OŠ Brinje Grosuplje Ljubljanska cesta 40a, Grosuplje

Vodenje avta na daljavo z uporabo VR očal

Raziskovalna naloga

Področje: elektrotehnika, elektronika in robotika



Avtorji: Svit Verhovšek, Matic Pucelj in Svit Selan Mentorja: Matej Kastelic in Petra Gumzej

Vsebina

1.	POVZETEK	. 4
2.	UVOD	. 5
3.	OPIS REŠITVE	. 6
4.	VOZILO	. 8
	4.1 LEGO MINDSTORMS EV3	. 8
	4.2 RAC3 TRUCK	. 9
	4.3 CENTRALNA ENOTA VOZILA	. 9
	4.4 STREŽNIK	12
	4.5 TESTIRANJE	13
	4.6 UGOTOVITVE	14
5.	UPRAVLJANJE VOZILA	15
	5.1 VR OČALA	15
	5.2 PROGRAMIRANJE VR OČAL	16
	5.3 TESTIRANJE	19
	5.4 UGOTOVITVE	20
6.	SENZORJI	21
	5.1 VID	21
	5.2 MINI RAČUNALNIK	22
	5.3 KAMERA	22
	6.4 PRIKAZ NA VR OČALIH	23
	6.4.1 PRIKAZ VIDEA	23
	6.4.2 PRIKAZ SLIK	24
	6.5 UGOTOVITVE	25
7.	ZAKLJUČEK	26
8.	VIRI	27
9.	PRILOGA A: PROGRAM ZA VOZILO	28
10	PRILOGA B: NASTAVITVE ZA MOTION	29
11	. PRILOGA C: PRVA STRAN SPLETNEGA STREŽNIKA NA RASPBERRY PI	30

Kazalo slik

Slika 1: Načrt projekta	6
Slika 2: Lego Mindstorms EV3	8
Slika 3: Vozilo Rac3 Truck	9
Slika 4: Motorji na vozilu	9
Slika 5: Priprava Linuxa za Brick (korak 1)	10
Slika 6: Priprava Linuxa za Brick (korak 2)	10
Slika 7: Priprava Linuxa za Brick (korak 3)	10
Slika 8: SSH povezava na EV3	11
Slika 9: Zasedenost vrat	12
Slika 10: Odgovor strežnika na klic spletne storitve	14
Slika 11: VR očala Oculus Quest 2	15
Slika 12: 3D slika iz dveh zaslonov	15
Slika 13: Novi projekt - korak 1	16
Slika 14: Novi projekt - korak 2	16
Slika 15: Novi projekt - korak 3	17
Slika 16: Novi projekt - korak 4	17
Slika 17: Novi projekt - korak 5	17
Slika 18: Novi projekt - korak 6	18
Slika 19: Novi projekt - korak 7	
Slika 20: Novi projekt - korak 8	18
Slika 21: Novi projekt - korak 9	19
Slika 22: Klicanje spletne storitve iz Unreal Engine 4	19
Slika 23: Kamera in RaspBerry PI	21
Slika 24: RaspBerry PI Imager	22
Slika 25: Nastavitve za stream media source v Unreal Engine 4	24
Slika 26: Web browser komponenta v Unreal Engine 4	24
Slika 27: Test videa	24

1 POVZETEK

Namen raziskovalne naloge je ugotoviti, ali lahko z uporabo VR očal upravljamo vozilo na daljavo, pri tem pa uporabljamo 3D prikazovanje na VR očalih. Vozilo bi bilo zgrajeno iz Lego Mindstorms EV3 kock. Na njem bi namestili ev3dev Debian operacijski sistem in v programskem jeziku Python razvili program za sprejemanje ukazov iz VR očal ter upravljanje motorjev in senzorjev. Vozilo bi imelo svojo videokamero in bi bilo priklopljeno na RaspBerry PI 3B mini računalnik. Na RaspBerry PI bi namestili RaspBerry PI OS operacijski sistem in namestili program za prikazovanje slik iz kamere. Za VR očala bi uporabili Oculus Quest 2. Vsi sistemi bi bili povezani preko WIFI brezžične povezave. Na očalih bi razvili upravljalno aplikacijo v Unreal Engine 4. Aplikacija bi prikazovala pretočni video iz kamere. Na koncu smo ugotovili, da Oculus Quest 2 in Android nista sposobna prikazovati pretočnega videa, zato je bila raziskovalna naloga delno uspešna.

2 UVOD

V naši raziskovalni nalogi smo poskušali razviti vozilo iz komponent robotskega kompleta Lego Mindstorms EV3[1], ki bi ga upravljali na daljavo preko očal za navidezno resničnost (VR očal) in z uporabo priloženih dveh kontrolerjev.

Poskušali smo ugotoviti, ali 3D prikaz omogoča boljše upravljanje in koliko truda ter znanja je potrebno, da bi lahko razvili to vozilo. Na vozilu smo poskušali uporabiti različne senzorje, pri čemer je imela najpomembnejšo vlogo kamera.

Pri razvoju na VR očalih smo uporabili razvojno orodje Unreal Engine 4[2], na Lego Mindstorms EV3 pa programski jezik Python.

Pri raziskovalni nalogi smo hoteli odgovoriti na sledeča vprašanja:

- 1. Ali je Lego Mindstorms EV3 dovolj hiter, da lahko deluje kot strežnik in sprejema ukaze od VR očal?
- 2. Ali VR očala lahko vodijo in usmerjajo vozilo?
- 3. Ali VR očala lahko prikažejo sliko iz vozila?

Naše hipoteze so bile:

- 1. Računalniški sistem Lego Mindstorms EV3 bo zmogel hitro sprejemati ukaze in jih izvajati.
- 2. Upravljanje vozila je možno preko krmilnikov VR očal.
- 3. VR očala zmorejo prikazati video iz kamere, nameščene na vozilu, preko mini računalnika RaspBerry PI[3].

3 OPIS REŠITVE

Glede vozila smo se odločili, da bo veljalo sledeče:

- vozilo bo sestavljeno iz delov Lego Mindstorms EV3,
- vozilo bo imelo dva pogonska motorčka,
- vozilo bo imelo en motorček za določanje smeri,
- vozilo bo imelo senzor smeri,
- vozilo bo imelo senzor razdalje,
- vozilo bo imelo centralno enoto Brick kot notranji računalnik,
- Brick bo imel nameščen operacijski sistem Linux[4],
- Brick se bo programiral s Programskim jezikom Python,
- vozilo se bo upravljalo z VR očali,
- za VR očala smo izbrali Oculus Quest 2[5],
- kontrolerji na VR očalih bodo usmerjali vozilo,
- preko kamere bo imelo vozilo svoj vid,
- kamera bo povezana z mini računalnikom (RaspBerry PI),
- vse naprave bodo povezane preko brezžične povezave WIFI,
- na mini računalniku bo nameščen Linux in
- na mini računalniku bosta nameščena program za zajem slike in spletni strežnik.



Slika 1: Načrt projekta

Lego centralna enota Brick bi vozilu ukazovala, kaj naj počne. Vozilo bi lahko delalo naslednje:

- vozi naprej,
- vozi nazaj,
- miruje,
- zavija levo in
- zavija desno.

Razmišljali smo, da bi tudi spremljali tudi zunanje senzorje vozila in to prikazovali na VR očalih. V Lego Mindstorms EV3 paketu smo imeli senzor razdalje, senzor smeri in senzor za barve. Med izdelavo raziskovalne naloge bi ugotovili, katere senzorje bi v resnici potrebovali.

Ukaze bi enota Brick dobivala od kontrolerjev na VR očalih. En kontroler bi uporabljali za premik, drugega pa za usmerjanje. Premik bi naredili, če bi premaknili igralno ročico naprej ali nazaj. Obračanje na levo ali desno bi naredili z obračanjem igralne ročice na levo ali desno.

Kamera bi bila povezana z RaspBerry PI mini računalnikom, ki bi pretakal video na VR očala. Če to ne bi delovalo, bi poskušali prikazati trenutne slike iz kamere.

4 VOZILO

Ker smo doma že imeli Lego Mindstorms EV3, smo se odločili, da bo vozilo narejeno iz teh kock. Z njim smo se spoznavali že pri raziskovalni nalogi v 5. razredu. Nadgradili smo vozilo Rac3 Truck[6], za katerega smo imeli Lego načrt.

4.1 LEGO MINDSTORMS EV3

Lego Mindstorms EV3 so kocke, ki omogočajo zabavno in kreativno zasnovo robotov in vozil v kombinaciji z najnovejšo tehnologijo. Je popoln komplet za vse šolarje, ki želijo uporabiti svojo iznajdljivost in nadarjenost. Sistem omogoča, da razvijemo lastnega robota, ki hodi, govori in opravlja različne dejavnosti.

Škatla vsebuje 601 delov, računalnik imenovan Brick, tri motorje, daljinec, senzor za razdaljo, senzor za barve in tipko. Dodatno smo kupili še en senzor za barvo in senzor za ugotavljanje smeri.



Slika 2: Lego Mindstorms EV3

4.2 RAC3 TRUCK

To vozilo je sestavljeno po Lego navodilih na 32 straneh. Na vozilo smo pritrdili še kamero, RaspBerry PI mini računalnik in baterijo za napajanje mini računalnika. Vozilo se napaja s svojimi AA baterijami, ki se nahajajo v enoti Brick.



Slika 3: Vozilo Rac3 Truck

Za premikanje smo uporabili dva velika motorja, enega za levo kolo in enega za desno. Dodatni srednji motor smo uporabili za volan.



Slika 4: Motorji na vozilu

4.3 CENTRALNA ENOTA VOZILA

Lego centralna enota je računalnik v vozilu, ki pošilja ukaze motorjem in bere vrednosti senzorjev. Na njem smo namestili operacijski sistem Linux (različico EV3 Micropython – ev3dev). Tega smo dobili na spletni strani EV Micropython [7]. Datoteko s sliko operacijskega sistema smo nato ekstrahirali na disk.

Na spletni strani programa Etcher[8] smo naložili in namestili aplikacijo, s katero pripravimo SD kartico z Linuxom za enoto Brick.



Potem smo zagnali program Etcher in na njem izbrali našo EV3 sliko (IMG datoteko).

Slika 5: Priprava Linuxa za Brick (korak 1)

Nato smo izbrali spominsko kartico SD.

Stcher		-	
		Need	Help? 🌣
SELECT IMAGE	SELECT DRIVE	FLASH IMAGE	
1	2		
ev3-ev3dev-jessie-2015-12- 30.img.zip	Connect a drive		
Change			
TCHER IS		r 😒 resin.io 1.0.0-6-8	a.7

Slika 6: Priprava Linuxa za Brick (korak 2)



Na koncu pa smo izvedli zapis na spominsko kartico s pritiskom na gumb Flash!.

Slika 7: Priprava Linuxa za Brick (korak 3)

Pripravljeno SD spominsko kartico smo vstavili v Brick in jo zagnali. Ko se je sistem naložil, smo mu nastavili Wi-Fi in mu dodelili IP naslov 192.168.178.44.

Preko računalniškega terminala (RemoteNG) smo se povezali na EV3 preko SSH povezave. Uporabniško ime je bilo *robot*, geslo pa *maker*.



Slika 8: SSH povezava na EV3

Programski jezik Python je že del tega operacijskega sistema.

Najprej smo naredili nadgradnjo sistema z ukazom:

sudo apt-get update

Nato smo namestili Flask, ki je ogrodje (framework) za razvoj spletnih mikrostoritev za Python. Z njim pišemo spletne storitve. Namestili smo ga z ukazom:

sudo apt-get install python3-flask

Sledila je namestitev python namestitvene aplikacije PIP:

sudo apt install python3-pip

Na koncu smo še namestili knjižnico za delo z EV3:

pip3 install pybricks

4.4 STREŽNIK

Program za strežnik smo napisali sami z uporabo programskega jezika Python. Pri tem smo uporabili knjižnico Flask. Naš strežnik posluša na vratih 5000.

Na začetku naložimo knjižnice za Flask in za vozilo:

```
from flask import Flask
from pybricks import ev3brick as brick
from pybricks.ev3devices import *
from pybricks.parameters import *
from pybricks.tools import *
from pybricks.robotics import *
```

Da lahko delamo z vozilom, ustvarimo objekt brick:

```
brick = EV3Brick()
```

Za premikanje vozila ustvarimo objekt motor, ki krmili dva velika motorja na vratih A in B:

```
levi_motor = Motor(Port.A)
desni_motor = Motor(Port.B)
motor = DriveBase(levi motor, desni motor, 42, 140)
```

Za usmerjanje vozila ustvarimo objekt volan, ki krmili en srednji motor na vratih C:

```
volan = Motor(Port.C)
```



Slika 9: Zasedenost vrat

Za premikanje vozila kličemo funkcijo drive(), ki sprejme dve vrednosti:

• hitrost in

• rotacijo.

Hitrost je lahko pozitivna ali negativna številka, ker se lahko vrti v obe smeri. Da se vozilo začne premikati naprej, moramo nastaviti negativno vrednost -50: motor.drive(-50, 0)

Da se vozilo začne premikati nazaj, moramo nastaviti pozitivno vrednost 50: motor.drive(50, 0)

Da se vozilo ustavi, moramo nastaviti vrednost na 0:

```
motor.drive(0, 0)
```

Za usmerjanje vozila kličemo funkcijo run_angle(), ki sprejema dve vrednosti:

- hitrost in
- kot.

Da se vozilo začne premikati desno, mu moramo kot nastaviti na pozitivno vrednost 50°:

```
volan.run_angle(100, 50)
```

Da se vozilo začne premikati levo, mu moramo kot nastaviti na negativno vrednost -50°:

```
volan.run angle(100, -50)
```

Strežnik odgovori z JSON sporočilom:

{"status": 1}

4.5 TESTIRANJE

Pred testiranjem moramo zagnati strežnik:

sudo python3 vozilo.py

Za testiranje spletnih storitev kličemo sledeče povezave v brskalniku:

- za naprej: <u>http://192.168.178.44:5000/naprej</u>
- za nazaj: <u>http://192.168.178.44:5000/nazaj</u>
- za ustavitev: <u>http://192.168.178.44:5000/stop</u>
- za zavijanje desno: <u>http://192.168.178.44:5000/desno</u>
- za zavijanje levo: <u>http://192.168.178.44:5000/levo</u>



Slika 10: Odgovor strežnika na klic spletne storitve

4.6 UGOTOVITVE

Ker je Python interpretiran programski jezik (program se šele pri zagonu pretvori iz besedila v računalniške ukaze – strojno kodo), potrebuje 27 sekund za zagon našega strežnika. Strežnik se odziva hitro, ampak ker micropython ni enak Python-u, ne zna zaganjati Flaska. Zaradi tega je naša prva hipoteza hipoteza, da računalniški sistem robotskega kompleta Lego Mindstorms EV3 zmore hitro sprejemati ukaze in jih izvajati, zavrnjena.

5 UPRAVLJANJE VOZILA

Vozilo lahko upravljamo preko VR očal Oculus Quest 2. VR očala se uporabljajo za prikaz navideznega 3D sveta[9]. Prednost VR očal je to, da svet vidimo v treh dimenzijah. Zato smo se odločili, da jih bomo uporabili pri naši raziskovalni nalogi.



Slika 11: VR očala Oculus Quest 2

5.1 VR OČALA

VR očala sestavljata dva zaslona, za vsako oko eno, ki prikazujeta razmaknjeni sliki tako, da naše oko oziroma možgani obe sliki vidijo kot eno sliko, ki je 3D.



Slika 12: 3D slika iz dveh zaslonov

Za našo aplikacijo smo izbirali med programerskimi orodji:

- Unreal Engine in
- Unity 3D.

Izbrali smo Unreal Engine 4. To je popularno programersko orodje, ki ga uporablja veliko razvijalcev igric. Z njim je Epic Games razvil igrico Fortnite, ki je ena izmed najbolj popularnih strelskih iger na svetu.

5.2 PROGRAMIRANJE VR OČAL

Na začetku smo morali namestiti aplikacijo Epic Store[10], preko katere smo nato namestili Unreal Engine 4.27.

Ko smo zagnali Unreal Engine, smo naredili nov projekt. Izbrali smo tip projekta za igre:

	Games Start your game development journey with one of our key classes, levels, and examples.
5	Film, Television, and Live Events Choose from templates and examples for nDisplay, VR Scouting, and virtual production workflows.
	Architecture, Engineering, and Construction Select a starting point for multi-user design reviews, photorealistic architectural design visualizations, sunlight studies, or stylized renderings.
.	Automotive, Product Design, and Manufacturing Find templates for multi-user design reviews, photobooth studio environments, and product configurators.

Slika 13: Novi projekt - korak 1

Za predlogo smo izbrali navidezno resničnost:



Slika 14: Novi projekt - korak 2

Projektu smo dali ime in mu dodali začetne komponente:

Project Settings					
With Starter Content	Enable to include an additional content pack containing simple placeable meshes with basic materials and textures. You can also add the Starter Content to your project later using Content Browser .				
	Select a location for your project to be stored. C:\Users\Svit\Documents\Unreal Projects				
	Back	Create Project	Cancel		

Slika 15: Novi projekt - korak 3

Ko se je projekt naložil, smo v nastavitvah za projekt nastavili začetno mapo, ki jo doda predloga za VR očala (VR svet):

Designet Mana O Madaa	
^a Project - Maps & Modes	
Default maps, game modes and other map related settings.	
$\eta_{ m III}$ These settings are saved in DefaultEngine.ini, which is currently writable.	
⊿ Default Modes	
Default GameMode	GameModeBase ▼ ← ♀ +
▲ Selected GameMode	
Default Pawn Class	DefaultPawn - + D +
HUD Class	+ Q → ▼
Player Controller Class	PlayerController
Game State Class	GameStateBase ▼ ← ♀ +
Player State Class	PlayerState 🔷 🔶 🗩 🕇
Spectator Class	SpectatorPawn - + 0 +
⊿ Default Maps	
Editor Startup Map	MotionControllerMap -
Game Default Map	MotionControllerMap ← ₽

Slika 16: Novi projekt - korak 4

Za podprti platformi smo nastavili Android, ki ga uporablja Oculus Quest 2, in Windows:



Slika 17: Novi projekt - korak 5

Na mobilnem delu smo nastavili način dela za grafiko (MSAA je 4x):

4	⊿ Mobile		
	Disable vertex fogging in mobile shaders	v	
	Maximum number of CSM cascades to render	2	2
	Mobile MSAA	4x MSAA	-
	Allow Dithered LOD Transition		
	Support Software Occlusion Culling		
	Support movable light CSM shader culling	✓	
	Mobile Ambient Occlusion		
	Planar Reflection Mode	Usual	-
	Supports desktop Gen4 TAA on mobile		

Slika 18: Novi projekt - korak 6

Nastavili smo ime paketa, ki je povezan z računom na portalu za razvijalce za Android. Minimalna verzija Androida mora biti 23, ciljana verzija Androida pa 28:

Accept SDK License	
Build Folder	Open Build Folder
Android Package Name ('com.Company.Project', [PROJECT] is replaced with project name)	com.Pronic.RazVr
Store Version (1-2147483647)	1
Store Version offset (armv7)	0
Store Version offset (arm64)	0
Store Version offset (x86_64)	0
Application Display Name (app_name), project name if blank	
Version Display Name (usually x.y)	1.0
Minimum SDK Version (19=KitKat, 21=Lollipop)	23
Target SDK Version (19=KitKat, 21=Lollipop)	28
Install Location	Internal Only 🔻
Enable Lint depreciation checks	
Package game data inside .apk?	
Generate install files for all platforms	

Slika 19: Novi projekt - korak 7

Na koncu smo vklopili vtičnik za Oculus VR:



Slika 20: Novi projekt - korak 8

Za klicanje spletnih storitev smo naložili vtičnik VaRest[11] in ga vklopili:



Določili smo, da ob pritisku na sprednjo tipko levega krmilnika, pokličemo spletno storitev <u>http://192.168.178.44:5000/naprej</u>, ki začne premikati vozilo.



Slika 22: Klicanje spletne storitve iz Unreal Engine 4

Podobno smo naredili za desno tipko na krmilniku, ki pokliče spletno storitev <u>http://192.168.178.44:5000/stop</u> za ustavitev. Ostalih krmilnikov tekom izdelave raziskovalne naloge nismo programirali, ker nas je samo zanimalo, ali krmilniki delujejo.

5.3 TESTIRANJE

Za testiranje upravljanja s krmilnikom smo preverili izpise ob pritisku na ustrezen gumb krmilnika.

Primer izpisa na strežniku, ki se pojavi, če VR očala uspešno pokličejo spletno storitev na RaspBerry PI mini računalniku:

Naprej 192.168.178.66 - - [20/Mar/2022 14:32:23] "GET /naprej HTTP/1.1" 200 -

5.4 UGOTOVITVE

Uspelo nam je poklicati spletno storitev na enoti Brick in s tem smo potrdili drugo hipotezo, da je upravljanje vozila možno preko krmilnikov VR očal.

Z VR očali je mogoče upravljati Lego Mindstorms EV3 vozilo.

6 SENZORJI

Lego Mindstorms EV3 vsebuje kar nekaj senzorjev:

- senzor razdalje,
- senzor barve,
- senzor dotika in
- senzor smeri.

Na žalost nam noben od teh senzorjev ni koristil, saj smo potrebovali senzor za zaznavo okolice, kot sta kamera in LIDAR senzor. Ker je LIDAR senzor pretežak za uporabo za osnovnošolce, smo se odločili, da bomo poskušali s kamero.

6.1 VID

Za namen raziskovalne naloge smo uporabili Logitech C270 spletno USB kamero, ki stane manj kot 50 €.

Ker Brick ni dovolj močan, da bi lahko posredoval VR očalom pretočni video, in ker so bila edina USB vrata določena za WiFi ključ, smo kamero priključili na nadomestni mini računalnik.



Slika 23: Kamera in RaspBerry PI

6.2 MINI RAČUNALNIK

Za mini računalnik smo uporabili RaspBerry PI 3B.

Ta mini računalnik ima sledeče lastnosti:

- 64-bitni štiri jedrni procesor ARM Cortex-A53 (1,2 GHz),
- 1 GB RAM,
- 4 USB vrata,
- HDMI za monitor,
- 32 GB SD kartica kot disk in
- WiFi in Ethernet.

Najprej smo namestili operacijski sistem RaspBerry PI OS, ki smo ga našli na RaspBerry PI strani[12]. Namestili smo jo preko aplikacije RaspBerry PI Imager in jo zagnali.

Raspberry Pi Imager v1.8						
	Operating System CHOOSE OS	Storage CHOOSE STORAGE				

Slika 24: RaspBerry PI Imager

Izbrali smo RaspBerry PI OS with Desktop. Aplikacija nam je na SD kartico zapekla sliko operacijskega sistema. SD kartico smo vstavili v mini računalnik in ga zagnali.

Kamero smo nato priklopili na USB vrata. Za napajanje smo uporabili 10000 mAh akumulator. Mini računalniku smo določili IP naslov 192.168.178.69 in mu vklopili SSH.

6.3 KAMERA

Da RaspBerry PI lahko zajame sliko in jo pretaka VR očalom, je potrebno namestiti aplikacijo Motion. Ta iz kamere zajame slike in iz njih sproti naredi MJPEG video. Prav tako zna zajemati JPEG slike. Preizkusili smo dva načina prenosa videa:

- 1. Video smo pretakali kot MJPEG.
- 2. Zajete slike smo pošiljali eno za drugo preko spletne strani, zaradi česar smo namestili spletni strežnik Apache.

Najprej smo namestili spletni strežnik:

```
sudo apt install apache2 -y
```

Spletni strežnik bere strani iz direktorija /var/www/html. Tam se nahaja datoteka index.html, ki smo jo popravili, da prikazuje zajeto sliko in to ponavlja v nedogled.

Nato smo namestili aplikacijo Motion:

sudo apt install motion

Popravili smo konfiguracijsko datoteko /etc/motion/motion.conf:

target_dir /var/www/html videodevice /dev/video0 width 320 height 240 framerate 10 text_left OŠ BRINJE text_right Raziskovalna naloga picture_output on picture_filename snapshot movie_output off

Nato smo zagnali Motion:

sudo motion

Popravili smo tudi prvo stran spletnega strežnika, do katerega pridemo s povezavo <u>http://192.168.178.69</u>.

6.4 PRIKAZ NA VR OČALIH

Poskušali smo prikazati video in spletno stran z zajetimi slikami.

6.4.1 PRIKAZ VIDEA

V Unreal Enginu smo dodali MediaPlayer in StreamMediaSource. V nastavitvah smo vpisali povezavo na pretočni video <u>http://192.168.178.69:8081</u>.

. Distinguist		
a Platforms		
	📫 Android	Automatic -
	द् दि HoloLens	Automatic -
	📹 ios	Automatic -
	🔔 Linux	Automatic -
Player Overrides	🔔 Linux (AArch64)	Automatic -
	🐥 Lumin	Automatic -
	🕰 Mac	Automatic -
	🥳 tvOS	Automatic -
	द Windows (64-bit)	Automatic -
I Stream		
	http://192.168.178.69:808	1 5

Slika 25: Nastavitve za stream media source v Unreal Engine 4

Video nam je uspešno pokazalo na osebnem računalniku, ne pa tudi na Oculus Quest 2, na katerem teče Android. Na spletu smo našli podatek, da ta problem še ni bil rešen.

6.4.2 PRIKAZ SLIK

Poskušali smo tudi prikazati osnovno spletno stran z zajetimi slikami z uporabo web browser komponente.



Slika 26: Web browser komponenta v Unreal Engine 4

Tudi tukaj je delovalo na osebnem računalniku, na Oculus Quest 2 pa ne.



Slika 27: Test videa

6.5 UGOTOVITVE

Uspelo nam je usposobiti pretočni video iz kamere, a na žalost VR očala niso sposobna tega videa prikazati. Mislimo, da je to zaradi tega, ker so VR očala v glavnem namenjena računalniškim igram. Hipoteza, da VR očala znajo prikazati video iz kamere, nameščene na vozilu, preko mini računalnika RaspBerry PI, je zavrnjena.

7 ZAKLJUČEK

Na začetku smo upali, da bomo uspeli potrditi vse tri hipoteze. Programski jezik Python smo spoznali že v 6. razredu pri pouku predmeta Računalništvo, zato programiranje na Lego Mindstorms EV3 ne bi smel biti problem. Ker smo kamero priklopili na RaspBerry PI mini računalnik, smo morali ugotoviti, kako lahko računalnik zajema sliko na kameri in jo predvaja naprej na računalniškem omrežju. Na koncu pa smo morali pretočni video prikazati na VR očalih in s kontrolerji upravljati vozilo.

Na Mindstorms EV3 smo s Python-om naložili Linux, na katerem smo uspešno razvili spletne storitve za upravljanje vozila. Ker smo vozilo upravljali s knjižnico PyBricks, smo morali Python zamenjati z micropythonom. Na žalost ta ne podpira Flaska, zato vozila nismo mogli premikati. Rešitev bi bila zamenjava EV3 knjižnice ali pa zamenjava programskega jezika za npr. C, Go ali kaj drugega.

Na Unreal Enginu 4 smo razvili 3D aplikacijo, ki sprejema ukaze na obeh krmilnikih VR očal in kliče spletne storitve preko HTTP povezave. Ta del raziskovalne naloge je bil uspešen. Tukaj bi lahko HTTP zamenjali s stalno povezavo za hitrejše pošiljanje ukazov, toda uporaba stalne povezave je bila za nas prezahtevna.

Ne glede na različne poskuse uporabe aplikacije Motion, nam ni uspelo prikazati slike od kamere na VR očalih, kljub temu da na osebnih računalnikih to deluje.

Pri raziskovalni nalogi sta bili dve hipotezi zavrženi, ena pa potrjena.

V naslednjem letu se želimo še dodatno poglobili v VR očala in jih uporabiti npr. pri učenju na šoli.

VIRI

[1] Lego Mindstorms EV3 - Wikipedia. Dosegljivo: https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3. Zadnji popravek: 2022. [Dostopno:12.2.2022]

[2] Unreal Engine 4 – Wikipedia. Dosegljivo: <u>https://en.wikipedia.org/wiki/Unreal_Engine</u>. Zadnji popravek: 2022. [Dostopno: 13.2.2022]

[3] RaspBerry PI – Wikipedia. Dosegljivo: <u>https://en.wikipedia.org/wiki/Raspberry_Pi</u>. Zadnji popravek: 2022. [Dostopno: 15.2.2022]

[4] Linux – Wikipedia. Dosegljivo: <u>https://en.wikipedia.org/wiki/Linux</u>. Zadnji popravek: 2022. [Dostopno: 15.2.2022]

[5] Oculus Quest 2 – Wikipedia. Dosegljivo: <u>https://en.wikipedia.org/wiki/Oculus_Quest_2</u>. Zadnji popravek: 2022. [Dostopno: 15.2.2022]

 [6] Lego Rac3 Truck navodila za sestavljanje. Dosegljivo: <u>https://www.lego.com/cdn/cs/set/assets/blt8d8677b8321b803e/RAC3_TRUCK.pdf</u>. 2013.
 [Dostopno: 16.2.2022]

[7] EV3 Micropython. Dosegljivo: <u>https://pybricks.com/ev3-micropython/startinstall.html</u>. [Dostopno: 25.2.2022]

[8] Etcher. Dosegljivo: https://etcher.io/. [Dostopno: 25.2.2022]

[9] VR očala – Wikipedia. Dosegljivo: <u>https://en.wikipedia.org/wiki/Virtual_reality_headset</u>. Zadnji popravek: 2022. [Dostopno: 1.3.2022]

[10] Epic Store. Dosegljivo: <u>https://store.epicgames.com/en-US/</u>. [Dostopno: 2.3.2022]

[11] VaRest. Dosegljivo: <u>https://www.unrealengine.com/marketplace/en-US/product/varest-plugin</u>. [Dostopno: 4.3.2022]

[12] RaspBerry PI OS. Dosegljivo: <u>https://www.raspberrypi.com/software/</u>. [Dostopno: 6.3.2022]

PRILOGA A: PROGRAM ZA VOZILO

```
#!/usr/bin/env pybricks-micropython
from flask import Flask
from pybricks.hubs import *
from pybricks.ev3devices import *
from pybricks.parameters import *
from pybricks.tools import *
from pybricks.robotics import *
app = Flask( name )
brick = EV3Brick()
levi motor = Motor(Port.A)
desni motor = Motor(Port.B)
volan = Motor(Port.C)
motor = DriveBase(levi motor, desni motor, 42, 140)
@app.route("/naprej")
def naprej():
    motor.drive(-50, 0)
    return "{\"status\":1}"
@app.route("/nazaj")
def nazaj():
    motor.drive(50, 0)
    return "{\"status\":1}"
@app.route("/stop")
def stop():
    motor.drive(0, 0)
    return "{\"status\":1}"
@app.route("/desno")
def desno():
    volan.run angle(100, 50)
    return "{\"status\":1}"
@app.route("/levo")
def levo():
    volan.run angle(100, -50)
    return "{\"status\":1}"
print("Started")
if __name__ == '__main__':
    app.run()
```

PRILOGA B: NASTAVITVE ZA MOTION

daemon on setup mode off log level 6 target dir /var/www/html videodevice /dev/video0 width 320 height 240 framerate 10 text left OŠ BRINJE text_right Raziskovalna naloga emulate motion on threshold 1500 despeckle filter EedDl minimum motion frames 1 event_gap 60 pre capture 3 post capture 0 picture output on picture_filename snapshot movie_output off movie max time 3600 movie quality 45 movie codec mkv movie filename movie webcontrol port 8080 webcontrol localhost off webcontrol_parms 0 stream_port 8081 stream localhost off

PRILOGA C: PRVA STRAN SPLETNEGA STREŽNIKA NA RASPBERRY PI

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta http-equiv="refresh" content="0" >
    <title>Slika</title>
    <style>
body {
 background-color: #000000;
}
   </style>
 </head>
 <body >
   <img src="snapshot.jpg" width="80%" height="80%"></img>
 </body>
</html>
```