

Opleidingsonderdeel:
Internet Of Things

EINDWERK

Docent:
Frederick Roegiers

Emiel De Vleeschouwer
Kobe Coene

2 NMD
Academiejaar 2019-20, Semester 2

Discover	3
Briefing	3
Define	4
Analyse	4
Concurrentie	4
Software & hardware	4
Inspiration	5
Ideaboard	5
Develop	6
Code snippets: Discord bot (node.js)	6
Code snippets: Sense hat (Python)	8
Schermafbeeldingen	9
Deliver	10
Handleiding	10
Timesheet	10
Deploy	11
IoT-NMD-gamebot	11
Add bot to your discord server	11
installations	11
.env	11
Run Bot	11
Run Python script	11

Discover

Briefing

Voor ons eindproject van Internet of Things hebben we de opdracht gekregen om een toepassing te maken in verband met de Raspberry Pi. Wij hebben gekozen om een bot te maken die op Discord kan draaien. Discord is een online chatapp zoals Skype of Teamspeak dat werkt met voicecalls, videocalls en textchat. Er wordt vaak gewerkt met servers waardoor veel mensen tegelijk kunnen bellen, chatten, ... Op zo'n server kunnen er bots geïnstalleerd worden. Dit zijn third party programma's die het gebruik van discord, vergemakkelijken, leuker maken, raar maken, noem maar op.

Het doel van onze bot is om gegevens op te vragen van het gameplatform Steam. Daarvan halen we gegevens op zoals nicknames, avatars, games, vriendenlijst, ... en zorgen we dat iedereen die op de discord server zit deze gegevens kan opvragen. Dit kan makkelijk door een command in de chatbox in te typen zoals *!friends kobecoene*. Wij wilden ook dat de bot met een Raspberry Pi kan interageren, meer bepaald de SenseHat. De bedoeling is dat wanneer je iemand zijn opvraagt, de avatar van die persoon wordt opgevraagd en doorgestuurd naar de Firebase databank. Daarna haalt de Pi deze avatar op en toont hij deze op de sensehat in rgb-formaat.

Natuurlijk zijn wij niet te de enigen die een bot maken. Daardoor is er al informatie beschikbaar over hoe we dit best opstellen, en zijn er ook typs die we kunnen gebruiken.

Als doelpubliek hebben we eigenlijk enthousiaste gamers die graag met elkaar gamen en die graag ook met discord werken. Het is ook belangrijk dat we daarom ons ook richten tot die doelgroep. Dit doen we door alles eenvoudig te houden zodat er weinig tijd verloren gaat met het leren van de bot.

Define

Analyse

Concurrentie

Er zijn duizenden andere bots die online staan. De meeste staan op <https://top.gg/>. Dat wil niet zeggen dat alle ideeën al zijn ingenomen. Ieder van deze bots heeft zijn unieke functie, sommige uitgebreider dan anderen. Ons idee heeft zeker potentieel om op die website te staan omdat het een uniek idee is.

Met bots te maken is er ook een mogelijkheid om geld te verdienen. Het is natuurlijk te zien hoe ver je wilt gaan. Maar je kan sommige features betalend maken. Niet iedereen wilt dit natuurlijk dus het is niet echt een bron van inkomst.

Software & hardware

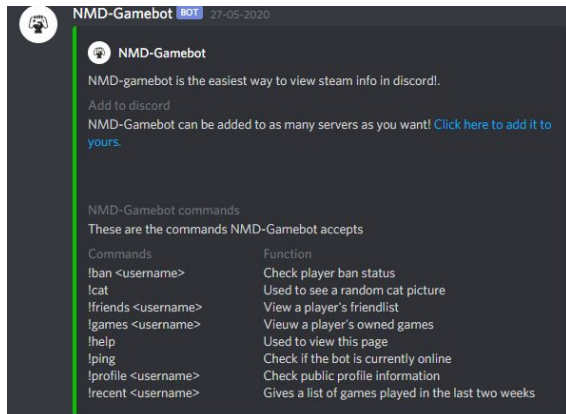
Voor deze bot hebben we op zich niet veel nodig gehad:

- Discord: om onze bot op te runnen (dit gebeurt enkel op een server)
- Vscod: als texteditor om de javascript en python code te maken en runnen
- Firebase: Als database voor gegevens op te slaan en door te sturen naar de Pi
- Raspberry Pi: Hierop wordt de python code gedraaid die verantwoordelijk is voor het opvragen van gegevens van Firebase en deze door te sturen naar de SenseHat
- SenseHat: De sensehat leest code uit en displayed de gegevens van de database hierop

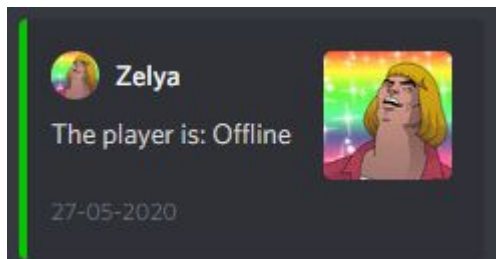
Als iemand geen Raspberry Pi en/of geen sensehat heeft dan is dat geen probleem, de bot zal nog altijd blijven werken maar er zullen enkele features niet aan bod komen dan zoals de foto tonen op de sensehat.

Inspiration

Ideaboard



Een bot op discord heeft meestal dezelfde layout als je deze aanroept in de server. Toch zorgen we ervoor dat de informatie die getoont wordt op een gebruiksvriendelijke manier getoont wordt zoals hier als je het commando *!help* intypt: *zie foto link*. De bot ziet er uit als persoon en geeft dus info weer over alle mogelijke commando's die je kunt uitvoeren.



Ook als je het profiel van iemand opvraagt door *!profile EFragnk1* wordt dit gedaan op een zo gebruiksvriendelijk mogelijke manier door niet te veel info weer te geven.

Develop

Code snippets: Discord bot (node.js)

> Detect if the user sends a message starting with “!” followed with one of the commands. (here only player-command)

```
client.on('message', msg => {
  if (msg.content.substring(0, 1) == '!') {
    var args = msg.content.substring(1).split(' ');
    var cmd = args[0];
    var username = args[1]

    switch (cmd) {

      ....
      ....

      case 'profile':
        msg.react('👍')
        findSteamId(username, msg, 1)

        break;
    }
  }
});
```

> Initialize firebase

```
firebase.initializeApp(firebaseConfig)
let serviceAccount =
require('./iot-eindproject-firebase-adminsdk-jxtfi-69fb32dd3c.json');
admin.initializeApp({
  credential: admin.credential.cert(serviceAccount)
});

let db = firebase.firestore();
```

> Function to find steam-id of requested user via Steam-API

```
function findSteamId(username, msg, integer) {
  https.get("https://api.steampowered.com/ISteamUser/ResolveVanityURL/v0001/?key=" +
    process.env.STEAM_TOKEN + "&vanityurl=" + username, res => {
    res.setEncoding("utf8");

    let bodySteam = "";

    res.on("data", steamData => {
      bodySteam += steamData;
    });

    res.on("end", () => {
      bodySteam = JSON.parse(bodySteam);

      if (bodySteam.response.success === 1) {
        searchId = (bodySteam.response.steamid)

        steamProfile(searchId, msg, integer)

      } else {
        msg.channel.send({
          "embed": {
            "title": "❌ No such account exists :(",
            "color": 640001
          }
        });
      }
    });
  });
}
```

> Show message in discord (after acquiring requested data)

```
msg.channel.send({
  "embed": {
    "title": realName,
    "description": "The player is: " + playerstatus,
    "color": 640001,
    "timestamp": new Date(),
    "thumbnail": {
      "url": avatar
    },
    "author": {
      "name": nickname,
      "url": profileUrl,
      "icon_url": avatar
    }
  }
});
```

Code snippets: Sense hat (Python)

> Detect if joystick is pressed and change “arrayindex” accordingly

```
for event in sense.stick.get_events():
    print/avatarArray)
    if event.action == "pressed":
        if event.direction == "right":
            if (arrayIndex == len/avatarArray) - 1 ):
                print(arrayIndex)
                arrayIndex = 0
            else:
                arrayIndex += 1
                print(arrayIndex)
        if event.direction == "left":
            if (arrayIndex <= 0):
                arrayIndex = len/avatarArray) - 1
                print(arrayIndex)
            else:
                arrayIndex -= 1
                print(arrayIndex)
```

> Get 32x32 image, convert to 8x8 and display on sensehat

```
response = requests.get/avatarArray[arrayIndex])
img = Image.open(BytesIO(response.content))
imgSmall = img.resize((8,8), resample=Image.BILINEAR)

rgb_img = imgSmall.convert('RGB')
image_pixels = list(rgb_img.getdata())
sense.set_pixels(image_pixels)
```

> On startup get array of user avatars (can be expanded to more data)

```
avatarArray = []

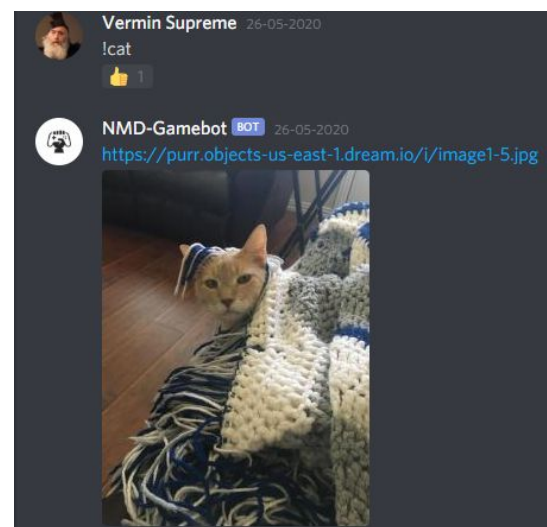
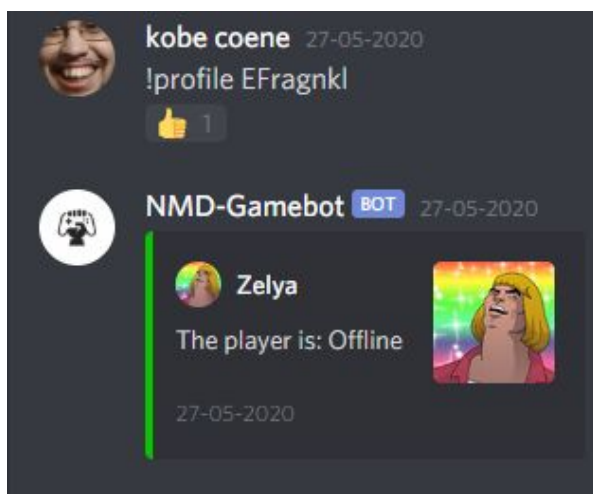
for doc in docsFirst:
    avatars = doc.to_dict().get('avatar')
    avatarArray.append(avatars)
```


> When there is an update in the database add new avatar to array

```
def on_snapshot(col_snapshot, changes, read_time):
    for doc in col_snapshot:
        avatars = doc.get('avatar')
        avatarArray.append(avatars)

    callback_done.set()
    print(avatarArray)
doc_watch = col_query.on_snapshot(on_snapshot)
```

Schermafbeeldingen



Deliver

Handleiding

1. Voorbereidingen
 - a. Registreer een bot account op het Discord developer portal (<https://discord.com/developers/applications>)
 - b. Genereer een steam api-key om de steam-api te kunnen gebruiken (<https://steamcommunity.com/dev/apikey>)
 - c. Maak een firebase databank aan (<https://firebase.google.com/>)
2. Discord bot maken met node.js
 - a. Minimum benodigde functies:
 - i. Message listener (voor het detecteren van commands)
 - ii. Firebase initieren
 - iii. Functie om het gevraagde steam-id op te zoeken
 - iv. Functie om via deze steam-id gegevens op te vragen
3. User informatie naar de firebase databank versturen
 - a. Na het ophalen van data, deze via een functie versturen naar firebase
4. Python code
 - a. Sense-hat linken aan raspberry pi
 - b. Firestore Database definieren
 - c. Minimum benodigde functies
 - i. Functie om foto te verkleinen naar 8x8 px
 - ii. Functie om foto's op te halen en in een array te steken
 - iii. Functie om Sensehat joystick movement te detecteren en te schakelen tussen foto's

Timesheet

De eerste uren programmeren waren in node.js, Een goed werkende bot is essentieel. Hier zijn dan ook een tiental uren in gekropen aangezien de Steam-api niet zo gebruiksvriendelijk is.

Daarna zijn we aan het werk gegaan met de Firebase connectie. Daar was soms een beetje verwarring doordat we zijn veranderd van firestore naar realtime database, en terug naar. naar firestore. Dit heeft wat tijd ingenomen maar uiteindelijk lukte het perfect.

Op woensdag 27 mei zijn we allebei dan naar de campus geweest om samen te werken aan de bot. Hier hebben we van 9:45 tot 12:00 gewerkt en hebben samen dingen overlopen die nog niet goed werkten, nog niet compleet waren, beter konden, ... Dit was een groot voordeel omdat we konden zien wat we misten en konden we de problemen samen oplossen Het enige nadeel was dat we niet tegelijk konden werken, maar dat zorgde niet voor problemen.

Na de meeting op de campus zijn er nog enkele kleine bugfixes gebeurd, die namen niet veel tijd in beslag en daarna zijn we aan het werk gegaan aan dit dossier en de presentatie.

Deploy

IoT-NMD-gamebot

A bot to fetch steam profile or game data

Add bot to your discord server

https://discordapp.com/api/oauth2/authorize?client_id=702489115383300126&permissions=67584&scope=bot

installations

```
$npm install
```

.env

```
don't forget to make a .env file
DISCORD_TOKEN=Yourtoken
STEAM_TOKEN=YourSteamApiToken

apiKey: [apiKey],
authDomain: [authDomain],
databaseURL: [databaseURL],
storageBucket: [storageBucket],
messagingSenderId: [messagingSenderId],
appId: [appId],
measurementId: [messagingSenderId]
```

Run Bot

```
$ node bot.js
```

Run Python script

```
$ python3 app.py
```