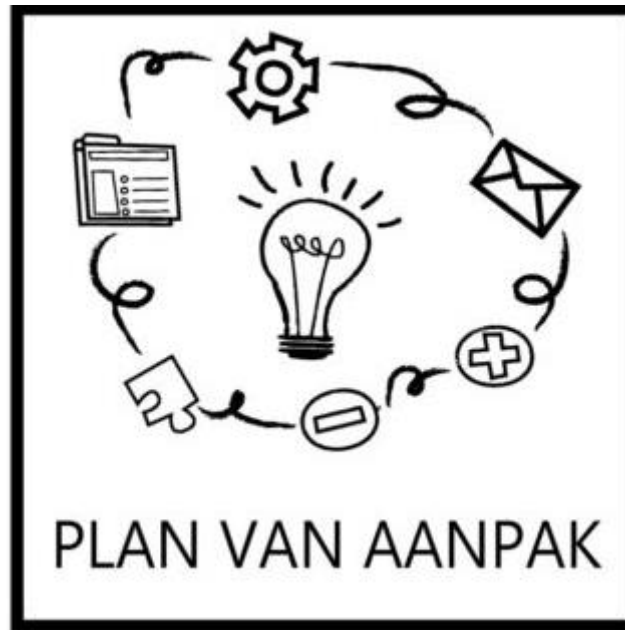


# Plan van Aanpak

## Pixel Perfect



Groep 24 | IPIXPE | 05-12-2023

## Inhoud

Inleiding .....	3
Projectdoel .....	4
Scope .....	4
Doelgroep.....	4
Risico's en kansen .....	5
Delivarables .....	6
Product Breakdown Structure (PBS) .....	9
Work Breakdown Structure (WBS) .....	10
Planning en verantwoording (logboek) .....	11
Verantwoording gemaakte keuzes .....	14
Projectdocumentatie (Handleidingen).....	15
<b>Handleiding website Jelle:</b> .....	17
<b>Handleiding Puzzel Jelle:</b> .....	18
Handleiding puzzel Emma (gebruiker): .....	19
Handleiding puzzel Emma (beheerder): .....	20
Handleiding website Emma (gebruiker): .....	21
Handleiding Emma (beheerder): .....	22
<b>5.</b> 23	
<b>Handleiding Puzzel Bjorn:</b> .....	25
Navigatie:.....	28
Navigatie:.....	29
Literatuur .....	43

## **Inleiding**

Dit is het plan van aanpak van een groepsproject voor een module voor eerstejaars informaticastudenten die worden geïntroduceerd aan HTML, CSS, Javascript en User Experience. Het doel van het project is om binnen 6 weken 5 puzzels te demonstreren, waarbij de oplossingen leiden tot een specifieke letter/cijfercombinaties, die kan worden ingevuld op een groepswebsite, die na correct ingevoerd te zijn doorverwijst naar de verschillende individuele websites die geschreven zijn in HTML, CSS en javascript. De scope wordt bepaald om duidelijk te maken aan welke eisen het project moet doen. Ook is de doelgroep vastgesteld waarvoor de puzzels en de websites zijn gemaakt. Risico's en kansen zijn tegen elkaar gezet en alle deliverables op Gradework zijn in een tabel gezet zodat duidelijk is wat er precies moet worden ingeleverd. Een work breakdown structure en een project breakdown structure zijn gemaakt om in een overzichtelijk schema alle taken te weergeven. Ook is er een logboek bijgehouden om te zien wie wat en wanneer heeft gedaan. Ten slotte zijn in dit verslag de handleidingen voor de websites te vinden.

## Projectdoel

Het doel van het project is om binnen 6 weken 5 puzzels te demonstreren, waarbij de oplossingen leiden tot een specifieke letter/cijfercombinaties, die kan worden ingevuld op een groepswebsite, die na correct ingevoerd te zijn doorverwijst naar de verschillende individuele websites die geschreven zijn in html, css en javascript.

## Scope

In de Scope wordt er duidelijk gemaakt wat er gemaakt wordt, welke activiteiten worden verricht en ook wat er juist niet gedaan gaat worden.

De opdracht is om in groepsverband een escaperoom ervaring te maken, door een website te designen in HTML en CSS. Deze website moet een code hebben waarvan je de combinatie krijgt door een aantal puzzels op te lossen. Door elke groepslid wordt een puzzel gemaakt door hardware te koppelen aan pythoncode die wordt uitgevoerd op een Raspberry pi. Zodra de code gekraakt is kan je navigeren naar individueel gemaakte websites van de groepsleden. De opdracht wordt uitgevoerd om te leren op het gebied van design, hardware en persoonlijk leiderschap.

## In Scope

In scope is het totaal van de op te leveren producten en uit te voeren activiteiten. Deze zijn hieronder op een rij gezet:

Groepswebsite geschreven in HTML en CSS.  
5 individuele websites, geschreven in HTML en CSS  
5 hardware puzzels, gemaakt op een breadboard aan de Raspberry pi, gecodeerd in Python  
Styleguides van de websites  
Een Gantt-Chart met de planning  
5 vergaderingen waarbij ieder groepslid de leiding neemt

## Out of Scope

Er zijn ook een aantal zaken waar het product niet aan hoeft te voldoen, namelijk:

- De website wordt niet meer bijgewerkt na oplevering
- De puzzels worden niet gerepareerd of bijgewerkt na oplevering

## Doelgroep

De bepaalde doelgroep is gebaseerd op wie het project te zien krijgen en de puzzels gaan oplossen. Dit zijn voornamelijk eerstejaars hbo-informaticastudenten en docenten van de studie. De leeftijd van de studenten ligt ongeveer tussen de 17 en 27 jaar en het gros van de docenten zijn boven de 40 jaar oud.

Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)

## Risico's en kansen

Elk project brengt risico's, maar ook kansen met zich mee. De risico's worden in een versimpelde risicoanalyse in de tabel hieronder weergegeven.

<b>Risico</b>	<b>Beheersmaatregel</b>
Niet werkende hardware onderdelen	Vraag nieuwe bij het techlab van de hogeschool. Koop eventueel nieuwe online of leen van andere groepsleden
Groepslid die stopt met de module/studie	Verdeel de taken van het afgefallen groepslid over de rest van de groepsleden. In het geval dat er meer groepsleden uitvallen kan er eventueel gevraagd worden aan de docenten of het mogelijk met een andere groep samen te werken
Communicatieproblemen	Implementeer een effectief communicatieprotocol, gebruik samenwerkingshulpmiddelen en stimuleer open communicatie binnen de groep.
Scopecreep	Documenteer de projectscope duidelijk en implementeer een wijzigingsbeheerproces om nieuwe eisen te evalueren voordat ze worden geaccepteerd.
Onvoldoende testen	Plan voldoende tijd in voor grondige tests op verschillende niveaus en betrek het team bij het testproces.
Afwezigheid van documentatie	Moedig het team aan om tijdens elke fase documentatie bij te houden en leg nadruk op het belang van gestructureerde code en documentatie.

De kansen die dit project met zich mee brengt zijn:

- Leren om gestructureerd te werken aan een goede gebruikerservaring.
- Creatief leren omgaan met oplossingen bedenken voor technische uitdagingen.
- Ervaring opdoen in HTML, CSS en Javascript.
- Beter leren werken in groepsverband.
- Persoonlijk leiderschap verbeteren en professioneel omgaan met meetings, emails en deadlines.

## Deliverables

Hier onder volgt een lijst van alle, zowel gezamenlijke als individuele, GradeWork deliverables per opdracht voor het portfolio.

<i>Opdracht 1: Ontwerp van de websites</i> Styleguide (PDF)	Individueel
Styleguide (PDF)	Gezamenlijk
Groepswebsite (ZIP)	Gezamenlijke
Individuele website (ZIP)	Individueel

### *Opdracht 2: Onderbouwing van het ontwerp van de websites*

Tekstuele onderbouwing van de ontwerpkeuzes voor de groepswebsite (PDF)	Gezamenlijk
Tekstuele onderbouwing van de ontwerpkeuzes voor de individuele websites (PDF)	Individueel
Pagina met onderbouwingsdocumenten en, zowel gezamenlijk als de individuele, bijbehorende styleguides (Webpagina)	Individueel

### *Opdracht 3: Individuele website*

De volgende componenten verdeeld over een of meerdere webpagina's van je individuele website: <ul style="list-style-type: none"><li>- Gantt Chart planning (webpagina)</li><li>- Plan van aanpak (webpagina)</li><li>- Brainstormdocumenten en gemaakte keuzes (webpagina)</li><li>- Vergaderdocumentatie (webpagina)</li><li>- Professionele e-mail (webpagina)</li><li>- Feedbackdocumenten (webpagina)</li><li>- Tijdlijn (webpagina)</li></ul>	
--	--

*Opdracht 4: Planning in de Gantt-chart*

Planning in de vorm van een Gantt-chart (PDF)	Gezamenlijk
---	-------------

*Opdracht 5: Plan van aanpak en documentatie*

Plan van aanpak (PDF)	Gezamenlijk
-----------------------	-------------

*Opdracht 6: Brainstormdocumentatie en gemaakte keuzes*

Een document (PDF) met daarin: <ul style="list-style-type: none"><li>- Beschrijving van het proces</li><li>- Ervaringen vanuit de brainstorm</li><li>- Afbeelding van het resultaat van de brainstorm</li></ul>	Gezamenlijk
---	-------------

*Opdracht 7: Vergaderdocumentatie*

Een document (PDF) met daarin: <ul style="list-style-type: none"><li>- De vergaderagenda</li><li>- Notulen (inclusief een overzicht met concrete actiepunten met deadlines)</li><li>- Pluspunten, verbeterpunten en gebruikte gesprekstechnieken die je gaat inzetten in de volgende module.</li></ul>	Gezamenlijk (m.u.v de plus/verbeterpunten en de gesprekstechnieken)
--	---

*Opdracht 8: Een professionele e-mail*

Een document (PDF) met daarin: <ul style="list-style-type: none"><li>- De eerste versie van de mail</li><li>- De feedback die is ontvangen van een medestudent &amp; jouw eigen beoordeling van de gekregen feedback.</li><li>- De definitieve versie van de mail waarin de ontvangen feedback is verwerkt.</li></ul>	Individueel
---	-------------

*Opdracht 9: Feedbackdocumenten*

Een document (PDF) met daarin: <ul style="list-style-type: none"><li>- Het ingevulde feedbackformulier</li><li>- YouTube link naar de video</li></ul>	Individueel (m.u.v. de link, die is gezamenlijk)
---	--

*Opdracht 10: Tijdlijn*

De interactieve tijdlijn op ieders individuele website, welke voldoet aan de volgende eisen: <ul style="list-style-type: none"><li>- Het bevat opdracht 4 t/m 9 + 12</li><li>- Het is gemaakt met HTML, CSS en Javascript</li><li>- Je kan met de muis over een bestand (van een opdracht) zweven om het zichtbaar te krijgen</li></ul>	Individueel
---	-------------

*Opdracht 11: Groepswebsite*

Groepswebsite (ZIP)	Gezamenlijk
---------------------	-------------



# Product Breakdown Structure (PBS)

in dit overzicht is te zien aan welke eisen de opdrachten moeten voldoen.



# Work Breakdown Structure (WBS)

In het volgende overzicht is te zien welke subtaken elke opdracht heeft.



Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)

## Planning en verantwoording (logboek)

ID:	Product:	Deelproduct:	Duur:	Teamlid:	Datum:	Afwijking van planning:
1	PBS	Opzetten vijf hoofdcategorieën	0.5H	Emma	09-12-2023	n.v.t
2	Vergadering	Eerste vergadering 12/12	0.5H	Hele Team	12-12-2023	n.v.t

3	PBS	Onderverdelen van deliverables onder de vijf hoofdcategorieën.	0.5H	Emma	09-12-2023	Twee dagen later af dan gepland.
4	Planning	Actiepunten van de eerste vergadering verwerken in de YouTrack Gantt chart.	0.5H	Emma	12-12-2023	n.v.t.
5	Moodboard	Toevoegen eerste ideeën voor vormgeven puzzel en kleurenpaletten.	1.5H	Iedereen	12-12-2023	n.v.t
6	Deliverables	Schema's met deliverables per opdracht opzetten	1H	Emma	12-12-2023	n.v.t
7	Vergadering	Tweede vergadering	0.5H	Iedereen	19/12/2023	n.v.t
8	Styleguides	Gezamenlijk en alle individuele	2.5H	Iedereen	18-12-2023	n.v.t
9	Deskresearch over Scope	PVA kopje 'Scope'	1H	Jelle	13/12/2023	n.v.t
10	Puzzel	Prototypes maken	4H	Jelle	20/12/2023	n.v.t.
11	Feedback verwerken	Individuele styleguide verbeterd met behulp van de ontvangen feedback van medestudenten in FeedbackFruits.	0.5H	Emma	22-12-2023	n.v.t
12	Individuele website	Navigatiebar implementeren	2H	Emma	22-12-2023	n.v.t
13	Individuele website	Secties opzetten en start tijdlijn maken	5.5H	Emma	22-12-2023	n.v.t
13	Individuele website	Html structuur semantisch maken	2H	Jelle	04-01-2024	n.v.t.
14	Individuele website	Downloadlinks in tijdlijn laten werken + achtergrond website maken	7H	Emma	26-12-2023	
14	Puzzel	Redesign en code schrijven	8H	Jelle	14-01-2024	Puzzel had al af moeten zijn maar redesign was nodig
15	Individuele website	CSS voor de secties van website toegevoegd en tijdlijn afgemaakt	6H	Emma	15-01-2024	n.v.t
15	Puzzel	Code puzzel afmaken	8H	Jelle	19-01-2024	n.v.t.

Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)

16	Individuele website	CSS toepassen aan website	8H	Jelle	11-01-2024	n.v.t.
17	Individuele website	Meer CSS toepassen	8H	Jelle	16-01-2024	n.v.t.
18	Individuele website	Meer CSS, responsive maken en navigatie verbeteren	8H	Jelle	17-01-2024	n.v.t.
19	Puzzel testen	Puzzel getest op gebruiksvriendelijkheid door doelgroep	4H	Emma	19-01-2024	n.v.t.
19	Handleidingen	Handleidingen schrijven	2H	Emma	21-01-2024	n.v.t.
19	PVA	Netjes maken zodat het op te leveren is	2H	Jelle	21-01-2024	Had eerder moeten maar was ziek, wel af voor de nieuwe deadline.
20	Brainstormdocumentatie	Samenvoegen verzamelde brainstormdocumenten en beschrijvingen bijvoegen	3H	Emma	24-01-2024	Dit stond eerder op de planning, maar aangezien we zo recent als gisteren nog extra documenten hebben verzameld doe ik dit nu.
21	Presentatie voorbereiden	Presentatie opzetten en doorspreken	2H	Iedereen	25-01-2024	
22	Groepswebsite	Volledig functionerende groepswebsite gemaakt	10H	Iedereen	24-01-2024	
23	Gantt-Chart maken	Begin gemaakt Gantt-Chart	1H	Kevin	08-12-2023	
24	Gantt-Chart afronden na feedback	Op basis van feedback de Gantt-Chart aangepast en afgerond	0.5H	Kevin	17-12-2023	
25	Puzzel bedenken en basis maken	Puzzel bedacht en basis gemaakt	5H	Kevin	09-1-2024	
26	Begin gemaakt met JavaScript op groepswebsite	Basis gemaakt van JavaScript code voor inloggen met puzzelcode	1.5H	Kevin	15-12-2023	
27	Individuele Website	Basis gemaakt van individuele website	7H	Kevin	13-12-2023	

Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)

28	Individuele Website	Styling toegevoegd op de individuele website	4H	Kevin	14-12-2023	
29	Individuele Website	Documenten op individuele website plaatsen	2H	Kevin	25-1-2024	
30	Bug fixes op groepswebsite	Bugs opgelost die op de groepswebsite voorkwamen	3H	Kevin	25-1-2024	
31	Handleiding	Handleidingen gemaakt	0.5H	Kevin	25-1-2024	
32	Meeting	Vorbereiding meeting	1H	Jordy	9-12-2023	
33	swot	SWOT-analyse	1H	Jordy	12-12-2023	
34	Style guide	Verantwoording style guide	1H	Jordy	18-1-2024	
35	Handleiding	Handleiding gemaakt	2H	Jordy	25-1-2024	

## Verantwoording gemaakte keuzes

### *Thema:*

Omdat onze doelgroep best jong is, maar de docenten wat ouder, moest er een thema gekozen die beide leeftijdscategorieën aanspreekt. De jonge generatie houdt overduidelijk van games, en docenten die de groei van Nintendo consoles hebben meegemaakt kunnen ook nostalgisch worden van de oude generaties van de Gameboy.

### *styleguides:*

Omdat ons thema 8-bit nintendo was, lag het voor de hand om kleuren te kiezen voor de websites die veel zijn gebruikt in Nintedoproducten. Zo is er gekeken naar elke generatie van Nintendo-consoles zoals de Gameboy, Gameboy Advanced en Nintende DS. Het gekozen lettertype is een font-family die heel blokkig is dus lijkt alsof het uit grote pixels bestaat.

## Projectdocumentatie (Handleidingen)

### Gebruikershandleiding (Groeps Website):

1. Als je de pagina opent kom je op een scherm dat lijkt op een Game Boy, hierin moet je op het groene scherm de code invullen die je van alle puzzels bij elkaar krijgt. (Als er op de doos "1" staat dan moet je die code op positie 1 zetten.)
2. Als je de correcte code hebt ingevuld dan krijg je een scherm te zien met 5 links van mensen hun namen, achter elke naam (link) zit een naam.
3. Als je op iemands naam klikt wordt je succesvol doorgestuurd naar een individuele pagina!

### Beheerdershandleiding (Groepswebsite):

1. Op de index pagina (beginscherm) staat er een form, hierin moet je de geheime code invullen.

```
<section class="green-screen">
  <label for="hiddenInput"></label>
  <form class="myForm" name="myForm" onsubmit="return validateForm()" method="post">
    <input type="text" id="hiddenInput" name="password" placeholder="Vul de geheime code in...">
  </form>
</section>
```

Zodra de gebruiker het doorstuurt wordt het naar de JavaScript functie validateForm() gestuurd.

2. Bij deze JavaScript functie haalt het de waarde van de name "password" op, als het wachtwoord leeg is krijg je een foutmelding dat je niks hebt ingevuld. En als het wachtwoord niet overeenkomt dan zie je een melding dat het wachtwoord fout is. En als het wachtwoord wel overeenkomt dan wordt je doorgestuurd naar de pagina waar alle individuele websites staan. Ook kan je het wachtwoord veranderen bij de else if doormiddel van: (password === "gekozen wachtwoord").

```
1+ usages Kevin
function validateForm() : boolean {
  event.preventDefault();

  const password = document.forms["myForm"]["password"].value;

  if (event.key === "Enter") {
    event.preventDefault();
    validateForm();
  }

  if (password === "") {
    alert("Vul een wachtwoord in.");
    return false;
  } else if (password === "AY8X3") {
    window.location.href = "paginas.html";
    return false;
  } else {
    alert("Fout wachtwoord.");
    return false;
  }
}
```

3. Als je probeert om op de pagina te komen met de website links gaat de JavaScript controleren of je ooit al het juiste wachtwoord hebt ingevuld, als dat niet zo is wordt je teruggestuurd naar de inlog pagina.

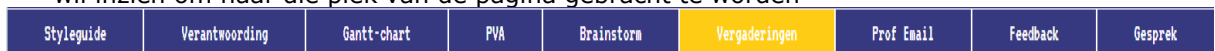
```
1+ usages Kevin
function checkAccessOnPaginas() : void {
  const toegang : string = sessionStorage.getItem(key: "toegang");

  if (toegang === false) {
    window.location.href = "index.html";
  }
}
```

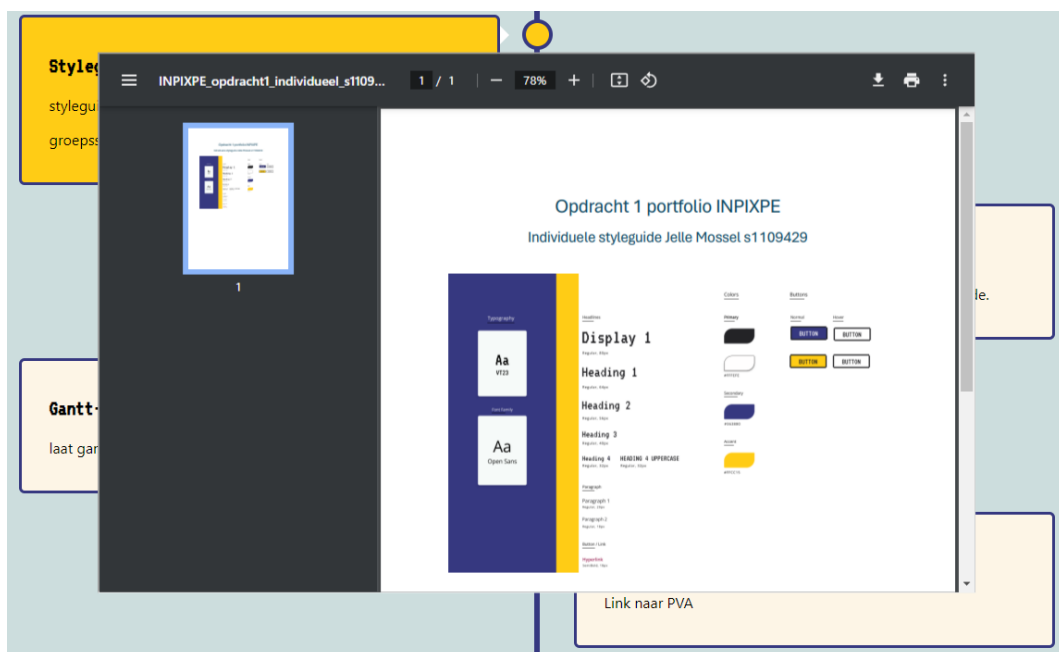


## Handleiding website Jelle (gebruiker):

1. Bovenaan de pagina is een navigatiebalk te vinden. Klik op een van de onderwerpen die je wil inzien om naar die plek van de pagina gebracht te worden



2. Hover je muis over een tekstbubbel om het bijbehorende document in te zien



3. Lees het gewenste document.

## Handleiding website Jelle (beheerder):

IDE: VSCode

Taal: HTML (index.html), CSS (style.css), JS (main.js)

Files: index.html, style.css, main.js, .pdf-files

### HTML:

#### De Head:

In de Head staat alles wat in het HTML bestand geladen moet worden, denk aan style sheet, javascript en fonts die ingeladen moeten worden. ook de titel staat in de head.

#### Body:

De body bestaat uit een navigatiebalk en de main.

#### Header/Navigatie:

De volledige navigatie is een unlisted item waarvan elk blokje een <li> tag heeft met daarin een <a> tag om te linken naar een tekstbubbel van de tijdlijn. Deze staat in de header.

Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)

**Main:**

Elke section is hetzelfde opgebouwd. Het is 1 tekstvak, met <h2> als titel, <p> met tekstuele beschrijving van de titel en een <iframe> met het document die het tekstvak beschrijft.

**Footer:**

In de footer staat door wie de website is gemaakt.

**CSS:**

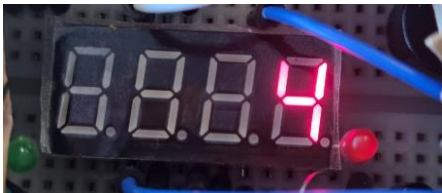
Voor de CSS is een styleguide gevolgd die te vinden is op de website.

**JS:**

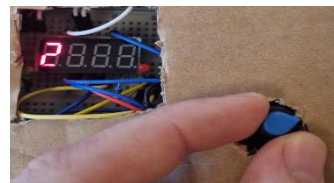
de enige javascript die is gebruikt is een functie die zorgt dat als je over een tekstbubbel hovert, je de iframe goed zichtbaar krijgt in het midden van je beeldscherm. Er is een for-loop gemaakt zodat deze functie kan worden hergebruikt voor elke tekstbubbel zonder alle id's apart in een variabele te zetten.

**Handleiding Puzzel Jelle (gebruiker):**

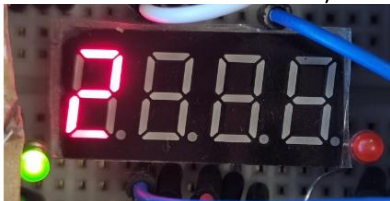
1. Sluit de raspberry pi aan aan de stroom. De puzzel start automatisch na enkele seconde.
2. Er start een countdown van 5 seconde op de multi-segment display:



3. Zodra de counter op 0 staat, hoor je een aantal tikjes uit de puzzeldoos komen. Luister hier goed naar. Als de tikjes voorbij zijn, draai je aan de rotary encoder en zie je het getal op de display omhoog of laag gaan. Als je denkt dat het getal gelijk is aan het aantal tikjes dat je hoorde. Druk je op de rotary encoder.



- 4.1 Als het antwoord correct is, brandt er een groen lampje:



- 4.2 Bij een fout antwoord, brandt er een rood lampje:



5. Als je 4 maal op rij goed antwoordt, verschijnt er een geheime code!

**Handleiding puzzel Jelle (beheerder):****Puzzel onderdelen:**

Raspberry Pi, breadboard, buzzer, multi-segment display, rotary encoder, rode LED, groene LED.

Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)

### **Pin configuratie:**

Om overzichtelijk te maken welke onderdelen aan welke GPIO pin zit aangesloten is er gebruik gemaakt van Enum classes. Hierin zijn alle onderdelen benoemd in een eigen class en zijn alle segmenten benoemd met bijbehorende GPIO pin. Zo zijn ze makkelijk aan te roepen met hun Enum-naam in plaats van het getal van de pin.

### **Multi Segment Display:**

Voor dit display is een kaart gemaakt. De cijfers en letters (die gebruikt worden) krijgen hier hun eigen teken toegekend aan alle segmenten van de display die stroom nodig hebben om aan te gaan.

Omdat het niet lukte om verschillende getallen tegelijk op de display te weergeven is er een while-loop gemaakt die zo snel aan en uit gaat dat de illusie wordt gewekt dat er 2 verschillende getallen worden weergegeven. Ook wordt er gebruik gemaakt van threading omdat de loop van de display werd onderbroken als de main-loop wordt gestart.

### **Buzzer:**

Voor de buzzer is er gebruik gemaakt van een library die zorgt dat het altijd een random aantal beeps geeft. De polariteit is zo gezet dat hij het dubbele aantal voltage krijgt (van 3.3V naar -3.3V) zodat het dubbel zoveel geluid kan maken. 0 en 1 beep aantal is eruit gehaald omdat dit niet gebruiksvriendelijk is.

### **Rotary Encoder:**

Er is ingesteld dat de input van de encoder 0.5 omhoog gaat omdat elke draai 2 inputs geeft. Zo wordt gezorgd dat er het getal op de display maar met 1 omhoog gaat. De knop indrukken is ingesteld als een soort select of oke knop.

### **Main functie:**

In de Main wordt eerst de thread van de display aangeroepen, zodat deze niet wordt onderbroken. Daarna start de main thread, er start een countdown van 5 tot 0.

Beep functie wordt aangeroepen die een willekeurig aantal beeps geeft.

Er wordt gewacht op een input van de rotary encoder, deze is goedgekeurd bij een klik op de encoder.

Als je input gelijk is aan het aantal beeps, brandt er een groene LED, fout is een rode LED.

Er wordt 4 maal doorheen geloopt, bij een fout gaat de teller weer op 0 en moet je opnieuw beginnen.

4 maal op rij correct geeft een succes signaal, de lampjes knipperen om en om en het programma sluit af.

### Handleiding puzzel Emma (gebruiker):

1. Sluit de Raspberry Pi aan op een stopcontact. Het apparaat en het spel zullen na 20-30 seconden automatisch opstarten.



2. Druk een keer op de groene knop om het spel te starten



3. Er gaat nu ongeveer elke seconde, willekeurig een van de LED lampjes aan.



4. Druk zo snel mogelijk op de knop met dezelfde kleur als het LED lampje om een punt te scoren.



5. Dit proces van een het branden van een LED en het moeten drukken op de bijbehorende knop herhaalt zich nog 19 keer.
6. Na afloop van het spel zullen de LED lampjes uitgaan, en kan je in het kijkgat bovenin de doos een '-' ofwel een ander teken, wat een onderdeel is van de wachtwoordsleutel die nodig is op de website, zien.



#### Handleiding puzzel Emma (beheerder):

1. Sluit de Raspberry Pi aan op een stopcontact. Het apparaat en het spel zullen na 20-30 seconden automatisch opstarten. Het script hier onder heeft er voor gezorgd dat het spel zich vanzelf opstart.



2. De Raspberry Pi krijgt stroom via deze kabel die aangesloten is aan de Raspberry Pi.



3. In de doos is het volgende aangesloten op het breadboard:
  - a. 3x LED (Groen, geel, rood)
  - b. 3x Knop (Groen, geel rood)
  - c. 1x 7-segment display
4. In de code voor de puzzel wordt er gebruik gemaakt van de gpiozero library, de time module en de random module voor het willekeurig laten genereren van elk spel.

```
import time
import random
from gpiozero import Button, LED, OutputDevice
```

5. In de functie setup() hier onder is duidelijk te zien op welke GPIO pinnen van de Raspberry Pi de verschillende knoppen en LED's zijn aangesloten. Dit zijn namelijk de blauwe getallen tussen de haakjes in de variabelen voor de verschillende LED's en buttons.

```

def setup():
    button_3 = Button(19) # Rood links
    led_3 = LED(5)

    button_2 = Button(13) # Geel
    led_2 = LED(22)

    button_1 = Button(26) # Groen
    led_1 = LED(27)

    leds = [led_1, led_2, led_3]
    buttons = [button_1, button_2, button_3]

    segmenten = [OutputDevice(pin) for pin in (25, 24, 12, 20, 21, 23, 18)]
    for segment in segmenten:
        segment.off()

    return leds, buttons, segmenten, button_1, button_3

```

6. In de functie `play_game()` bevindt zich alle logica omtrent het spel zelf. Er wordt 20 keer zeer kort een van de lampjes aangezet. Vervolgens krijgt de gebruiker 2 seconden de tijd om de bijbehorende knop in te drukken. Als dit lukt heeft de gebruiker een punt gescoord, als dit niet is gelukt heeft de gebruiker geen punt gescoord. Ik beide gevallen gaat na de 2 seconden het volgende lampje aan. Indien de gebruiker alle punten heeft gehaald wordt het teken van de geheime code weergegeven op de 7-segment display in de doos. Zo niet, dan weergeeft de 7-segment display een `.`.

```

def play_game(leds, buttons, segmenten):
    score = 0
    for compartement in range(20):
        index = random.randint(a=0, b=2)
        led = leds[index]
        button = buttons[index]

        led.on()
        time.sleep(0.8)
        led.off()

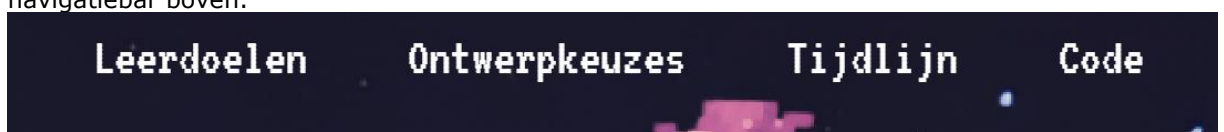
        start_time = time.time()
        while time.time() - start_time < 2:
            if button.is_pressed():
                score += 1
                time.sleep(0.4)
                break

        if score == 20:
            code = (25, 12, 21, 23)
            for segment in segmenten:
                if segment.pin.number in code:
                    segment.on()
                else:
                    segment.off()
        else:
            segmenten[-1].on()

```

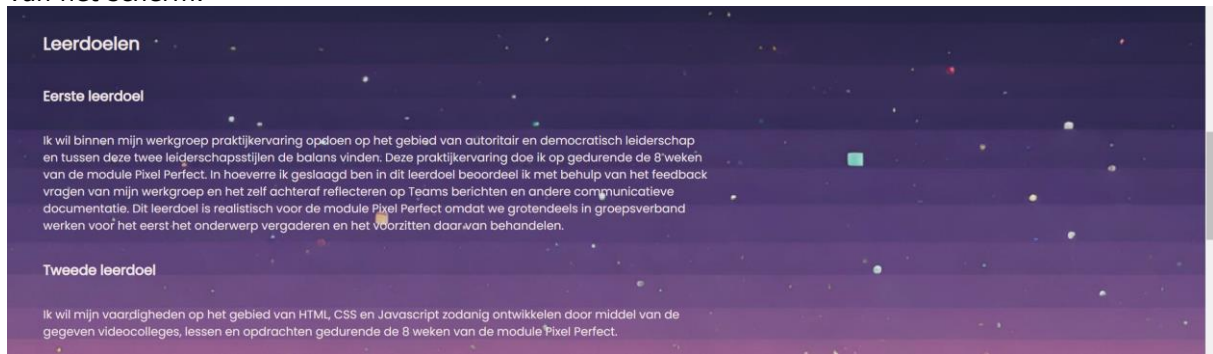
Handleiding website Emma (gebruiker):

1. Navigeer naar de gewenste plek op de website door met de muis er op te klikken in de navigatiebar boven.



Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)

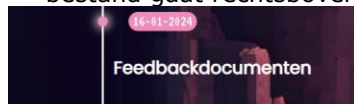
2. Scrol naar boven of naar beneden met behulp van uw muiswiel of de grijze scrol-balk rechts van het scherm.



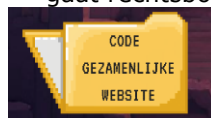
3. Navigeer op elk gewenst moment weer volledig naar boven door op de witte pijltjes rechtsonder in de pagina te klikken.



4. Download items uit de tijdlijn door met je cursor over het gewenste document te zweven waardoor deze iets groter zal worden en druk vervolgens de rechtermuisknop in. Het bestand gaat rechtsboven in je browserscherm downloaden.



5. Om een van de .zip bestanden te downloaden met de code van de websites beweeg je met je muis over het gewenste mapje en druk je vervolgens de rechtermuisknop in. Het bestand gaat rechtsboven in je browserscherm downloaden



#### Handleiding Emma (beheerder):

1. De website bestaat uit verschillende HTML sections met daarin de leerdoelen, ontwerpkeuzes, de tijdlijn en de code van de websites.
2. De navigatiebar werkt door de a-tags direct te linken naar de desbetreffende section van de pagina.

```
<nav class="navigation">
  <a href="#section1">Leerdoelen</a>
  <a href="#section2">Ontwerpkeuzes</a>
  <a href="#section3">Tijdlijn</a>
  <a href="#section4">Code</a>
</nav>
```

3. De knop om naar de bovenkant van de pagina te gaan bevindt zich boven de eerste sectie.

```
<button onclick="topFunction()" id="myBtn" title="Go to top"></button>
```

4. De pagina bevat een paar animaties in de vorm van .gif waaruit de eerste sectie bestaat.

```





```

5. De tijdlijn is een lijst van elementen met elk een eigen span voor de datum en header met een a-tag waarin de bestanden gelinkt zijn.

```
<section id="section3" class="section">
  <h2>Tijdlijn</h2>
  <div class="timeline">
    <ul>
      <li>
        <span>12-12-2023</span>
        <article class="content">
          <h3><a href="documentatie/ganttchart.pdf" download>Gantt-chart</a></h3>
        </article>
      </li>
    </ul>
  </div>
</section>
```

6. De sectie waarin je de code kan downloaden bestaat hoofdzakelijk uit een a-tag waarin het zip bestand van de code is gelinkt.

```
<section id="section4" class="section">
  <h2>Code</h2>
  <a href="gezamenlijkeCode.zip" download title="download group code">
    
  </a>
</section>
```

7. Deze functie opereert om de knop die de gebruiker naar de bovenkant van de pagina stuurt te laten werken. De knop verschijnt pas nadat de gebruiker meer dan 1000 pixels naar beneden heeft gescrold. Als dit nog niet het geval is dan wordt de knop nog niet weergegeven.

```
window.onload = function() :void {
  var mybutton :HTMLInputElement = document.getElementById( elementId: "myBtn");

  window.onscroll = function () :void {
    if (document.body.scrollTop > 1000 || document.documentElement.scrollTop > 1000) {
      mybutton.style.display = "block";
    } else {
      mybutton.style.display = "none";
    }
  };
};
```

8. Deze functie vergroot een element ten opzichte van de andere elementen uit de tijdlijn wanneer de gebruiker met zijn muis hier over heen beweegt en zorgt dat wanneer de gebruiker

dit niet doet de letters van het element even groot zijn als de andere elementen uit de tijdlijn.

```
window.addEventListener( type: 'DOMContentLoaded', listener: (event : Event) : void => {
  const timelineItems : NodeListOf<Element> = document.querySelectorAll( selectors: '.timeline ul li');

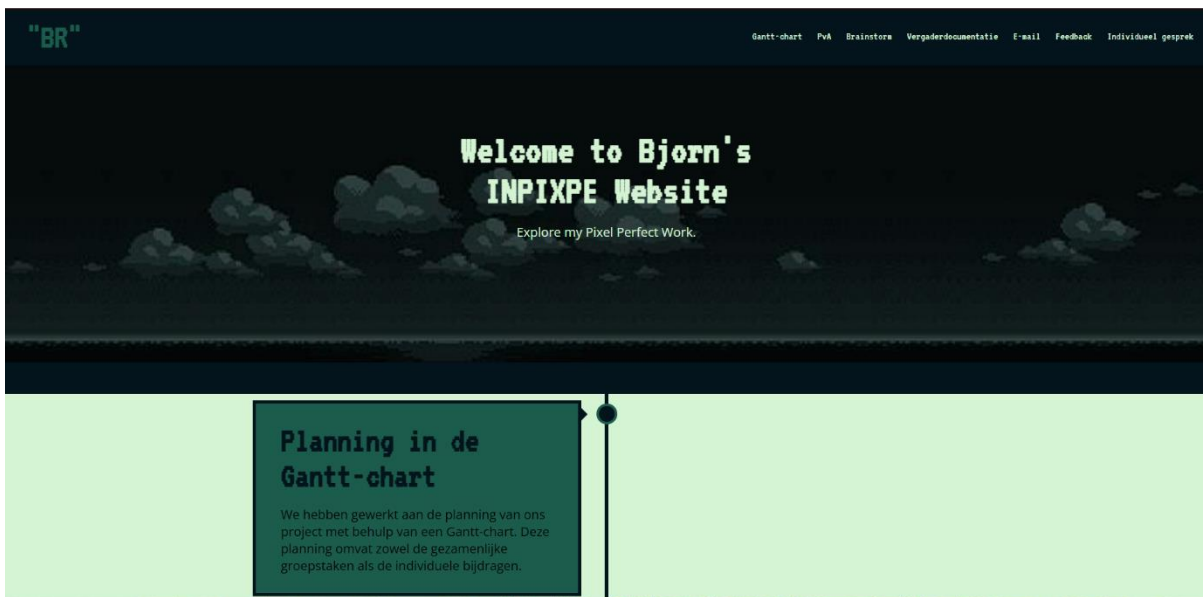
  timelineItems.forEach( callbackfn: item : Element => {
    const content : Element = item.querySelector( selectors: '.content');

    item.addEventListener( type: 'mouseover', listener: () : void => {
      content.style.transform = 'scale(1.04)';
    });

    item.addEventListener( type: 'mouseout', listener: () : void => {
      content.style.transform = 'scale(1)';
    });
  });
});
```



## Handleiding Website Bjorn (Gebruiker):

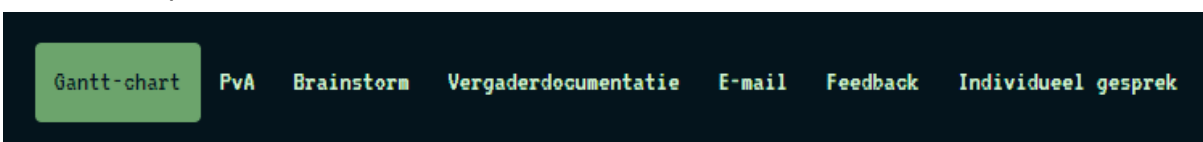


Bij het starten van deze website krijgt u dit voor u te zien. Dit is de hoofdpagina van de gehele website.



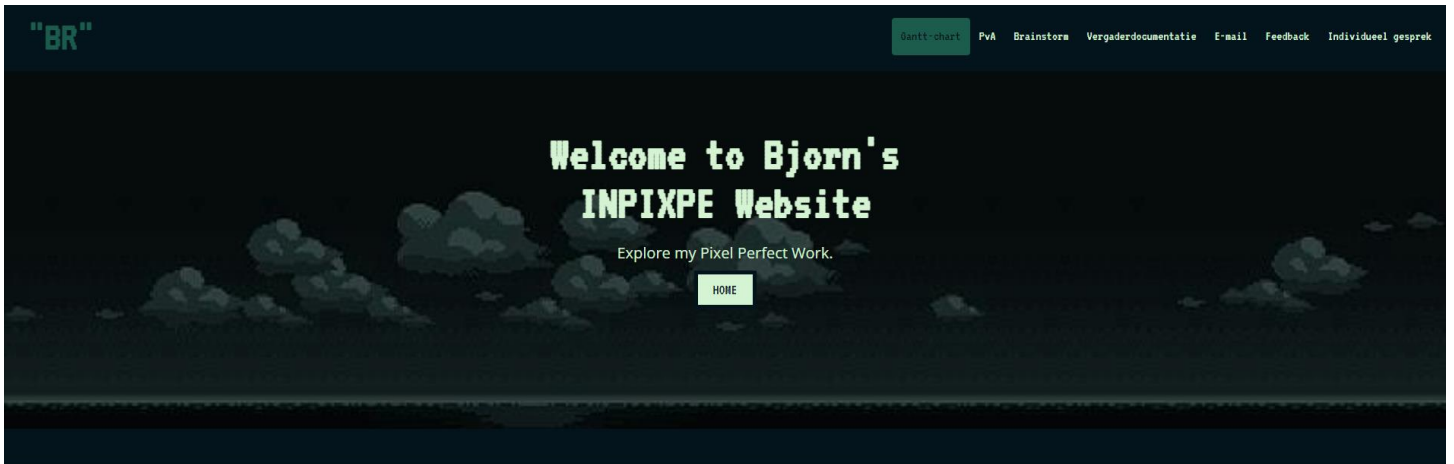
naar beneden scrolt zit u de tijdlijn met alle opgeleverde producten. Als u uw muis daar overheen "hovered" krijg u een preview van het bestand te zien. En met een klik wordt u door ver wezen naar de download pagina van dit product.

Een andere manier om bij de download pagina's te komen is met de navigatie bar aan de bovenkant van de pagina. Hier meer kan u in een keer kiezen naar welke pagina u wil zonder de preview.



Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)

Als het eenmaal gelukt is om bij de download pagina te komen ziet u dit scherm voor u:



## Planning in de Gantt-chart

Het product wat je maakt ontstaat natuurlijk niet plotseling. Voor alle producten met enige complexiteit moet je nadenken over deelproducten. Heb je deze scherp in beeld? Dan kan je bedenken welke activiteiten je moet ondernemen om die deelproducten te maken. Deze activiteiten heb je nodig om in je planning te verwerken. Hieronder zie je de stappen die je moet nemen om tot een goede planning te komen. Denk goed na over de invulling van elke stap om echt een goede planning neer te zetten.

Hier kan u uitleg vinden over de opdracht en een download link aan het einde van de pagina:

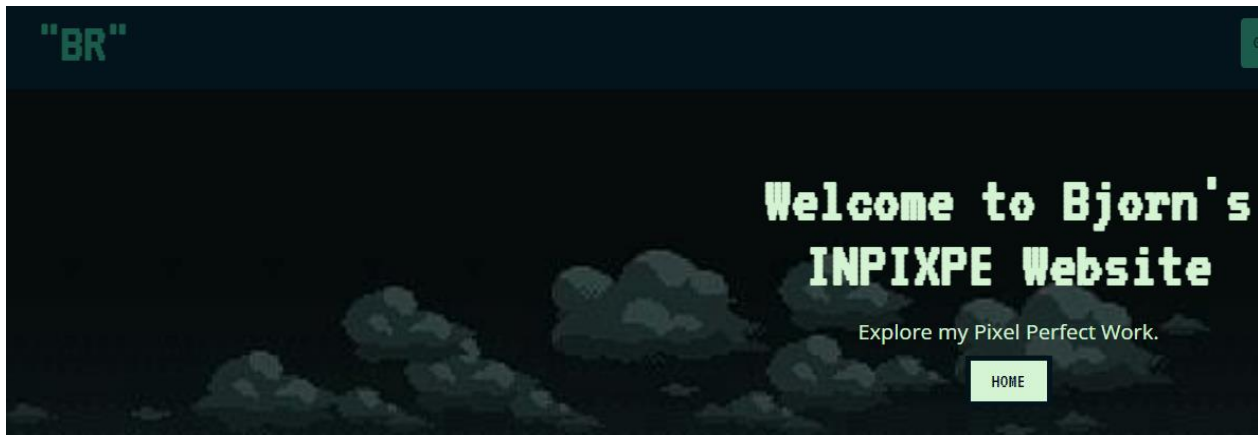
## Planning in de Gantt-chart

Het product wat je maakt ontstaat natuurlijk niet plotseling. Voor alle producten met enige complexiteit moet je nadenken over deelproducten. Heb je deze scherp in beeld? Dan kan je bedenken welke activiteiten je moet ondernemen om die deelproducten te maken. Deze activiteiten heb je nodig om in je planning te verwerken. Hieronder zie je de stappen die je moet nemen om tot een goede planning te komen. Denk goed na over de invulling van elke stap om echt een goede planning neer te zetten.

De Gantt-chart bevat zowel de planning voor het groepswerk als voor de individuele activiteiten. Je maakt binnen de planning dus inzichtelijk hoe jullie als groep samenwerken, maar ook hoe elk individuele lid de eigen activiteiten organiseert. De planning voor het groepswerk is immers afhankelijk van de individuele activiteiten en vice versa. Hetzelfde geldt voor de producten: het project is opdeelbaar in gezamenlijke (**totaal**) producten en (**individuele**) deelproducten.

### Download(s):

- [Download Gantt-chart](#)



Als u een andere wil zien pagina wil kan u de navigatie bar weer gebruiken aan de boven kan van de pagina. Als u terug wil naar het hoofdmenu kan u gebruik maken van de Home knop aan de boven kan van de pagina of u kan op het "BR" Logo klikken.

#### Handleiding Website Bjorn (Beheerder):

De website is opgebouwd uit HTML sections met daarin de leerdoelen, ontwerpkeuzes, de tijdlijn en de code van de websites.

- Deze zijn opgebouwd uit <section>-elementen in de HTML, zoals "header-image," "image-bar," en "timeline," waarin informatie over leerdoelen, ontwerpkeuzes en projecttijdlijn worden weergegeven. Hier kan ook overheen gehoverd worden voor een pdf-preview.

De navigatiebalk werkt door de a-tags direct te linken naar de desbetreffende pagina's van de website.

- De navigatiebalk bevat verschillende <a>-elementen met href-attributen die linken naar specifieke pagina's van de website linken, zoals gantt.html, brainstorm.html, enz.

De tijdlijn is een lijst van verschillende "article" elementen binnen de "timeline" sectie, elk met een "data-file-url" en "data-website-url" attribuut voor het linken van bestanden en websites.

De sectie waarin je de code kan downloaden staan op andere pagina's voor elk onderdeel speciek. Daarin staan download link voor de desbetreffende onderdelen.

Voor alle download pagina's is er een home knop te vinden onder de header image tekst om de gebruiker makkelijk terug te laten gaan naar de homepage. Ook is er rond dezelfde plek een navigatie bar die de gebruiker snel door stuurt naar de andere download pagina's

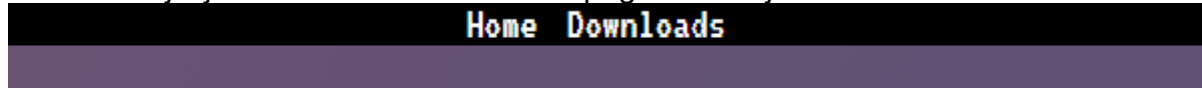
De pdf-preview laat de eerste pagina van de opgeleverde pdf. Hierbij kan de gebruiker kan de gebruiker ook meteen de tekst beter lezen, omdat automatische de beschrijving wordt gehighlight.

# Gebbruikershandleiding (Individuele Website)

## Kevin van der Plas:

### Navigatie:

- Op de bovenkant van de website is er een navigatiebalk, “home” stuurt je naar de tijdljn en “downloads” naar de pagina waar je alle PDFs kan downloaden.



### Bestanden bekijken:

- De bestanden zijn onzichtbaar gemaakt totdat jouw muis er over heen staat, dan kan je het bekijken.
- Klik op de tekst onder het om de volledige PDF te bekijken.



### Tijdljn interactie:

- De pagina is gestructureerd in rijen, elke rij heeft 3 secties. De middelste is altijd tekst, en er is een foto links of rechts (wordt per rij gespiegeld).
- Door op de onzichtbare foto te hoveren met jouw muis wordt het zichtbaar (de positie wordt getoond met een pijl naast de titel).
- Klik op de tekst onder het om de volledige PDF te bekijken.



### Tips:

- Maak gebruik van de navigatiebalk bovenin om te wisselen van pagina.

- Onthoud, als de foto op de rij boven jou links stond, dat die op de volgende rij rechts staat.
- Kijk naar de progressie balk onderop jouw scherm hoever je op de website bent.

## Beheerdershandleiding (Individuele Website) Kevin van der Plas:

### Navigatie:

- Op de bovenkant van de website is er een navigatiebalk, dat valt in de code onder "nav". Als je een extra pagina wilt toevoegen moet je een li toevoegen, en als je er een wilt verwijderen verwijder je de li.

```
<header>
  <nav>
    <ul class="nav-list">
      <li class="nav-item"><a href="#">Home</a></li>
      <li class="nav-item"><a href="downloads.html">Downloads</a></li>
    </ul>
  </nav>
  <div class="progress-container">
    <div class="progress-bar" id="myBar"></div>
  </div>
</header>
```

### Bestanden bekijken:

- De bestanden zijn onzichtbaar gemaakt totdat jouw muis er over heen staat, dan kan je het bekijken. Dit wordt gedaan door initieel de foto onzichtbaar te

Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)

maken, en dan met JavaScript wordt er een CSS class toegevoegd wat de foto zichtbaar maakt.

```
1+ usages
function showImage(element) : void {
  const img = element.querySelector('img');
  const paragraph = element.querySelector('p');

  if (img) {
    img.classList.add('visible');
  }

  if (paragraph) {
    paragraph.style.display = 'block';
  }
}

1+ usages
function hideImage(element) : void {
  const img = element.querySelector('img');
  const paragraph = element.querySelector('p');

  if (img) {
    img.classList.remove('visible');
  }

  if (paragraph) {
    paragraph.style.display = 'none';
  }
}

.img-left-desktop img,
.img-right-desktop img,
.img-left-desktop p,
.img-right-desktop p {
  display: block;
  margin: 0 auto;
  opacity: 0;
  transition: opacity 0.3s ease;
}

.img-left-desktop:hover img,
.img-right-desktop:hover img,
.img-left-desktop:hover p,
.img-right-desktop:hover p {
  opacity: 1;
}
```

- Klik op de tekst onder het om de volledige PDF te bekijken. Om aan te passen welk PDF wordt geopend moet je in de code de bestandsnaam aanpassen.

```
<article class="img-right-desktop" onmouseover="showImage(this)" onmouseout="hideImage(this)">
  <div class="image-text-container">
    <figure></figure>
    <a href="pdf/ontwerpwebsite.pdf" target="_blank"><p>Klik hier om de pdf te bekijken!</p></a>
  </div>
</article>
<article class="text-container">
  <h2>< Ontwerp van de website </h2>
  <p>In dit document worden de styleguides getoond van de groepswebsite en de individuele website.</p>
</article>
```

### Tijdelijk interactie:

- De pagina is gestructureerd in rijen, elke rij heeft 3 secties. De middelste is altijd tekst, en er is een foto links of rechts (wordt per rij gespiegeld). Als je een rij wilt toevoegen met content is het de bedoeling dat je de template volgt (deze code), en de teksten aanpast.

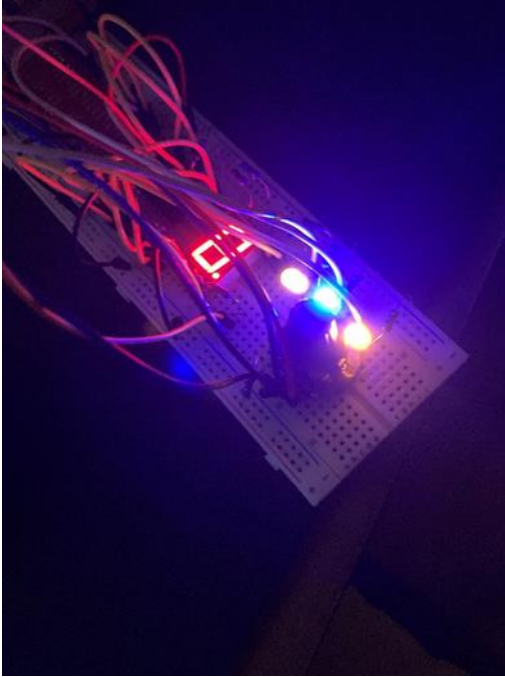
```
<section class="container-left-desktop">
  <section class="row-1-desktop">
    <article class="img-right-desktop" onmouseover="showImage(this)" onmouseout="hideImage(this)">
      <div class="image-text-container">
        <figure></figure>
        <a href="pdf/ontwerpwebsite.pdf" target="_blank"><p>Klik hier om de pdf te bekijken!</p></a>
      </div>
    </article>
    <article class="text-container">
      <h2>Ontwerp van de website 🍷</h2>
      <p>In dit document worden de styleguides getoont van de groepswebsite en de individuele website.</p>
    </article>
  </section>
</section>
```

## Gebruikershandleiding (Puzzel) Kevin van der Plas:

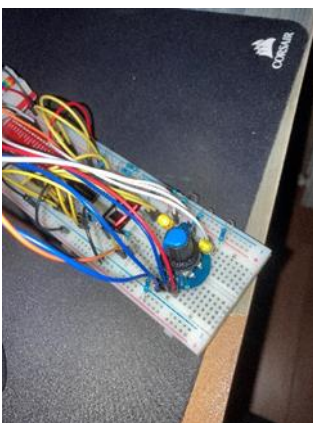
Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)

De puzzel heet “Het Grote Gokspel”. Bij deze uitdaging heb je de beschikking over 3 LEDs, en het doel is om 5 keer op rij de juiste LED te selecteren! Als het je lukt om gedurende 5 opeenvolgende pogingen de correcte LED te kiezen, win je het spel en wordt de code onthuld.

1. Als je het spel opstart dan gaan alle LEDs aan en de display, zo weet je dat die werkt.

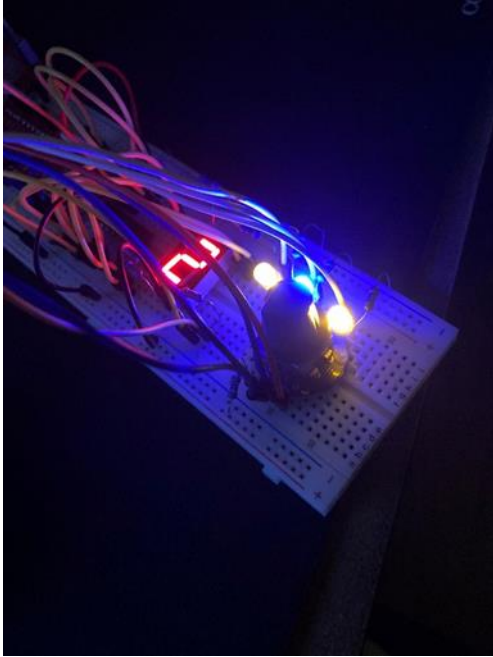


2. Met de controller kan je een LED selecteren (degene die je hebt geselecteerd staat aan), door de controller in te drukken bevestig je je selectie. Daarna kijkt het spel of je wel of niet de juiste led hebt gekozen. Je kan zien welk level je bent op het scherm links van de LEDs en controller.

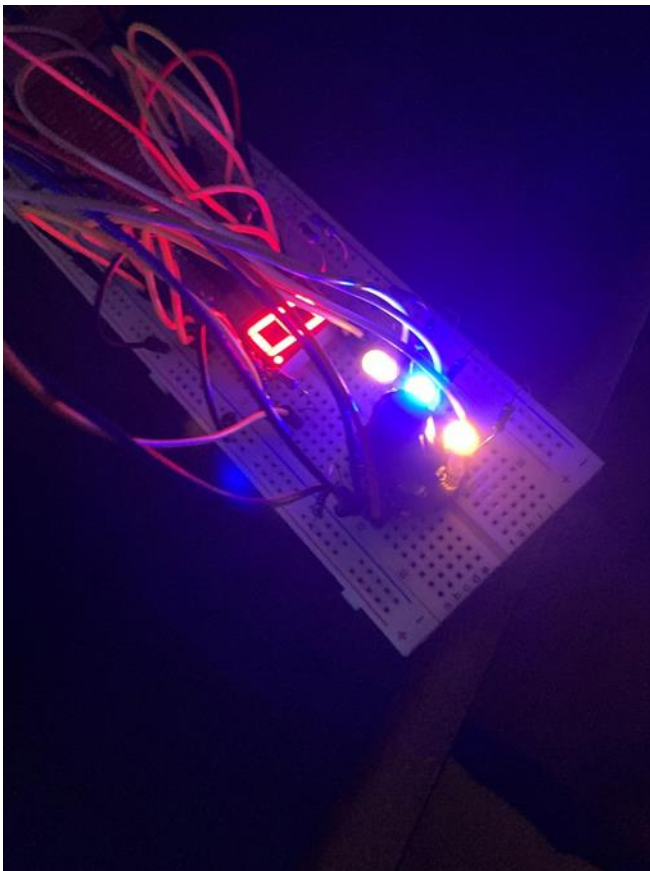


3. Nadat je de juiste LED selecteert gaan alle lampjes aan en schiet het level met 1 omhoog op de display.





4. Nadat je 5x achter elkaar de juiste LED hebt gekozen, moet je 5 seconde in spanning afwachten totdat de LEDs aan en uit gaan knipperen meerdere keren. Daarna krijg je de code te zien op de display!



# Beheerdershandleiding (Puzzel) Kevin van der Plas:

1. Sluit de Raspberry PI op aan het stroom, hierdoor gaat de puzzel na enkele seconde starten.
2. Deze code geeft de binaire combinatie om cijfers/letters te tonen met de display.

```
segments = {  
  0: [1, 1, 1, 1, 1, 1, 0],  
  1: [0, 0, 1, 1, 0, 0, 0],  
  2: [1, 1, 0, 1, 1, 0, 1],  
  3: [0, 1, 1, 1, 1, 0, 1],  
  4: [0, 0, 1, 1, 0, 1, 1],  
  5: [0, 1, 1, 0, 1, 1, 1],  
  8: [1, 1, 1, 1, 1, 1, 1],  
  "C": [1, 1, 0, 0, 1, 1, 0],  
  "0": [1, 1, 1, 1, 1, 1, 0],  
  "D": [1, 1, 1, 1, 0, 0, 1],  
  "E": [1, 1, 0, 0, 1, 1, 1]  
}
```

3. Deze code laat het spel opstarten.

```

def main():
    try:
        GPIO.setmode(GPIO.BCM)
        set_up_gpio()

        led_orders = initialize_led_orders()

        for led_order in led_orders:
            move_stick(led_order)

    finally:
        GPIO.cleanup()
        print("GPIO cleanup voldaan.")

usage
def set_up_gpio():
    GPIO.setup([LED_PIN_1, LED_PIN_2, LED_PIN_3], GPIO.OUT)
    GPIO.setup([S1_PIN, S2_PIN, KEY_PIN], GPIO.IN, pull_up_down=GPIO.PUD_UP)
    GPIO.setup([DATA_PIN, LATCH_PIN, CLOCK_PIN], GPIO.OUT)

```

4. Deze functie zorgt er voor dat de LEDs worden geshuffled voor elk level. Verander de range als je het spel meer levels wilt geven.

```

usage
def initialize_led_orders():
    led_orders = []
    for _ in range(5):
        led_order = [LED_PIN_1, LED_PIN_2, LED_PIN_3]
        random.shuffle(led_order)
        led_orders.append(led_order)

        GPIO.output(led_order, GPIO.HIGH)
        time.sleep(1)
        GPIO.output(led_order, GPIO.LOW)

    return led_orders

```

5. Deze functie zorgt er voor dat je je rotary stick kan bewegen, het controleert ook welke LED je selecteert en of het de correcte is. En stuurt je door naar de volgende level als je de goede kiest, of stuurt je terug naar level 1 als je de foute kiest.

```

1 usage
def move_stick(led_order):
    global huidige_led, current_level

    last_s1_rotary = GPIO.input(S1_PIN)
    last_button_time = time.time()

    while True:
        huidige_s1_rotary = GPIO.input(S1_PIN)
        huidige_button_rotary = GPIO.input(KEY_PIN)

        if last_s1_rotary != huidige_s1_rotary:
            if GPIO.input(S2_PIN) != huidige_s1_rotary:
                huidige_led = (huidige_led + 1) % 3
            else:
                huidige_led = (huidige_led - 1) % 3

            GPIO.output(led_order, GPIO.LOW)
            GPIO.output(led_order[huidige_led], GPIO.HIGH)

        if huidige_button_rotary == GPIO.LOW:
            current_time = time.time()
            if current_time - last_button_time > 0.5:
                last_button_time = current_time

            print(f"Button Pressed. Selected LED: {huidige_led + 1}")

            if huidige_led == get_correct_led():
                current_level += 1
                print(f"Level Up! Current Level: {current_level}")
                display_digit(current_level)

            huidige_led = 0

```

Verander "current\_level == 5" naar dezelfde getal als range() als je je spel meer levels wilt geven.

```

        GPIO.output(led_order, GPIO.HIGH)

        if current_level == 5:
            print(f"Gefeliciteerd! Je hebt gewonnen!")
            time.sleep(5)

            blink_leds(led_order)

            current_level = 8
            display_digit(current_level)
            break

        else:
            current_level = 1
            print(f"You Failed! Back to Level 1")

    last_s1_rotary = huidige_s1_rotary

    display_digit(current_level)
    time.sleep(0.1)

```

Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)

6. Deze functies vertalen de bits die via de shift register wordt doorgestuurd om naar de display zodat die de correcte dingen kan tonen.

```
def display_digit(digit):
    if digit in segments:
        segment_data = segments[digit]
        shift_out(DATA_PIN, CLOCK_PIN, LATCH_PIN, sum(bit << i for i, bit in enumerate(segment_data)))
    elif str(digit) in segments:
        segment_data = segments[str(digit)]
        shift_out(DATA_PIN, CLOCK_PIN, LATCH_PIN, sum(bit << i for i, bit in enumerate(segment_data)))
    else:
        print("Invalid digit")

def display_bits(bits):
    for bit in bits:
        GPIO.output(DATA_PIN, bit)
        GPIO.output(CLOCK_PIN, GPIO.HIGH)
        GPIO.output(CLOCK_PIN, GPIO.LOW)

    GPIO.output(LATCH_PIN, GPIO.HIGH)
    GPIO.output(LATCH_PIN, GPIO.LOW)

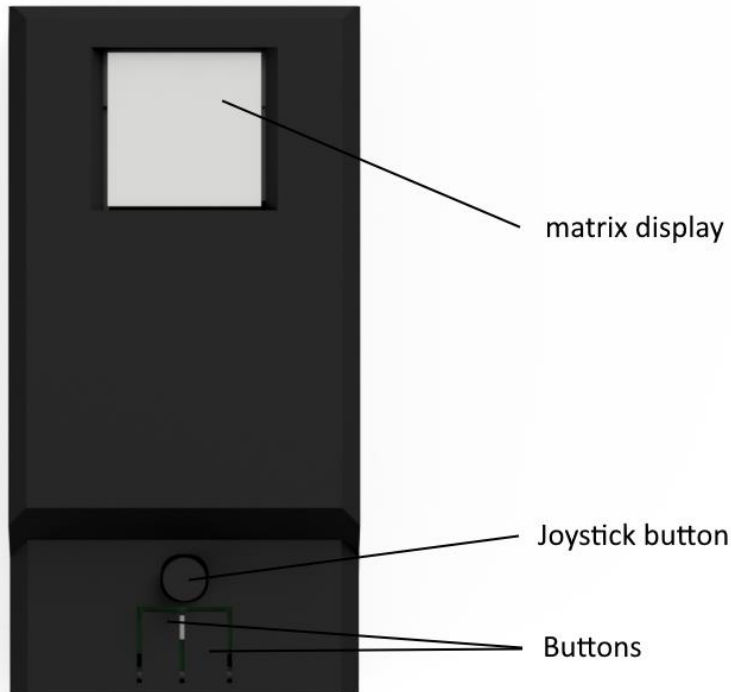
2 usages
def shift_out(data_pin, clock_pin, latch_pin, data):
    GPIO.output(latch_pin, GPIO.LOW)
    for bit in range(NUM_SEGMENTS):
        GPIO.output(clock_pin, GPIO.LOW)
        GPIO.output(data_pin, (data >> bit) & 1)
        GPIO.output(clock_pin, GPIO.HIGH)
    GPIO.output(latch_pin, GPIO.HIGH)

1 usage
def get_correct_led():
    return current_level % 3
```

## Handleiding hardware Jordy

### *De bediening*

Het spel heeft 3 knoppen waarvan twee drukknoppen en één joystick knop. Verder beschikt het spel over een 8x8 matrix display waar het spel te zien is. Met de joystick kan je de cursor links- en rechts en omhoog en omlaag verplaatsen. Door op de joystick te drukken kan je een plaats in nemen op het 3x3 grid.



### *Het spel*

Het spel wat erop gespeeld kan worden is boter, kaas en eieren. Ik heb het spel een nieuwe toepasselijke naam gegeven, namelijk boter, kaas en Raspberry.

Het spel speel je tegen een AI. Jij mag als eerste een positie innemen op het grid. Deze plek zal gevisualiseerd worden met een vierkantje (2x2 pixels). Daarna zal de AI enkele seconden later ook zijn eerste zet innemen die wordt gevisualiseerd met een lijntje. Het spel blijft doorgaan totdat er ergens op het grid een rij van 3 vierkantjes of 3 lijntjes ontstaat. Als de lijn bestaat uit vierkantjes betekent het dat jij dus hebt gewonnen en dan zul je de code te zien krijgen voor de website. Heb je helaas 3 lijntjes in beeld dan krijg je een 'You lost :( ' berichtje in het scherm en zal je ook niet de code ontvangen.

### Stappen

1. Sluit het apparaat aan op een stroombron (netstroom of externe batterij) met een usb-c kabel. Zorg dat de kabel er goed in zit zodat de kabel niet plotseling los kan raken.
2. Wacht totdat er een roodvierkantje op het scherm verschijnt van 4x4 pixels. Let op: het kan even duren voordat het apparaat is opgestart.
3. Beweeg links en rechts of omhoog en omlaag door de joystick in een richting te duwen.
4. Speel het spel.
  - 4.1 Druk op de joystick om een positie in te nemen.
  - 4.2 Wacht tot de computer een beslissing neemt.
  - 4.3 Blijf doorgaan tot u wint of verliest.
5. Bent u klaar met het spel? Sluit het spel af door op de linker knop te drukken en wacht 10 seconden.
6. U kunt nu de kabel uit het apparaat halen.

### Website jordy

Als je de website opent kun je naar beneden scrollen om de tijdlijn te vinden of je kan op het hamburger menu klikken om automatisch naar de tijdlijn te scrollen.



Hover vervolgens met je muis over de tijdlijn items om een scrollbar zichtbaar te maken. Als je met de muis over de linkjes zweeft krijg je een berichtje te zien dat je de link kunt klikken om hem te openen.



Om een overzicht te krijgen van alle documenten die beschikbaar zijn kan je naar het hamburger menu gaan en op downloads klikken. Vervolgens krijg je een lijst met alle documenten die je kan downloaden.

#### Beheerders handleiding hardware

Alle knoppen van het spel zitten op de volgende pinnen aangesloten

- Lft = 16
- Rht = 20
- Up = 12
- Dwn = 21
- Set = 13
- Mid = 6
- COM = 3.3V

Al deze input zitten eerst nog aan een zelfgemaakt shield vast voordat ze naar de RaspberryPi gaan. Dat shield is bedoeld voor de pull down schakeling. Op dat shield zit een pin. Namelijk een GND-pin zodat alle input standaard als laag worden uitgelezen

Vervolgens zijn er nog een aantal pinnen die voor het display zijn gebruikt dat zijn de volgende pinnen:

- Data = mosi
- Spisclk = clk
- Spice0 = cs
- VCC = 5V
- GND= GND

In de code van het spel is gebruik gemaakt van de LUMA library voor de led matrix. Deze library stuurt de led matrix aan. Verder is er ook gebruik gemaakt van de volgende library's:

- Time – voor tijd gerelateerde handelingen
- GPIO – voor het aansturen van de RaspberryPi pinnen
- Random – voor het random plaatsen van lijnen op het matrix display als de computer een keuze moet maken
- Os – voor het afsluiten van de RaspberryPi na het drukken op de set knop

Het spel zit in een while-loop waarbij de matrix alleen wordt geüpdatet als een speler een zet heeft gemaakt of als er een positie is ingenomen. In een for-loop wordt de matrix uitgelezen en geprojecteerd op het display.

Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)



```
grid = [
    ['- ', '- ', '- '],
    ['- ', '- ', '- '],
    ['- ', '- ', '- ']]
```

Omdat het matrix display echt 8x8 en het grid een 3x3 grid is wordt steeds +3 opgeteld aan het einde van de for-loop.

```
while True:
    if entered:
        with canvas(device) as draw:
            print(valux)
            draw.rectangle((int(valux), int(valuy), int(valux)+3, int(valuy)+3), outline='white')

            gridy1 = 0
            gridy2 = 1

            for idxy, ycor in enumerate(grid):
                gridx1 = 0
                gridx2 = 1
                for idxx, xcor in enumerate(grid[idxy]):
                    if grid[idxy][idxx] == 'x':
                        draw.rectangle((int(gridx1), int(gridy1), int(gridx2), int(gridy2)), fill='white')
                        # draw.point((int(gridx2), int(gridy2)), fill='white')
                        # print(f'{gridx1} {gridy1} - {gridx2} {gridy2}')
                    elif grid[idxy][idxx] == 'o':
                        draw.line((int(gridx1), int(gridy1), int(gridx2), int(gridy2)), fill='white')
                        # draw.point((int(gridx2), int(gridy2)), fill='white')
                        # print(f'{gridx1} {gridy1} - {gridx2} {gridy2}')

                gridx1 += 3
                gridx2 += 3

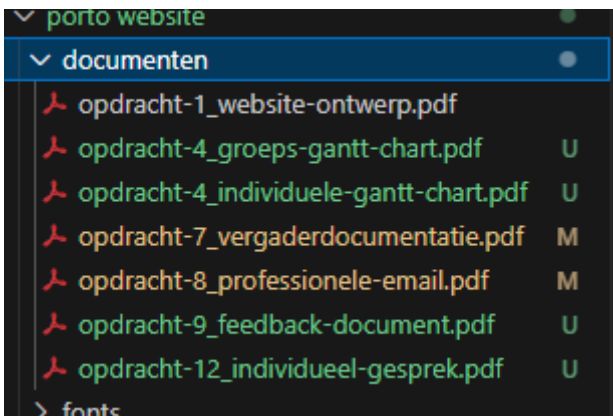
            gridy1 += 3
            gridy2 += 3
```

Zodra de speler klaar is met spelen en op de set knop drukt wordt het spel afgesloten.

```
elif GPIO.input(rst_input) == 1:
    os.system('sudo shutdown now -h')
    exit(0)
elif GPIO.input(mid_input) == 1:
```

Beheerders handleiding website jordy

Alle documenten die op de website te vinden zijn zitten in een documenten folder. Als je extra documenten wil toevoegen is het verstandig om die in deze folder toe te voegen.



Jelle Mossel (1109429), Jordy de Vries (1151166), Kevin van der Plas (1145852), Emma Nauta (1151416) & Bjorn Ravensbergen (1150431)

Om ze vervolgens ook bij de downloads te kunnen zetten moet er een extra li tag toegevoegd worden in downloads.html. Die kunnen dan hierbij worden toegevoegd in de unordered list in de section met de class downloads-frame.

```
54 <ul>
55 <li>
56 <a class="download-item" href="styleguide.png"
57 download="styleguide">styleguide</a>
58 </li>
59 <li>
60 <a class="download-item" href="documenten/opdracht-1_website-ontwerp.pdf"
61 download="website-ontwerp-24">Website ontwerp</a>
62 </li>
63 <li>
64 <a class="download-item" href="documenten/opdracht-4_groeps-gantt-chart.pdf"
65 download="groepsplanning-24">Groepsplanning</a>
66 </li>
67 <li>
68 <a class="download-item" href="documenten/opdracht-4_individuele-gantt-chart.pdf"
69 download="individuele-planning-24">Individuele planning</a>
70 </li>
71 <li>
72 <a class="download-item" href="documenten/opdracht-7_vergaderdocumentatie.pdf"
73 download="vergaderdocumenten-24">Download de vergaderdocumenten</a>
74 </li>
75 <li>
76 <a class="download-item" href="documenten/opdracht-8_professionele-email.pdf"
77 download="professionele-email">Professionele e-mail</a>
78 </li>
79 <li>
80 <a class="download-item" href="documenten/opdracht-9_feedback-document.pdf"
81 download="feedbackdocumenten-24">Feedback documenten</a>
82 </li>
83 <li>
84 <a class="download-item" href="documenten/opdracht-12_individueel-gesprek.pdf"
85 download="individueel-gesprek-24">Individueel gesprek</a>
86 </li>
87 <li>
88 <a class="download-item" href="nog-niet-beschikbaar"
89 download="nog-niet-beschikbaar">Brainstorm documenten en gemaakte keuzes</a>
```

De iconen op de website komen van de font awesome library. Als er meer iconen nodig zijn is het bijzonder makkelijk om met font awesome iconen toe te voegen. Iconen kunnen worden toegevoegd door een `<i>` tag toe te voegen met een specifieke class. Die class kan je op de website van font awesome vinden waar je overigens ook wat aanpassingen kan maken aan de iconen. Zo kun je de iconen bijvoorbeeld als thin line of volledige ingevuld hebben. De `<i>` tags kunnen makkelijk in een `<h>` tag worden toegevoegd of in een `<p>` tag.

```
<h3>Onderbouwing ontwerp <i class="fa-solid fa-comments fa-2xs" style="color: #4c4c4c;"></i></h3>
<h4>Thema</h4>
```

De tijdlijn heeft een scrollbar die standaard niet te zien is. Als je met de muis over de tijdlijn zweeft dan komt de horizontale scrollbar tevoorschijn. Op telefoon en andere apparaten die onder de 992px komen werkt dit net even anders. Door een media screen aan te maken is de scrollbar daar permanent zichtbaar. Echter zie je deze niet op een telefoon, dat heeft er mee te maken dat mobile browser websites net even anders behandelen dan desktopbrowsers dat doen

```
#timelinescroll {
  margin-top: 2em;
  overflow-x: scroll;
  white-space: nowrap;
  visibility: hidden;
  display: flex;
  flex-direction: row;
}

.timelineitem,
#timelinescroll:hover,
#timelinescroll:focus {
  visibility: visible;
}
```

## Literatuur

1: W3Schools, meerdere malen bezocht vanaf 11-01-2024 t/m 25-01-2024

<https://www.w3schools.com/>

2: Website over scope, bezocht op 12/18/2023:

<https://www.projectsucces.nl/scope/>