Szymon (S.M.) Kraczek

VISTA College, 2024

Documentatie Demonstrator

19-2-2024

Voorwoord

Voor u ligt de documentatie voor de Sphero demonstrator, in dit document worden de doelen en stappen besproken die gebruikt zijn voor het ontwikkelen van de demonstrator. Deze opdracht zal worden uitgevoerd binnen 5 lesweken in het DI-Lab in Brightlands, Heerlen.

Inhoud

[1 Aanleiding 2](#_Toc159245375)

[1.1 Doel 2](#_Toc159245376)

[2 Rolverdeling 2](#_Toc159245377)

[2.1 Werkmethode 2](#_Toc159245378)

[3 Sphero BOLTS 3](#_Toc159245379)

[3.1 Wat zijn Sphero BOLTS? 3](#_Toc159245380)

[3.2 Hoe Werken de BOLTS? 3](#_Toc159245381)

[3.3 Hoe gaan wij ze besturen? 3](#_Toc159245382)

[3.4 Functionele eisen 3](#_Toc159245383)

[Must Haves 3](#_Toc159245384)

[Could Haves 3](#_Toc159245385)

[Won’t haves 3](#_Toc159245386)

[4 Uitvoering 4](#_Toc159245387)

[4.1 Verbinding 4](#_Toc159245388)

[4.2 Besturing 4](#_Toc159245389)

[4.3 Camera 4](#_Toc159245390)

[4.4 Vormen 5](#_Toc159245391)

[4.6 Documentatie 5](#_Toc159245392)

[4.6.1 Trello 5](#_Toc159245393)

[4.6.2 GitHub 5](#_Toc159245394)

[4.6.3 Opnames 6](#_Toc159245395)

[5 Referenties 6](#_Toc159245396)

[BOLT 6](#_Toc159245397)

[Bleak 6](#_Toc159245398)

[GATT 6](#_Toc159245399)

[PyGame 6](#_Toc159245400)

[OpenCV 6](#_Toc159245401)

# 1 Aanleiding

Dit project is aangegeven en begeleid door Marcel Schmidz en Finn Alberts. Marcel en Finn hebben een demonstrator met Sphero’s, maar deze heeft veel stappen en handelingen die het proces van de demonstrator opbouwen complex en vermoeiend kunnen maken.

## 1.1 Doel

Het doel is om het op- en afbouwen van de demonstrator makkelijker te maken. Dit zullen wij doen door middel van de software van de demonstrator te verbeteren. Voor dit project krijgen wij 5 weken om deze te voltooien met voldoende documentatie en bijhorende eindpresentatie.

# 2 Rolverdeling

Het project bestaat uit een groep van 3 studenten die aangewezen zijn om deze te voltooien. Hieronder bevindt zich de rolverdeling van deze studenten en wat hun taken zullen zijn tijdens het werken van de opdracht.

|  |  |  |
| --- | --- | --- |
| Rol | Naam student | Taken |
| Teamleider | Szymon Kraczek | Zal afspraken maken met het team en de voortgang van het project bijhouden, hieronder valt: Planning, Documentatie en stand-ups |
| Primaire Developer | Mohamed Hasoun | Zal het meeste werken met de software van het project. Zal zorgen dat de functionaliteiten werken. |
| Secundaire Developer | Daniella Baena | Zal meehelpen met kleinere onderdelen van het project. Bijvoorbeeld: Posters of code-problemen. |

## 2.1 Werkmethode

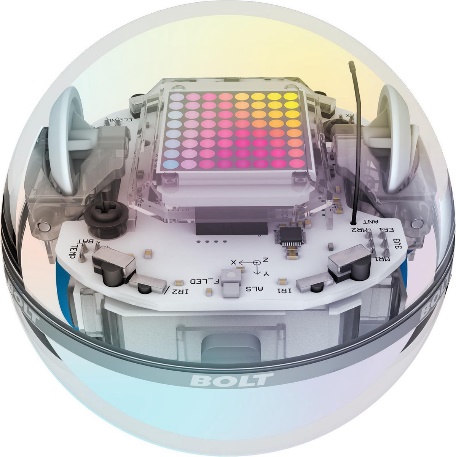
Het project zal worden uitgevoerd met behulp van de SCRUM methode. Hier gaat het om dat we met gebruik van dagelijkse stand-ups binnen in de groep. Hier bespreken we de stand van zaken en hoe ver wij zijn met een opdracht of taak. Verder zullen wij ook bespreken waarin we verder kunnen en wat er gedaan wordt die dag. Deze dagelijkse stand-ups zullen worden opgenomen voor reflectiemogelijkheden.

Verder worden er ook binnen het DI-Lab wekelijkse stand-ups gevoerd van de projecten. Hierbij zullen wij feedback zoeken naar onze progressie en verbetervoorstellen vragen.

# 3 Sphero BOLTS

## 3.1 Wat zijn Sphero BOLTS?

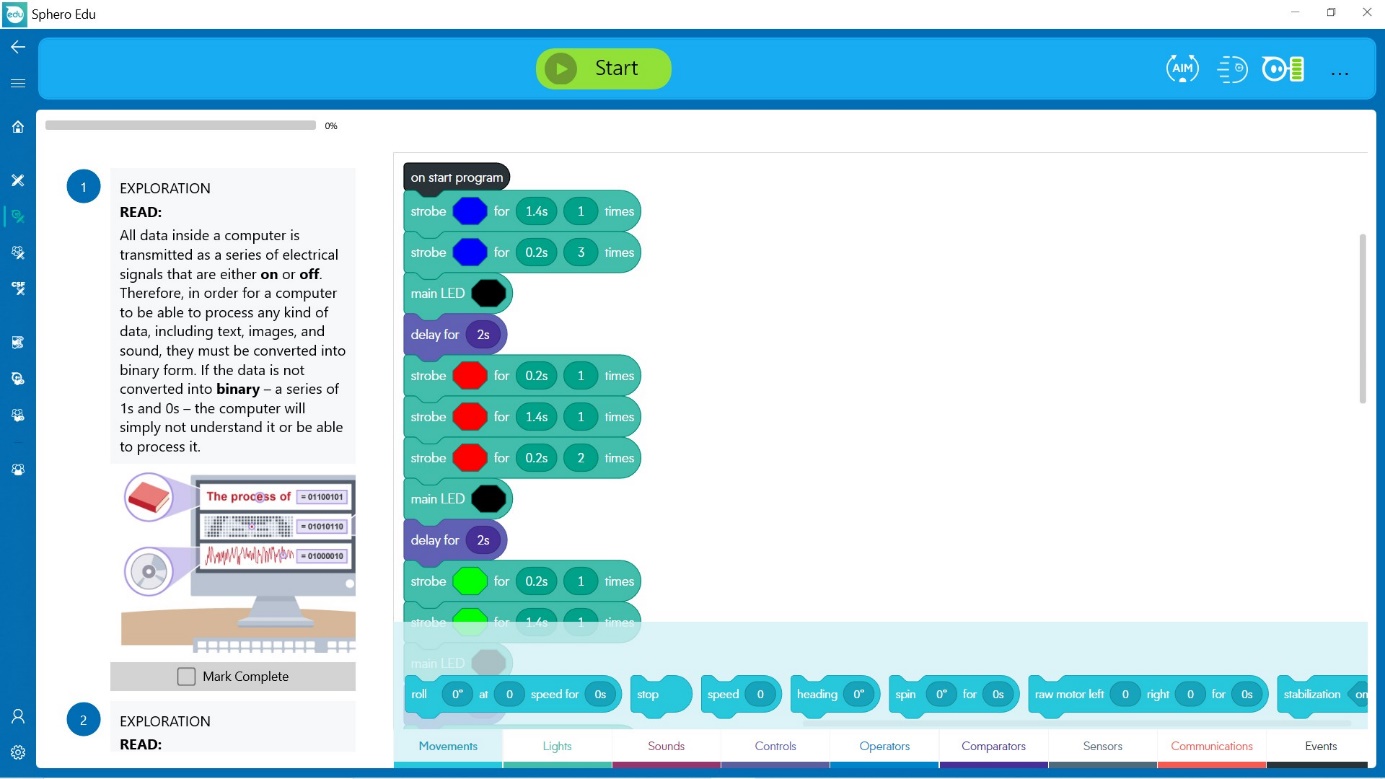
Sphero BOLTS zijn kleine programmeerbare balletjes. Hierin zitten er meerdere componenten zoals motoren, LEDs en speakers. Met het gebruik van de Sphero Edu app, kan je al een kleine voorproefje krijgen van de dingen die je kan uitvoeren met deze robots.



Afbeelding van een Sphero BOLT.

## 3.2 Hoe Werken de BOLTS?

De Sphero’s werken via bluetooth. Heel simpel. Je kan of met je mobiel of laptop verbinden met een Sphero en die besturen via de app. Verder kan je nog handmatig de Sphero programmeren met behulp van code blocks via de Sphero Edu applicatie. Deze app geeft ook mogelijkheden voor werken met Javascript code.



Screenshot van de Sphero Edu app.

## 3.3 Hoe gaan wij ze besturen?

Wij willen de Sphero’s zich automatisch laten aanpassen aan vormen. Deze vormen kunnen hypothetisch van een rechthoek tot een achthoek variëren, maar ons doel gaat simpel blijven. Wij willen met een camera met een “birds-eye view” een beeld krijgen van waar de Sphero’s zich kunnen bewegen. Hiermee zullen wij met de applicatie zelf, vormen invoegen in de camerabeelden en dan zullen de Sphero’s zichzelf aanpassen aan de aangegeven vorm.

## 3.4 Functionele eisen

|  |  |
| --- | --- |
| Must Haves | * Sphero BOLT Connectiviteit * Positionering van BOLTS veranderen via Python * Camerabesturingsmogelijkheden * BOLTS in vormen kunnen opzetten met gebruik van Python * BOLTS moeten terug kunnen gaan naar originele vorm als het aantal BOLTS verandert |
| Could Haves | * BOLT Besturing via controller met Pygame * Simpele besturing via UI |
| Won’t haves | * Aangepaste/getekende vormen van de gebruiker zelf |

# 4 Uitvoering

## 4.1 Verbinding

Met behulp van de [Bleak](#_Bleak) API met Python hebben wij onze laptops kunnen verbinden met de [BOLTS](#_BOLT). Hiermee kunnen wij verdere toevoegingen maken aan de applicatie waar wij signalen kunnen sturen naar de BOLTS. [Bleak](#_Bleak) is een simpele [GATT](#_GATT) client die de bluetooth capabiliteit van je laptop gebruikt om dichtbij zijnde apparaten hun bluetooth signalen te scannen. Hiermee kunnen wij de MAC-adres van de BOLTS oppakken en vanuit daar kunnen wij verder met Python een verbinding maken met de BOLTS.

## 4.2 Besturing

Nu dat we een verbinding hebben tussen de BOLTS en onze laptop, kunnen wij met simpele code deze BOLTS besturen. De BOLTS hebben al opdrachtregels voor beweging, dus de individuele motors aanspreken hoeft niet. Met gebruik van: **“await roll([snelheid], [heading])”** kunnen wij de BOLTS laten rollen in een specifieke directie en snelheid. Hiermee hebben wij verder gewerkt met [Pygame](#_PyGame) om de input van een Xbox 360 controller op te nemen. Met de input van de controller, kunnen wij deze een waarde geven die de BOLTS kunnen gebruiken en dus met de Xbox controller de BOLT besturen.

## 4.3 Camera

Met een standaard web camera, kunnen wij een video-feed binnen krijgen die wij gaan gebruiken voor de BOLT positionering. Met gebruik van [OpenCV](#_OpenCV), kunnen wij deze video-feed met Python lezen. Tijdens het lezen van de video-feed, kunnen wij voor specifieke informatie zoeken binnen de camerabeelden, zoals: kleurverschil, verlichtingsverschil en beweging. Met deze informatie kunnen wij de positie van de BOLTS bepalen.

Afbeelding met schermopname, Multimediasoftware, tekst, Grafische software

Automatisch gegenereerde beschrijving

Screenshot van het detectie systeem voor de BOLTS.

## 4.4 Vormen

Na het verbinden met de BOLTS is voltooid, kunnen wij met een web-interface de BOLTS vormen aangeven. Na het aanklikken van een van de vormen, zullen de BOLTS die vorm aannemen. Na het voltooien van een vorm, zijn de BOLTS klaar voor een andere vorm.

Afbeelding met tekst, schermopname, Multimediasoftware, software

Automatisch gegenereerde beschrijving

Screenshot van de web-interface.

## 4.6 Documentatie

Inclusief dit document, hebben wij tijdens dit project meerdere methodes toegepast om ons werk bij te houden en ook voort te zetten. Hierbij hebben we deze bezigheden toegepast:

* Daily stand-ups binnen het team
* Dagelijkse Trello bewerkingen
* Wekelijkse stand-ups met andere groepen.
* GitHub repository

Al deze processen zijn gedocumenteerd/opgenomen voor reflectie mogelijkheden en het toepassen naar de opdracht. Hieronder zijn voorbeelden hiervan:

### 4.6.1 Trello

Met gebruik van Trello hebben wij een kanban bord aangemaakt waarmee wij onze processen kunnen bijhouden en aanpassen. Deze is dagelijks aangepast door de teamleider en wordt ook besproken tijdens de dagelijkse stand-ups.

<https://trello.com/b/1ATTnzGC/brightlands-sphero-project>

### 4.6.2 GitHub

Wij hebben onze code en documentatie in deze GitHub geplaats. Hier zullen Marcel en Finn alle bestanden kunnen vinden die bij dit project horen.

<https://github.com/MuhammadHasoun/SpheroBolts.git>

### 4.6.3 Opnames

Hieronder bevindt zich een voorbeeld opname van de dagelijkst stand ups:



# 5 Referenties

## BOLT

BOLT is de naam van de robots waarmee wij zullen gaan programmeren. De BOLT is een van de vele robots die Sphero aanbiedt. De BOLT is een kleine, bal-vormige robot met motors en LEDs binnen in een plastic bal. Deze motors en LEDs kun je besturen via code of de Sphero Edu applicatie op je telefoon of laptop.

## Bleak

Bleak is een Python API waarmee je bluetooth signalen kan scannen. Bleak is een GATT client. Met Bleak, kunnen wij de apparaten om ons heen scannen. Deze apparaten zullen als MAC-Adressen opkomen in cmd(command line). Met de MAC-Adres van een BOLT, kunnen wij deze specifiek opzoeken en daarmee verbinden met Bleak.

## GATT

GATT is een acroniem voor Generic ATTribute Profile en definieert de manier waarop twee Bluetooth Low Energy-apparaten gegevens heen en weer overbrengen met behulp van concepten die Services en Characteristics worden genoemd. Het maakt gebruik van een generiek dataprotocol genaamd Attribute Protocol (ATT), dat wordt gebruikt om services, kenmerken en gerelateerde gegevens op te slaan in een eenvoudige opzoektabel met behulp van 16-bits ID's voor elke invoer in de tabel.

## PyGame

Pygame is een set Python-modules ontworpen voor het ontwikkelen van videogames. Pygame voegt functionaliteit toe bovenop de uitstekende SDL-bibliotheek. Hiermee kunt u volledig functionele games en multimediaprogramma's in de Python-taal maken. Pygame is zeer licht en draait op vrijwel elk platform en besturingssysteem.

## OpenCV

OpenCV staat voor Open Source Computer Vision. Een enorme open-sourcebibliotheek die wordt gebruikt voor computer vision-toepassingen, op gebieden die worden aangedreven door kunstmatige intelligentie of machine learning-algoritmen, en voor het voltooien van taken waarvoor beeldverwerking nodig is.