woord stemt overeen met het willekeurig geselecteerde woord en de speler wint het spel
2. Het geraden woord stemt niet overeen met het willekeurig geselecteerde woord en de speler verliest het spel

### 7.3.6 Catcher

Bij de Catcher minigame is het doel van de speler om zoveel mogelijk 'lightning' of electriciteit op te vangen.

Dit kan de speler bereiken door een 'catcher' of vanger te besturen dat gebonden is aan de horizontale bewegingen van de muis.

electriciteits objecten kunnen met willekeurige aantallen en met verschillende snelheden.

Telkens als de speler een electriciteits object vangt, dan krijgt de speler een extra leven en zal ook de valsnelheid van de vallende objecten stijgen.

Als de speler er niet in zou slagen om een lightning object op te vangen, dan verliest de speler 3 levens.

Als alle levens van de speler op zijn, dan verliest de speler het spel en dan zal het spel resetten.

## 7.4 DGO level

In de DGO level zal de speler met 'workers' te werk gaan om problemen op te lossen die zich in de stad zullen voordoen. Elk probleem zal zijn eigen 'severity' of moeilijkheidsgraad hebben. Sommige problemen zullen dus langer duren voordat ze opgelost zijn. De speler krijgt ook de mogelijkheid om bij elk probleem extra workers toe te voegen of te verwijderen om zo de tijd, die nodig is om het probleem op te lossen, te verminderen.

De speler kan zien welke problemen zich voordoen in het 'Problems' menu die de speler kan openen via het Happiness Panel.

De speler zal een beloning krijgen per opgelost probleem in de vorm van geld, waarmee de speler een extra worker kan aankopen. De prijs per worker zal exponentieel stijgen naarmate de speler meer workers bezit.

Ook deze level zal werken met een algemene 'happiness' of blijheid. Indien de speler geen workers inzet om een probleem op te lossen, zal per tijdseenheid de blijheid dalen afhankelijk van de 'severity' van het probleem.

Tevens zal deze level een quiz systeem bevatten dat werkt met pop-ups. De speler kan kiezen om een vraag te accepteren of te weigeren. Door de vraag correct te beantwoorden, zal de speler een bonus krijgen in blijheid. Indien de vraag geweigerd wordt, zal de algemene blijheid dalen. Om de speler toch aan te moedigen om regelmatig een vraag te accepteren, zal de blijheid doorheen de level blijven dalen.

### **Korte samenvatting van de voorwaarden om de level te verliezen:**

*Enkel één van deze voorwaarden is voldoende om het spel te verliezen.*

- Geld van de speler  $< 0$
- Algemene blijheid  $< 0\%$

### **Korte samenvatting van de voorwaarden om de level te winnen:**

*Enkel één van deze voorwaarden is voldoende om het spel te winnen.*

- Aantal opgeloste problemen  $> 50$
- Aantal consistent correct beantwoorde vragen  $> 20$

## 7.5 Supplier level

In de supplier level is het de taak van de speler om de juiste balans te zoeken tussen energie aankopen van de producenten en het verbruik van de consumenten.

Alsook is het de taak van de leverancier om zoveel mogelijk klanten van de concurrenten te laten overschakelen naar de leverancier van de speler.

Dit kan men gemakkelijk bereiken door de dag- en/of nachttarieven te verlagen of om meer groene dan grijze energie aan te kopen.

De dag- en/of nachttarieven verlagen heeft natuurlijk als gevolg dat de speler minder geld zal verdienen, maar meer consumenten zal laten overstappen.

Het energieniveau zal elke vijf seconden dalen wat ervoor zorgt dat de speler tijdig nieuwe electriciteit zal moeten aankopen bij een producent.

De speler kan kiezen tussen vier producenten die groene energie produceren en tussen vier producenten die grijze energie produceren.

Elke producent heeft zijn eigen prijs en zijn eigen hoeveelheid energie dat deze producent verkoopt. Het kan dus soms voordeliger zijn om bij een producent te kopen dat grijze energie verkoopt dan bij een producent dat groene energie verkoopt.

Na elke aankoop worden de prijzen en hoeveelheden bij alle producenten gereset.

Ook zal de speler moeten oppassen dat hij niet te veel energie aankoopt want dat zal resulteren in een boete dat de speler moet betalen wegens verspilling van energie als ook een daling in blijheid.

### **Korte samenvatting van de voorwaarden om de level te verliezen:**

*Enkel één van deze voorwaarden is voldoende om het spel te verliezen.*

- Geld van de speler  $< 0$
- Algemene blijheid  $< 0\%$

### **Korte samenvatting van de voorwaarden om de level te winnen:**

*Enkel één van deze voorwaarden is voldoende om het spel te winnen.*

- De speler heeft alle consumenten laten overstappen naar zijn eigen leverancier

## 7.6 Consumer level

In deze level is het de bedoeling om de speler de beste keuzes te laten maken als consument.

De speler zal bij aanvang van de level enkele keuzes moeten maken waarmee hij de rest van de level zal verderzetten. Elke keuze zal ook voor- en nadelen met zich meebrengen die invloed zullen hebben op één of meerdere factoren.

Deze factoren zijn:

- Energieverbruik
- Geld
- Type stroomverbruik (groene of grijze energie)

Na aanvang van de startkeuzes, zal er willekeurig gekozen worden tussen een meerkeuze vraag of een dilemma.

De meerkeuze vragen hebben geen invloed op de factoren waarop de speler zo goed mogelijk moet scoren.

De dilemma's hebben in tegenstelling tot de meerkeuze vragen wel een invloed op één of meerdere factoren.

Afhankelijk van de vraag en van het antwoord dat de speler geeft, zal er stijging of daling plaatsvinden bij één of meerdere factoren.

Telkens als men een meerkeuze vraag of dilemma beantwoordt, zal er na een 5 seconde interval opnieuw gekozen worden tussen een meerkeuze vraag of een dilemma.

### **Korte samenvatting van de voorwaarden om de level te verliezen**

- Kleur van de energieverbruik meter is volledig rood
- Kleur van de geld meter is volledig rood
- Kleur van de type stroomverbruik meter is volledig rood

### **Korte samenvatting van de voorwaarden om de level te winnen:**

- Aantal consistent correct beantwoorde vragen > 30
- Kleur van de energieverbruik meter is volledig groen
- Kleur van de geld meter is volledig groen
- Kleur van de type stroomverbruik meter is volledig groen

## 7.7 Mecoms level

In de Mecoms level draait het vooral rond het Mecoms product zelf.

Bij aanvang van de level wordt er een simpel keuze menu getoond waarop de speler kan kiezen wat voor vraag de speler wilt beantwoorden.

De speler kan kiezen uit:

- Multiple choice vragen, waarbij de speler een vraag moet beantwoorden waarbij de mogelijke antwoorden getoond worden.
- Correct order vragen, waarbij de speler een proces in de juiste volgorde moet zetten met behulp van een Drag & Drop systeem.
- Willekeurige vraag, waarbij de speler één van de drie voorafgaande soorten vragen moet beantwoorden.

Indien de speler een vraag beantwoord heeft, krijgt de speler opnieuw het keuze menu te zien en moet de speler opnieuw kiezen welke vraag de speler wilt beantwoorden.

### **Korte samenvatting van de voorwaarden om de level te verliezen:**

In deze level kan de speler niet verliezen.

### **Korte samenvatting van de voorwaarden om de level te winnen:**

- Totaal correct beantwoorde vraagstukken > 100

## 8 Support pagina

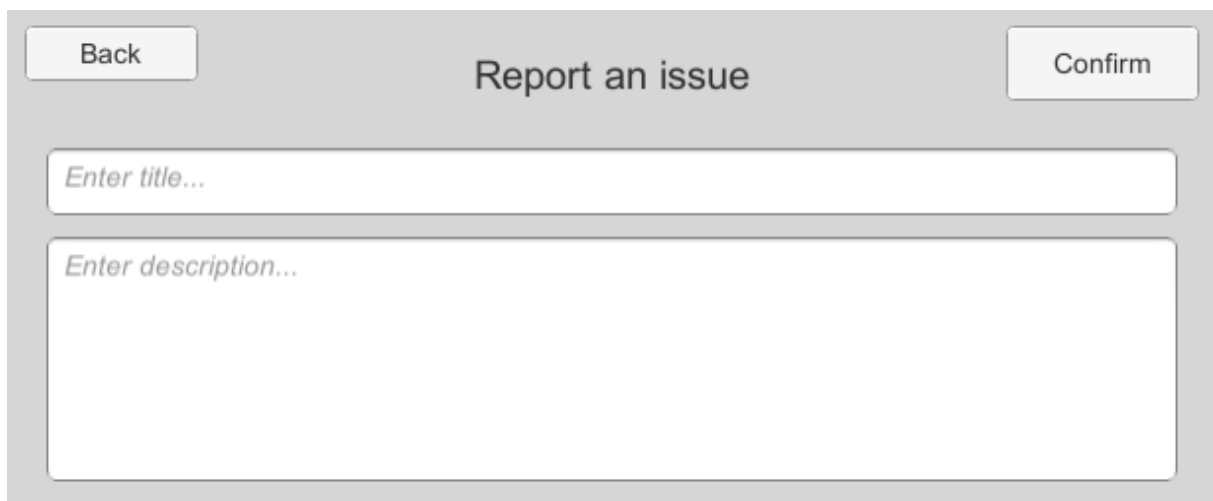
De support pagina is een pagina waar de gebruiker de developers kan helpen door problemen te melden die de speler doorheen het spel ondervonden heeft, door suggesties te maken aan de developers of door potentiële nieuwe vragen toe te voegen aan

Deze problemen, suggesties en vragen worden elks toegevoegd aan een XML bestand dat wordt opgeslagen op de Ferranti omgeving.

### 8.1 Probleem rapporteren

Hier kan de speler een probleem melden dat de speler tijdens het spelen van het spel heeft ondervonden.

Een probleem of report bestaat uit een abstracte titel dat duidelijk weergeeft wat het onderwerp van het probleem is en ook een descriptie om problemen grondiger uit te leggen, zodat er een beter beeld kan worden gegeven aan de developers.

The image shows a web form titled "Report an issue". At the top left is a "Back" button, and at the top right is a "Confirm" button. Below the title, there are two input fields. The first is a single-line text box with the placeholder text "Enter title...". The second is a larger multi-line text box with the placeholder text "Enter description...". The entire form is enclosed in a light gray border.

*Figuur 4 – user interface: probleem rapporteren*



### 8.1.1 Code: toevoegen en verwijderen van een report

Via de input fields worden de titel en descriptie parameters meegegeven aan het script.

Eerst zal er nagekeken worden of de titel en de descriptie niet leeg zijn. Indien dit wel het geval is, zal de gebruiker een melding krijgen dat alle velden ingevuld moeten worden.

Daarna zal er een nieuwe report entry aangemaakt worden dat ook een titel en een descriptie bevat. Ook bevat deze nieuwe entry de huidige datum, zodat er steeds een beeld kan gevormd worden wanneer het potentiële probleem zich heeft voorgedaan. Hierna zal deze entry toegevoegd worden aan de lokale reports databank en wordt deze databank opgeslaan.

Ook zoals eerder vermeld zal er een e-mail verzonden worden waarbij de titel van het probleem in de titel van de mail zal terecht komen en de descriptie van het probleem zal in de descriptie van de mail terecht komen.

Hierna krijgt de gebruiker een confirmatie bericht dat aantoont dat het probleem succesvol gerapporteerd is.

```
1 reference | Sam Van Bogaert, 101 days ago | 1 author, 1 change
public void AddReport(string title, string description)
{
    ReportEntry entry = new ReportEntry
    {
        title = title,
        description = description,
        dateLogged = string.Format("logged on: {0}", DateTime.Now.ToString(new CultureInfo("en-GB")))
    };
    reportDB.list.Add(entry);
}

1 reference | Sam Van Bogaert, 101 days ago | 1 author, 1 change
public void RemoveReport(int index)
{
    reportDB.list.RemoveAt(index);
}
```

*Figuur 5 - code: toevoegen en verwijderen van een report*

```
1 reference | Sam Van Bogaert, 87 days ago | 1 author, 2 changes
void ConfirmReport()
{
    //throw error if inputfields are left blank
    if(titleField.text == "" || descriptionField.text == "")
    {
        commentText.color = new Color(1, 0, 0); // make text color red
        commentText.text = "Please fill in a title and a description.";
    }
    //confirm report
    else
    {
        string title = titleField.text;
        string description = descriptionField.text;

        XMLManager.instance.AddReport(title, description);
        XMLManager.instance.SaveReports(false);

        EmailController.instance.SendEmail("Report", title, description, DateTime.Now.ToString(new CultureInfo("en-GB")));

        titleField.text = "";
        descriptionField.text = "";
        commentText.color = new Color(0, 1, 0); // make text color green
        commentText.text = "Thank you for reporting an issue!";
    }
}
```

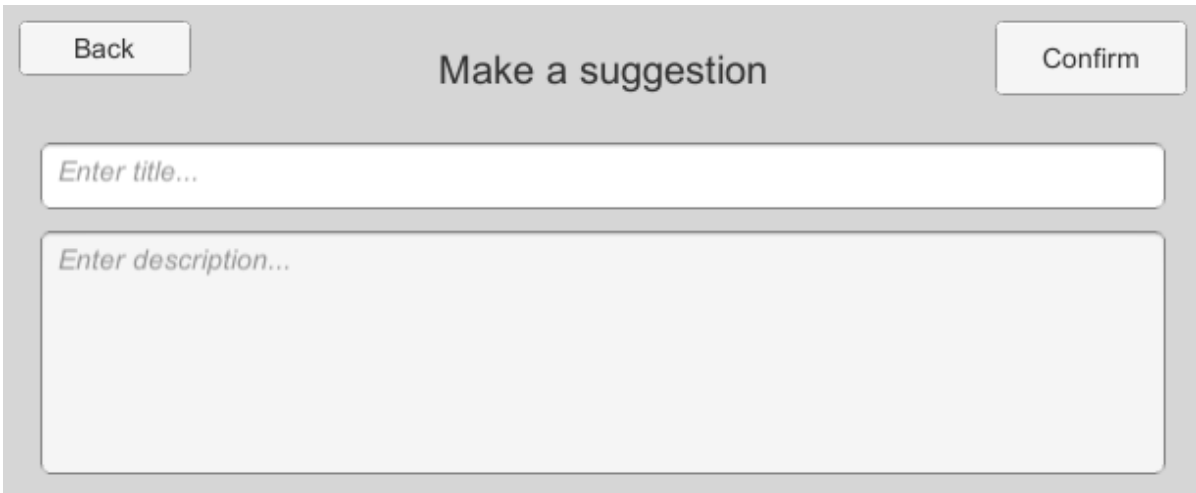
*Figuur 6 - code: report confirmeren*

## 8.2 Suggesties maken

Hier kan de speler een suggestie maken aan de developers.  
Suggesties werken op dezelfde manier als problemen reporten.

Een suggestie bestaat, net zoals een probleem, uit een abstracte titel dat duidelijk weergeeft wat er eventueel ontbreekt of op een alternatieve manier gepresenteerd kan worden, en een descriptie om de suggestie beter uit te leggen.

De code om suggesties toe te voegen of te verwijderen is zeer vergelijkbaar met het toevoegen en verwijderen van problemen.

The image shows a user interface for making a suggestion. It features a light gray background with a central title "Make a suggestion" in a bold, dark gray font. On the left side, there is a "Back" button, and on the right side, there is a "Confirm" button. Below the title, there are two input fields. The first is a single-line text field with the placeholder text "Enter title...". The second is a larger, multi-line text field with the placeholder text "Enter description...". Both input fields have a light gray border and a subtle shadow.

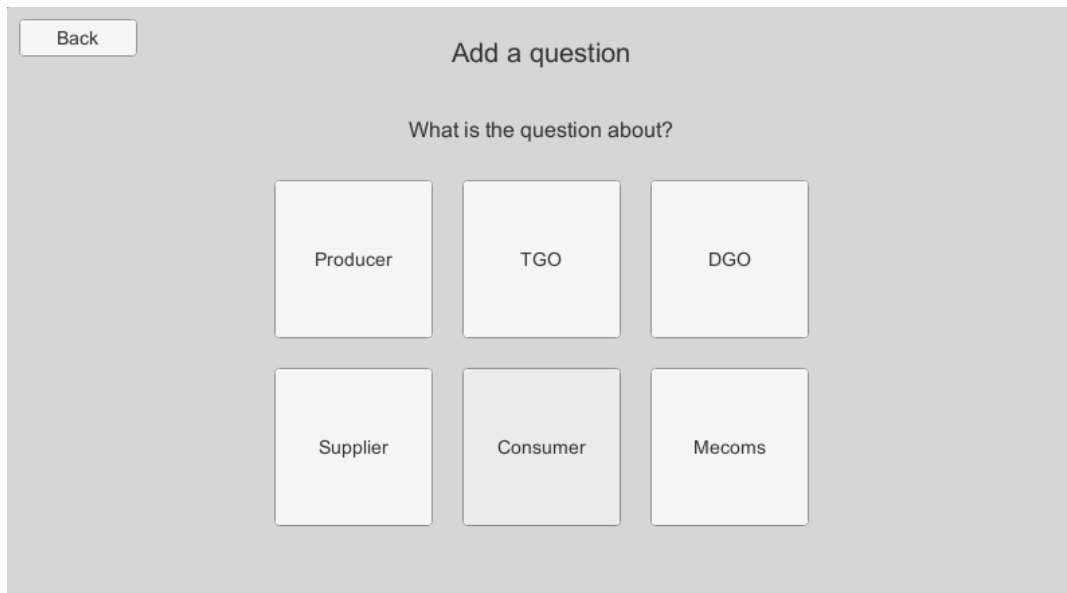
*Figuur 7 – user interface: suggestie maken*

### 8.3 Vragen toevoegen

Vragen kan men net zoals andere items toevoegen via de support page.

Eerst moet men kiezen welke level in het spel het beste past bij het onderwerp van de vraag.

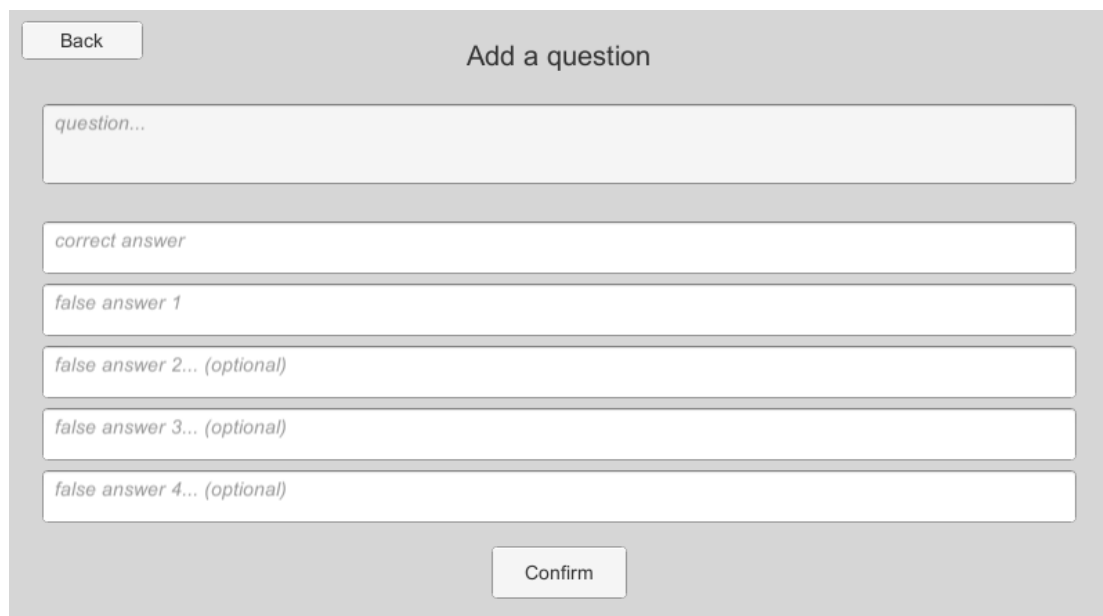
De gebruiker krijgt enkele knoppen te zien waar telkens de naam van een level op staat.



The interface is titled "Add a question" and includes a "Back" button in the top left. Below the title, it asks "What is the question about?". There are six buttons arranged in a 2x3 grid: "Producer", "TGO", "DGO" in the top row, and "Supplier", "Consumer", "Mecoms" in the bottom row. The "Consumer" button is highlighted with a light gray background.

*Figuur 8 – user interface: kiezen van level*

Als de speler de level gekozen heeft waarbij de vraag het beste past, kan hij de vraag toevoegen aan dat level. Elke vraag bestaat uit een vraag en minstens één juist en één fout antwoord. Optioneel kan de speler er ook nog voor kiezen om 3 extra foutieve antwoorden toe te voegen. Deze vraag wordt dan toegevoegd aan een XML file dat momenteel op een gedeelde locatie wordt opgeslagen.



The interface is titled "Add a question" and includes a "Back" button in the top left. It contains several input fields: "question...", "correct answer", "false answer 1", "false answer 2... (optional)", "false answer 3... (optional)", and "false answer 4... (optional)". A "Confirm" button is located at the bottom right.

*Figuur 9 – user interface: invullen van vraag & antwoorden*

## 8.4 Admin pagina

Als notificatie voor de developers wordt er bij elk gelogd item een e-mail verzonden naar het support account van de developers. Zo blijven de developers steeds op de hoogte over nieuwe problemen of suggesties.

Elke e-mail bevat als onderwerp het gelogde tijdstip gevolgd met het type van item (bv. Report, suggestion of question) gevolgd door de titel van het gelogde item.

In de content van de e-mail kan men de descriptie van het gelogde item terugvinden.

Op de UI zal ook een password field staan. Indien de speler of de developer het correcte password invult, kan de speler of de developer de admin pagina openen.

Op deze admin pagina is het ook mogelijk om een lijst van de gelogde items te bekijken via de respectievelijke 'radio buttons' en hebben bij elk item in deze lijst de mogelijkheid om het item te verwijderen.

## **9 Project structuur**

### **9.1 Doc**

#### **9.1.1 Nieuwe documentatie**

Deze folder bevat alle documentatie over het project dat geschreven werd na aanvang van de stage. In deze folder kan men de feedback terugvinden na het test moment met meerdere werknemers. In dit feedback bestand kon de gebruiker een score meegeven tussen één en tien voor enkele eigenschappen zoals bv. Correcte informatie, Intuitieve UI, aantal ondervonden bugs ... Ook werd er altijd ruimte voorzien om extra feedback te bezorgen indien dit nodig zou moeten zijn.

In de documentatie folder kan men ook een bronnenlijst terugvinden waar de bronnen terugkomen die gebruikt werden voor bugfixing, nieuwe informatie dat in het spel verwerkt werd en zaken zoals tutorials die een grote hulp waren tijdens het ontwikkelen van enkele onderdelen van het project.

Er zijn ook mockups beschikbaar die gebruikt werden bij het visualiseren van ideeën voor nieuwe levels.

#### **9.1.2 Originele documentatie**

Deze folder bevat alle oude documentatie over het project dat geschreven werd voor de aanvang van de stage.

In deze folder kan men de training terugvinden die origineel aan de nieuwe trainees werd gegeven. Uit deze training komt de meeste informatie terug voor dat in het spel verwerkt werd.

Ook kan men hier een eerste handleiding terugvinden over de eerste versie van het spel en ook een handleiding over het project zelf.

## 9.2 Src

### 9.2.1 Mecity project

#### 9.2.1.1 Assets

- \_TerrainAutoUpgrade
- Animations
  - Bevat alle animaties die men kan terugvinden in het spel
- BOXOPHOBIC
  - Bevat de 'Skybox' (sky texture met geanimeerde wolken) van het spel
- Buildings
  - Bevat alle gebouwen die men kan terugvinden in de stad van het spel
- Decorations
  - Bevat textures van wegen, bomen ...
- European Buildings
  - Bevat meer building assets die men kan terugvinden in de stad van het spel
- Hand Painted Grass Texture
  - Bevat de configuratie van de terrein kleur van de stad
- Logo & images
  - Bevat alle logo's en afbeeldingen (2D en 3D) die men kan terugvinden in het spel
- Low Poly Water / LowPoly Environment Pack/ LowPolyWater\_Pack
  - Bevat het geanimeerde water dat men kan terugvinden in de stad van het spel
- Materials
  - Gebruikte materials (color materials & physics materials)
- Mobile
  - Bevat nog meer building assets van MeCity
- Nnj3de\_crucarp
  - Bevat de geanimeerde vissen die men kan terugvinden in de high score menu
- NuclearPowerPlant
  - Bevat de building assets voor de Nuclear Power Plant van de stad
- Prefabs
  - Bevat alle 'Prefabs' (geconfigureerde en meestal herbruikbare game objecten)
    - Bv. PauseCanvas, IntroCanvas, ScoreCanvas ...
- Resources
  - Bevat alle XML files waarin men de introducties en vragen en antwoorden kan terugvinden
- Scenes
  - Bevat alle geconfigureerde 'Scenes' die men kan aanpassen via de Unity Editor
- Scripts
  - Bevat alle scripts die gebonden zijn aan de gameobjecten en die zorgen voor functionaliteit
- Sounds & music
  - Bevat alle muziek en geluiden die men kan terugvinden in het spel
- Tree\_textures
  - Bevat textures voor alle bomen die men kan terugvinden in de stad van het spel
- Videos
  - Bevat de video's die gebruikt worden in de titlescreen en de Megan Intro

### 9.2.1.2 Assets/scripts

Deze folder bevat alle scripts onderverdeeld in subfolders die aanduiden waar de scripts gebruikt worden.

Als men de script bestanden opent, kan men in de commentaar van de code nog extra uitleg vinden dat de werking van de code gedetailleerder zal uitleggen.

#### 1. /admin - scripts voor en- & decryptie van oude highscores

##### a. decrypt.cs

Decryptie van oude highscores.

##### b. encrypt.cs

Encryptie van oude highscores.

#### 2. /consumer - scripts voor consumer level

##### a. ConsumerDilemmaController.cs

Gespecialiseerde quiz controller dat meer gericht is op de dilemma's in de consumer level dat ook controleert wanneer een dilemma beantwoord is.

Hierin worden ook de sliders aangepast bij de gemaakte keuze's op de UI van de consumer level.

##### b. ConsumerLevelController.cs

Zorgt voor de start van de consumer level met startdilemma's en voor de willekeurige keuze tussen een dilemma of multiple choice vraag, elke keer dat er een dilemma of vraag beantwoord is.

##### c. ConsumerQuizController.cs

Gespecialiseerde quiz controller dat meer gericht is op de multiple choice vragen in de consumer level dat ook controleert wanneer een vraag beantwoord is.

#### 3. /dgo – scripts voor DGO level

##### a. DGOEventSystem.cs

Zorgt voor de problem pop-up's en voor de quiz pop-up's die na een bepaald interval zullen voorkomen.

##### b. DGOProblemController.cs

Zorgt voor alle functionaliteit op de 'Problems' menu (toevoegen van workers aan problemen, toevoegen en verwijderen van problemen ...) en de verandering van de 'Happiness Slider' op de User Interface.

##### c. DGOProblemGridFiller.cs

Zorgt ervoor dat alle problemen in de grid op de 'Problems' menu komen te staan.

##### d. DGOUpgradeGridFiller.cs

Zorgt ervoor dat alle upgrades in de grid op de 'Upgrades' menu komen te staan.

##### e. DGOWorkerController.cs

Zorgt ervoor dat de speler workers kan kopen via de 'Buy Workers' menu en dat elke worker meer zal kosten naarmate de hoeveelheid workers dat de speler in zijn bezit heeft.

#### **4. /highscores – scripts voor oude highscores**

##### **a. addFish.cs**

Voegt een vis toe voor elke speler dat in het highscores bestand staat en past de grootte van de vis aan naarmate de score van de speler; hogere score betekent een grotere vis.

##### **b. btnCloseHighscores.cs**

Zorgt ervoor dat de highscores lijst sluit en dat vissen terug willekeurig bewegen.

##### **c. btnScoresOnClick.cs**

Zorgt ervoor dat vissen stoppen met bewegen en dat er een lijst opent met alle spelers en hun score.

##### **d. CameraScript.cs**

Transformeert camera naar een orthografische camera voor beter beeld van de vissen.

##### **e. MoveRandomly.cs**

Laat alle vissen willekeurig bewegen in de highscores menu binnen het beeld van de camera

#### **5. /intro – scripts voor de Megan intro**

##### **a. MeganIntro.cs**

Zorgt voor de tekst en functionaliteit van de Megan Intro.

#### **6. /introduction – scripts voor de introduction level**

##### **a. ClickHouse.cs**

Zorgt ervoor dat het Megan canvas opent als men op het huis klikt.

##### **b. CloseCanvas.cs**

Sluit het producer canvas en speelt de laatste 'Overview' animatie af dat een overzicht geeft over de stad.

##### **c. EnableClickHouse.cs**

Zorgt ervoor dat bij aanvang van de introductie de tekst 'To Begin, click the house in front of you' laat zien.

##### **d. EnableNextLevel.cs**

Zorgt ervoor dat de UI wordt aangezet na de introductie van alle partijen en dat de speler de introductie opnieuw kan bekijken.

##### **e. LoadDGO.cs**

Speelt de DGO animatie af, sluit het supplier canvas en opent het DGO canvas.

##### **f. LoadProducer.cs**

Speelt de Producer animatie af, sluit het TGO canvas en opent het Producer canvas.

##### **g. LoadSupplier.cs**

Speelt de Supplier animatie af, sluit het Megan canvas en opent het Supplier canvas

##### **h. LoadTGO.cs**

Speelt de TGO animatie af, sluit het DGO canvas en opent het TGO canvas

##### **i. StartIntroduction.cs**

Sluit alle canvassen en opent het info canvas

#### **7. /levels – scripts voor bijhouden van score in levels**

##### **a. LevelHighscore.cs**

Houdt de score en de naam van de speler bij in elke level



## **8. /login – scripts voor login pagina**

### **a. Login.cs**

Checkt of de naam dat de speler opgeeft correct is en voegt de naam toe aan het DataScript.

## **9. /mecoms – scripts voor mecoms level**

### **a. MecomsDragHandler.cs**

Zorgt ervoor dat de speler met de antwoorden kan dragen in het 'correct order' menu

### **b. MecomsDrophandler.cs**

Zorgt ervoor dat de antwoorden omgewisseld worden als de speler het antwoord los laat

### **c. MecomsLevelController.cs**

Zorgt ervoor dat het 'Pick' menu opent elke keer dat de speler een vraag beantwoordt.

## **10. /producer – scripts voor producer level**

### **a. ProducerContractController.cs**

Zorgt voor de functionaliteit op het 'Contracts' menu zodat er een contract toegevoegd, aangepast of verwijderd kan worden.

### **b. ProducerContractGridFiller.cs**

Zorgt ervoor dat er een contract toegevoegd, aangepast of verwijderd wordt uit de grid op het 'Contracts' menu.

### **c. ProducerEventSystem.cs**

Zorgt voor de quiz en contract pop-ups (en hun functionaliteit) die na een bepaald interval zullen verschijnen.

### **d. ProducerMarketController.cs**

Zorgt voor de functionaliteit op het 'Market' menu zodat de speler gebouwen kan aankopen en verkopen.

### **e. ProducerMarketGridFiller.cs**

Zorgt ervoor dat grid op het 'Market' menu wordt gevuld met gebouwen dat de speler kan kopen of verkopen

## **11. /supplier – scripts voor supplier level**

### **a. changeDGO.cs**

Zorgt voor de functionaliteit op het 'Market' menu, zodat de speler meer energie kan kopen

### **b. CubeSatisfaction.cs**

Checkt of de benodigheden om over te schakelen zijn behaald en beheerd ook de kleuren van de kubussen boven de huizen.

### **c. HomeCanvasScript.cs**

Opent het 'Home' menu als men op een huis klikt en geeft elk huis willekeurige benodigheden die de speler moet behalen om over te schakelen naar de speler.

### **d. Satisfaction.cs**

Zorgt voor de quiz en invoice pop-ups dat de speler na een bepaald interval zal ontvangen.

### **e. TarifScript.cs**

Zorgt voor functionaliteit op het 'Tariff' canvas, waar de speler zijn prijzen kan aanpassen.

## 12. /support – scripts voor support pagina

### a. AddQuestion.cs

Zorgt voor functionaliteit om een vraag toe te voegen aan een XML bestand via de support pagina

### b. MakeSuggestion.cs

Zorgt voor functionaliteit om een vraag toe te voegen aan een XML bestand via de support pagina

### c. ReportIssue.cs

Zorgt voor de functionaliteit om een vraag toe te voegen aan een XML bestand via de support pagina

### d. Support.cs

In tegenstelling tot de meeste levels, waarbij canvassen onzichtbaar worden gemaakt, zet dit script alle canvassen uit bij aanvang van de support pagina, zodat er geen overlap is met inputfields.

## 13. /tgo – scripts voor tgo level

### a. /breakout – scripts voor breakout minigame

#### i. TGOBall.cs

Geeft de bal een willekeurige starthoek en snelheid bij aanvang van het spel

#### ii. TGOBottom.cs

Checkt of de bal de onderkant van het breakout scherm raakt, en herstart zo het spel.

#### iii. TGOBreakoutController.cs

Checkt wanneer alle tegels vernietigd zijn en of de speler nog voldoende levens heeft.

#### iv. TGOBrickGrid.cs

Vult de Breakout grid met het aantal tegels gekozen door de speler.

#### v. TGOBricks.cs

Voegt punten aan de score van de speler toe indien een tegel wordt vernietigd.

#### vi. TGOPaddle.cs

Zorgt voor de beweging van de paddle en dat de paddle gelimiteerd blijft tot het speelveld.

### b. /catcher – scripts voor catcher minigame

#### i. TGOBucket.cs

Zorgt voor de beweging van de emmer en verhoogt de snelheid van de electriciteitsobjecten telkens dat de speler een electriciteitsobject kan vangen.

#### ii. TGOCatcher.cs

Zorgt ervoor dat de electriciteitsobjecten met een willekeurige snelheid, invalshoek en hoeveelheid vallen.

#### iii. TGOCatcherBottom.cs

Checkt of er electriciteitsobjecten de onderkant van het speelveld raken en trekt levens af van de speler

### c. /hangman – scripts voor hangman minigame

#### i. TGOHangman.cs

Zorgt voor alle functionaliteit in de hangman minigame.

Checkt of letter in het woord voorkomt, checkt of woord overeen komt met het antwoord ...

**d. /mastermind – scripts voor mastermind minigame**

**i. TGOMastermind.cs**

Zorgt alle functionaliteit in de mastermind minigame.

**ii. TGOMastermindDragHandler.cs**

Zorgt voor de drag functionaliteit van de kleuren.

**iii. TGOMastermindDropHandler.cs**

Zorgt voor de drop functionaliteit van de kleuren.

**iv. TGOMastermindGridFiller.cs**

Zorgt ervoor dat alle antwoorden opgeslagen worden en worden weergegeven aan de speler.

**e. /memory – scripts voor memory minigame**

**i. TGOHomeController.cs**

Zorgt voor de volledige functionaliteit van de memory minigame.  
Omdraaien van afbeeldingen, resetten van het spel ...

**f. /simonsays – scripts voor simon says minigame**

**i. TGOSimonSays.cs**

Zorgt voor de volledige functionaliteit van de simon says minigame.  
Willekeurige sequentie, checkt of sequentie van speler correct is ...

**ii. TGOSimonSaysButton.cs**

Zorgt ervoor dat knop oplicht en dat er een geluid afspeelt indien er op de knop wordt gedrukt.

**g. TGOMinigamesController.cs**

Zorgt voor de algemene werking van de TGO level.  
Uitschakelt de minigames knop uit indien er een spel wordt gestart ...

**14. AudioHandler.cs**

Checkt om wat voor 'scene' het gaat en speelt corresponderende muziek af en checkt player prefs of audio bij voorkeur af moet staan of niet.

**15. audioMute.cs**

Checkt welke texture de mute knop moet hebben en geeft functionaliteit aan de mute knop om muziek af te zetten.

**16. CameraControl.cs**

Zorgt voor de bewegingen van de camera.

**17. CameraMovement.cs**

Zet restricties op de bewegingen van de camera.

**18. CloseWindow.cs**

Zorgt voor functionaliteit om canvassen uit te zetten.

**19. CorrectOrderController.cs**

Algemeen quiz controller script voor 'Correct order' vragen.

**20. DataScript.cs**

Script dat naam en highscore van de speler bijhoudt tijdens het spelen

**21. Drive.cs**

Zorgt ervoor dat de auto's in de stad rondrijden

**22. EmailController.cs**

Script dat gebruikt wordt om e-mails te versturen naar een g-mail account omtrend suggesties, reports en vragen ingestuurd door de speler.

**23. Encryption.cs**

Encryptie script om data te encrypteren en te decrypteren

#### **24. EndOfGame.cs**

Algemeen script dat checkt wanneer de speler een level gewonnen of verloren heeft.

#### **25. HighscoreController.cs**

Nieuw highscoreController script dat alle highscores van spelers uitleest uit een XML bestand en deze als een lijst weergeeft in de 'Highscores' menu

#### **26. LevelIntroduction.cs**

Algemeen script dat gebruikt wordt om levels te introduceren

#### **27. LevelStart.cs**

Algemeen script dat gebruikt wordt om alle canvassen in een level uit te schakelen behalve het introcanvas.

#### **28. OpenWindow.cs**

Zorgt voor functionaliteit om canvassen aan te zetten.

#### **29. QnAscore.cs**

Script dat alle vragen die de speler tijdens een level beantwoord heeft, weer te geven op de 'Score' menu

#### **30. QuitGame.cs**

Script dat zorgt voor het laden van alle levels

#### **31. QuizController.cs**

Algemeen script dat zorgt voor de multiple choice vragen op alle levels.

#### **32. XMLManager.cs**

Script dat verschillende data types opslaadt naar een XML bestand. (bv. Suggesties, reports en vragen ingestuurd door de speler)

## **10 How to's**

### **10.1 Vragen toevoegen of aanpassen**

Vragen kunnen momenteel snel toegevoegd worden via de support pagina.

Zoals eerder vermeld moet de speler een vraag achterlaten over een bepaald onderwerp en moet hierbij minstens één correct antwoord en één foutief antwoord meegeven.

Ook kunnen er vragen manueel aan de bestand toegevoegd worden, maar dit valt ten zeerste af te raden. De XML structuur moet bij elke vraag overgenomen worden. Indien er hier een fout wordt gemaakt, dan zal het bestand niet meer werken en kan het spel de vragen niet meer laden.

### **10.2 Variabelen aanpassen**

Indien er variabelen in scripts publiek zijn, dan kunnen deze snel aangepast worden in de Unity Inspector. De variabelen kan men vinden in het script waarop het object is geplaatst.

Een andere manier om variabelen aan te passen is de waarden van de variabelen aanpassen in het script zelf.

### **10.3 Levels toevoegen**

Levels kan men snel toevoegen door de template level te kopiëren en in de gekopieerde level en alle aanpassingen te maken die nodig zijn voor de nieuwe level.

Ook zal men in het QuitGame.cs script een methode moeten toevoegen dat ervoor zorgt dat de nieuwe level kan worden geladen.

Er zal ook een nieuw element toegevoegd moeten worden in de levels grid van de Levels scene om de speler de mogelijkheid te geven om de nieuwe level te spelen.

Aan dit element moet dan de nieuwe methode gekoppeld worden van het QuitGame.cs script, dat men kan terugvinden op de Main Camera van de Levels scene.

#### **10.3.1 Introductie toevoegen**

Bij elke level hoort ook een introductie om de speler een idee te geven van wat de speler kan verwachten tijdens de level.

Introducties kunnen snel toegevoegd worden door dezelfde opmaak te gebruiken als de andere introductie bestanden die men kan terugvinden in de resources map van het project.

## 11 Uitbreidings mogelijkheden

Het spel bevat nog mogelijkheden voor uitbreiding. Zoals eerder vermeld kunnen er snel nieuwe levels toegevoegd worden of kan de huidige functionaliteit op verschillende manieren verbeterd worden.

### 11.1 Support pagina

De functionaliteit van de support en de admin pagina kan zeker en vast nog uitgebreid worden. Momenteel worden de vragen die de gebruiker toevoegd, zonder confirmatie toegevoegd aan het vragen bestand.

Dit zou uitgebreid kunnen worden door ze op de admin pagina eerst te confirmeren en pas daarna toe te voegen aan het vragen bestand.

### 11.2 Highscores

Ook de highscores zouden eventueel een aanpassing kunnen gebruiken.

Momenteel wordt er enkel nagekeken of de username, die na de login pagina opgeslagen wordt, reeds aanwezig is in het highscore bestand, nadien worden alle scores aan deze naam toegevoegd.

Dit is natuurlijk niet ideaal, want als er twee personen met dezelfde naam het spel spelen, wordt enkel de highscore van de persoon met de hoogste highscore.

Twee personen die dezelfde username ingeven, hebben dus geen afzonderlijke highscore.

Door eventueel ook gebruik te maken van een paswoord, kunnen deze gebruikers zichzelf wel onderscheiden.

### 11.3 Encryptie & decryptie

Doordat de huidige item bestanden (highscores, problemen, suggesties, vragen ...) niet geëncrypteerd worden opgeslaan, lopen deze bestand risico op ongewenste aanpassingen en lezingen van buitenaf.

Highscores die usernames en scores bevatten kunnen snel aangepast of bekeken worden door andere gebruikers.

Vragen kunnen potentiëel gewijzigd worden zodat ze allemaal foutive antwoorden bevatten, waardoor de speler ook foutive informatie krijgt aangeleerd.

Een mogelijke oplossing zou zijn om een vorm van encryptie te gebruiken, waardoor ongewenste aanpassingen door gebruikers vermeden kunnen worden.

### 11.4 Verschillende talen toevoegen

Een andere uitbreidingsmogelijkheid dat toegevoegd kan worden is het implementeren van verschillende talen. Momenteel staan alle introducties van alle levels in het engels in XML bestanden. Er zou dus eventueel een taalkeuze kunnen toegevoegd worden aan de PlayerPrefs, waarbij de speler een taal naar zijn keuze kan kiezen en ervoor zou zorgen dat de introducties van alle levels in deze taal worden weergegeven. Dit zorgt er wel voor dat er meerdere XML bestanden aanwezig moeten zijn in deze verschillende talen.

## 12 Bibliography

- Bostock, M. (2012, 01 14). *Fisher-Yates Shuffle*. Opgehaald van bostocks: <https://bost.ocks.org/mike/shuffle/>
- dontnetperls. (sd). *Any*. Opgehaald van dontnetperls: <https://www.dotnetperls.com/any>
- Elia. (sd). *Elia News*. Opgehaald van Elia: <http://www.elia.be/nl/>
- Entsoe. (sd). *Entsoe*. Opgehaald van Entsoe: <https://www.entsoe.eu/>
- Games, K. (2014, 12 12). *Unity UI Drag and Drop tutorial*. Opgehaald van Youtube: <https://www.youtube.com/watch?v=c47QYgsJrWc>
- gamesplusjames. (2016, 03 15). *Mobile Simon Game Tutorial Playlist*. Opgehaald van Youtube: [https://www.youtube.com/watch?v=PqkBuU08MOQ&list=PLiyfvtjWC\\_XccolAYZMnomMeehR0O2Be](https://www.youtube.com/watch?v=PqkBuU08MOQ&list=PLiyfvtjWC_XccolAYZMnomMeehR0O2Be)
- GeeksforGeeks. (sd). *Shuffle a given array using Fisher-Yates shuffle Algorithm*. Opgehaald van GeeksforGeeks: <https://www.geeksforgeeks.org/shuffle-a-given-array-using-fisher-yates-shuffle-algorithm/>
- graslany. (2014, 11 26). *Change music between scenes*. Opgehaald van Unity Answers: <https://answers.unity.com/questions/838467/change-music-when-changing-scene.html>
- grenade. (2009, 08 11). *Randomize a List<T>*. Opgehaald van stackoverflow: <https://stackoverflow.com/questions/273313/randomize-a-listt>
- Microsoft. (sd). *DateTime.Now Property*. Opgehaald van Microsoft: <https://docs.microsoft.com/en-us/dotnet/api/system.datetime.now?view=netframework-4.7.2>
- Net-informations. (sd). *How to send email from c#*. Opgehaald van Net-informations: <http://csharp.net-informations.com/communications/csharp-smtp-mail.htm>
- News, S. (2017, 07 25). *Unity UI Tutorial Grid Layout and Scroll Window*. Opgehaald van Youtube: <https://www.youtube.com/watch?v=Vylo5tINNeA>
- Peeters, L. (sd). *De digitale energiemeter*. Opgehaald van vlaanderen: <https://www.vlaanderen.be/bouwen-wonen-en-energie/zelf-energie-producereen/zonnepanelen-en-zonneboilers/de-digitale-energiemeter>
- Polygons, T. P. (2019, 11 30). *Unity Cross Scene Music/Audio and Mute Button*. Opgehaald van Youtube: <https://www.youtube.com/watch?v=OeZuwgG6HJM>
- property news team. (2018, 02 16). *Smart meters vs normal meters*. Opgehaald van Zoopla: <https://www.zoopla.co.uk/moving/buyers-guide/energy/smart-meters-vs-normal-meters/>
- techopedia. (sd). *Simple Mail Transfer Protocol*. Opgehaald van techopedia: <https://www.techopedia.com/definition/1710/simple-mail-transfer-protocol-smtp>
- The lockergone daily report. (2011, 09 14). *Traditional Electric Meter Vs. Smart Meter*. Opgehaald van lockergnome: <https://lockergnome.com/2011/09/14/traditional-vs-smart-meter/>
- Unity. (2019, 03 13). *Input.GetMouseButtonDown*. Opgehaald van Unity: <https://docs.unity3d.com/ScriptReference/Input.GetMouseButtonDown.html>
- Unity. (sd). *2D Roguelike tutorial*. Opgehaald van Unity: <https://unity3d.com/learn/tutorials/s/2d-roguelike-tutorial>
- Unity. (sd). *Extension Methods*. Opgehaald van Unity: <https://unity3d.com/learn/tutorials/topics/scripting/extension-methods>
- Unity. (sd). *Roll-a-ball tutorial*. Opgehaald van Unity: <https://unity3d.com/learn/tutorials/s/roll-ball-tutorial>
- Unity. (sd). *Tutorials*. Opgehaald van Unity: <https://unity3d.com/learn/tutorials>
- Viessmann. (sd). *Slimme meters*. Opgehaald van Viessmann: <https://www.viessmann.be/nl/Informatie-voor/Huiseigenaars/slimme-meters.html>

- Watson, M. (2009, 07 24). *Does C# have an "Or" list? eg if Variable == (1 || 2 || 3) {DoSomething();}*. Opgehaald van Microsoft Developer Network: <https://social.msdn.microsoft.com/Forums/en-US/7b393687-343b-4e63-bff3-7bc3ac46d368/does-c-have-an-quot-or-quot-list-eg-if-variable-1-2-3-dosomething?forum=csharp-language>
- Wikipedia. (sd). *Automatic meter reading*. Opgehaald van Wikipedia: [https://en.wikipedia.org/wiki/Automatic\\_meter\\_reading](https://en.wikipedia.org/wiki/Automatic_meter_reading)
- Wikipedia. (sd). *Elia (netbeheerder)*. Opgehaald van Wikipedia: [https://nl.wikipedia.org/wiki/Elia\\_\(netbeheerder\)](https://nl.wikipedia.org/wiki/Elia_(netbeheerder))
- Wyck. (2009, 05 19). *Shorthand for multiple ORs*. Opgehaald van Microsoft Developer Network: <https://social.msdn.microsoft.com/Forums/en-US/2bba32df-fa2a-4749-8505-5b8eddc4fe9c/shorthand-for-multiple-ors?forum=csharp-language>