

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ KATEDRA

**Nesudėtingų robotų prototipų projektavimas**

**Simple robot prototype design**

Bakalauro baigiamasis darbas

Atliko:	Osvaldas Žiukas	(parašas)
Darbo vadovas:	Irus Grinis	(parašas)
Darbo recenzentas:	prof. dr. Romas Baronas	(parašas)

Vilnius – 2015

## Santrauka

Šio darbo tikslas yra sukurti pigų robotą, kuris galėtų būti valdomas arba vartotojo arba programos ir atliktu numatytas pagal vartotoją užduotis, tiek saugos tiek mokomosiose srityse. Visas darbas susideda iš roboto, valdiklio, kameros ir programinės įrangos skirtos roboto valdymui. Siekiant sumažinti roboto savikainą, buvo panaudota radijo bangomis valdoma mašinėlė kurią valdo radijo bangomis pultelis, kontroliuojamas *ARDUINO* mikrokontroleriu. Pritaikant trečiųjų šalių bibliotekas ir rašyta programinį kodą, buvo siekiama naudojant vaizdo kamerą užfiksuoti pasirinkta informaciją: judėjimą, šviesos pokyčius ar tiesiog tirti, kad nuvykęs robotas galėtų atlikti pritaikytus darbus. Projektas palaiko kelių robotų valdymą ir keliolikos kamerų vaizdo medžiagos apdorojimą. Techninė projekto daliai buvo pasirinkti ir pritaikyti kaip įmanoma paprastesni komponentai, nereikalaujantys didelio išmanymo šioje srityje ir didžiausia darbo dalis buvo skirta atlikti programinės įrangos rašymui. Šiuo projektu tikimasi padidinti efektyvumą mokymosi srityje, bei pritaikyti pramonėje ar saugoje.

Raktiniai žodžiai: Robotas, vaizdo kamera, *Arduino*, *JavaFx*, *Java*, radijo bangomis valdoma mašinėlė, švietimas, *Ardulink*.

## Summary

The goal of this project is to create a cheap robot which could be controlled by user or program and would do tasks assigned to it in both security and teaching spheres. Whole project consist of the robot, controller, video camera and program which is designed to take data from devices it's connected, make necessary calculations and control robot respectively. For this project as robot was used a car controlled by radio waves. Arduino was chosen as a controller for purpose of communication between a remote controller of the car and computer itself. Using video camera and the third part library with program code to interpret camera's visual material the goal was to detect wanted information such as movement, light changes in the chosen area where programmed robot could perform tasks assigned to it. This project supports multiple cameras and robots at same time. For technical part of the project there was used simple components and parts which dont require much knowledge in that sphere and most important work was done on creating program for controlling radio car itself. Purpose of this project is to increase efficiency of security systems, provide tool for exploring certain environemnt, guidance in teaching children at schools and providing a toy to play with for a cheap price.

Keywords: Robot, video camera, *Arduino*, *JavaFx*, *Java*, radio remote control, education, *Ardulink*.

## **Turinys**

Įvadas.....	5
Darbo eiga.....	8
1. Buvusio sistemos projekto aptarimas ir analizė .....	8
1.1. Programos valdymo komponentai.....	8
1.2. Platformos aptarimas.....	10
2. Naujos sistemos aptarimas ir analizė.....	10
2.1. Naujos sistemos reikalavimai.....	10
2.2. Naujos sistemos projektavimas .....	11
2.3. Naujos sistemos projektavimas .....	11
2.4. Programos valdymo komponentų aptarimas .....	12
1.1. Naujo roboto ir kameros pridėjimas.....	14
2.5. Valdymo režimai .....	15
2.6. Platformos vertinimas .....	16
2.7. Kameros valdymas .....	16
2.8. Radijo siųstuvo valdymas.....	17
3. Valdymo schema.....	17
Rezultatai ir išvados .....	19
Šaltiniai.....	20
Sąvokų ir santrumpų apibrėžimas.....	21

## Ivadas

Robotas – mechaninis prietaisas, galintis atlikti naudotojo užprogramuotas fizines užduotis. Robotas gali būti valdomas tiesiogiai žmogaus arba veikti valdomas užprogramuoto kompiuterio jau paruoštomis valdymo komandomis. Robotai gali atlikti užduotis, kurios yra pavojingos žmogui, dirbti žmonėms kenksmingoje aplinkoje ar atlikti užduotis kurios yra sunkios ar neįgyvendinamos. Jie taip pat gali atlikti monotoniškas, dažnai pasikartojančias užduotis, tokiu būdu pigiai pakeisdami žmogaus rankų darbą.

Robotai gali būti pritaikyti daugybėje sričių: pradedant vaikų priežiūra, švietimu, ligonių slauga, apsauga, ar kitų prietaisų valdymu ir priežiūra, baigiant – karyba, ūkių apsauga ir pagalba stichinėse situacijose. Pritaikant skirtingas metodikas kuriami kelių rūšių robotai: vieni imituoja žmones, jų išvaizdą ar elgesį, pakeisdami juos slaugos, vaikų priežiūros, ar bendravimo srityse, kiti imituoja gyvūnus, siekiant panaudoti geriausias jų konstrukcines savybes tokias kaip lipimas sienomis ar pralindimas pro menkiausias plyšius, taip padedami stichinėse nelaimėse, o sekantys kuriami pagal vartotojo siekius, siekiant išnaudoti paprastus ir pigius sprendimus pasiekti norimiems rezultatams. Tokie robotai dažniausiai turi vikšrus, variklius su ratais ar sraigtus.

Tačiau nors ir esant didžiulei robotų pasiūlai vis dar susiduriama su brangiais siūlomais sprendimais, kurie būna neprieinami, neinvestuojant didelių išteklių ar žinių. Su šia problema dažniausiai susiduria mažos įmonės, mokyklos ar namų ūkiai, kurie negali sau to leisti. Tačiau panaudojant pigias medžiagas ar pritaikant esamas konstrukcijas ir jas patobulinant, galima pasiekti pakankamai gerų rezultatų už mažą kainą. Mano darbo esmė yra pasiūlyti būtent tokį sprendimą suinteresuotai auditorijai.

Šio darbo tikslas yra trečio kurso praktinio darbo tęsimas, tolesnis platformos vystymas ir naujų funkcijų įdiegimas. Pridedant didesniu valdymo, sekimo, objektų atpažinimo ir vaizdinės informacijos apdorojimo funkcijas. Šis projektas skirtas kelioms pagrindinėms sritims:

- Lavinamoji švietimo – atsižvelgiant į technologijų poreikio augimo tendenciją, vienas iš tikslų ir yra paruoštą sistemą pateikti švietimo įstaigoms. Tik turinčios didelį biudžetą švietimo įstaigos gali sau leisti skirti robotikai ir naujausių technologijų mokymui lėšų. Ši situacija susidaro dėl to, kad pagrindiniai siūlomi sprendimai yra iš užsienio komercinių įmonių, kurios siekia pasididinti pelną. Šis projektas skirtas išspręsti šią bėdą, leidžiant pakeisti brangius robotus į pigesnius pakaitalus naudojant radijo

bangomis valdomas mašinėlės ar panašius prietaisus kuriuos galima nesunkiai valdyti valdikliu.

- Stebėjimui, saugai ir tyrimams – mašinėlę galima nesunkiai užprogramuoti taip, kad ji patruliuotu nustatytu maršrutu, tad mašinėlė pritaikoma kaip papildoma apsaugos priemonė prie saugos sistemų. Vartotojas taip pat gali pats valdyti mašinėlę ir nusiųsti ją į pavojingas teritorijas atlikti stebėjimo darbų.
- Laisvalaikio ir pramogoms – mašinėlę galima pritaikyti ir žaidimams bei kitoms pramogoms, pavyzdys: kuriant maršrutus ir varžantis kas pirmesnis suprogramuos komandų sąrašą teisingai įveikti pasirinktam maršrutui.

*Mikrokontroleris* – ATmega, PIC ir dar panašios mikroschemos – vienakristalės mikro skaičiavimo mašinos. Valdiklis – plokštė, kurioje įlituota MK mikroschema. Valdiklis, kaip jau ir sako pavadinimas, elektroninis įtaisas, kuris kažką valdo, reaguodamas į pasikeitusius aplinkos parametrus. Aplinkos parametrai nuskaitomi specialiais davikliais, dar kitaip vadinamais jutikliais. Valdymo įtaisai: motorai, elektroniniai moduliniai komponentai, įvairios relės, šviesos šaltiniai, *optoelektronika* ar *servo varikliai* – viskas ką galima valdyti paduodant elektros impulsus. Pavyzdžiui, kondicionieriuje yra valdiklis, kuris įjungia/išjungia oro šildymą / šaldymą, drėgmės reguliatorių tada, kai to reikia. Kondicionierius, MP3 grotuvas, GPS navigatorius, net ir mikrobangų krosnelė ar šaldytuvas – specializuoti valdikliai.

*Arduino* – tai vienas iš valdiklių, kurį galima priversti kažką daryti. Paprastai prijungiama prie kompiuterio platforma, naudojanti *Atmel* firmos mikrokontrolerius, kurie pasižymi savo itin didelėmis galimybėmis. Tai atviro kodo platforma, todėl ji ir tapo populiari visame pasaulyje. Ši platforma tai instrumentas, suteikiantis kompiuteriui didesnes galimybes „jausti“ ir valdyti fizinį pasaulį. *Arduino* naudoja C ir C++ programavimo kalbos „mišinį“. Išmokti šią programavimo kalbą nėra sudėtinga. Dar vienas *Arduino* privalumas – tai, kad jam yra sukurti pridedami skydeliai, kurie praplečia *Arduino* galimybes, leisdami prijungti prie interneto, WiFi ar Bluetooth ryšio. Taip pat suteikia galimybę naudoti SD atminties korteles ir dar daug visokių kitų galimybių. Su *Arduino* lengva kurti ir testuoti savo idėjas, nes sujungus prietaisus galima nesunkiai pamatyti galutinį rezultatą. Nereikia pamiršti kad privaloma laikytis tam tikrų taisyklių, kad nesugadinti valdiklio.

Išsirinkus valdiklį projektui, reikėjo pasirinkti sprendimą robotui, kurį jis valdys. Renkamasi buvo iš kelių variantų: suprojektuoti ir surinkti roboto modelį, nusipirkti jau surinktą,

ieškoti alternatyvių pasiūlymų. Galiausiai buvo nuspręsta panaudoti kaip modelį patį paprasčiausią variantą radijo bangomis valdoma mašinėlė, kurių pilna parduotuvėse.

Radijo bangomis valdoma mašinėlė – tai prietaisas, kurį galima įvardinti pačiu paprasčiausiu robotu. Jį sudaro motoras ar motorai, ratai, platforma, gaviklis, ir valdiklis. Iš pultelio siunčiamos radijo bangos, per gaviklį perduodamos į mašinėlės valdiklį, kontroliuoja mašinėlę, dėl šių savybių, mašinėlių kaina ypatingai maža, o jos naudojamos plačiai. Išsirinkus kas bus naudojamas kaip parodomasis robotas, sekantis žingsnis buvo kamera.

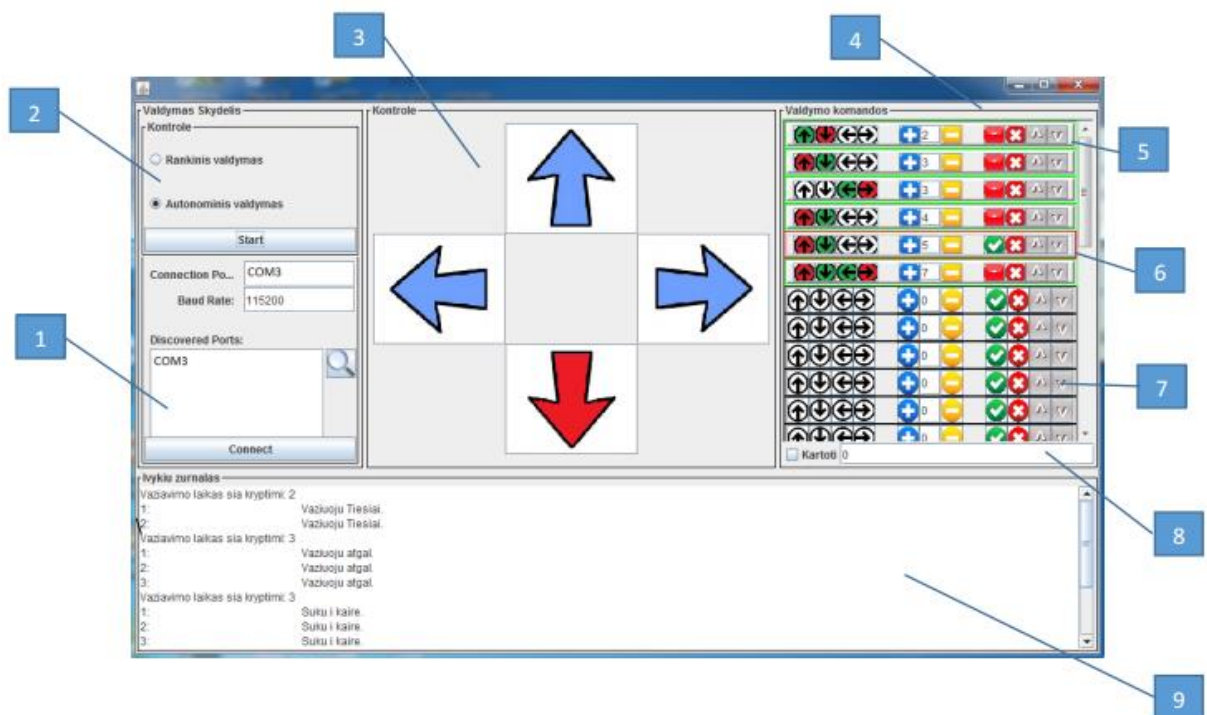
Internetinė vaizdo kamera – vaizdo ar fotokamera, dažniausiai jungiama prie kompiuterio naudojant *USB* jungtį ir leidžianti realiu laiku perteikti vaizdą. Dėl mažos rezoliucijos (352x288) tokios kameros išpopuliarėjo tarp mažą interneto greitį turinčių vartotojų. Didėjant interneto spartai, paskutiniaisiais metais vis dažniau gaminamos didesnės (640x480) rezoliucijų kameros.

Siekiant įgyvendinti šį projektą, pirmiausia buvo iškelti šie uždaviniai. Suprojektuoti naująją vartotojo sąsają, nes senoji trečio kurso nebetenkino poreikių, paruošti planą jos veikimui, kad būtų žinoma kokį funkcionalumą naujoji sistema turės palaikyti, įsigyti internetinę kamerą ir ją prijungti prie sistemos, kad programa galėtų interpretuoti iš kameros gaunamą informaciją ir reaguoti atitinkamai į tai ką ji rodo.

## Darbo eiga

### 1. Buvusio sistemos projekto aptarimas ir analizė

Šiame skyriuje bus kalbama apie trečio kurso platformos galimybes, privalumus ir trūkumus. Pradedant darbus pirmiausia teko susidurti su problema dėl jau esamos aplinkos savybių. Esama platforma nebuvo pakankamai rimta, kuri tiko švietimo tikslais dėl žaismumo ir paprastumo vaikams, bet rimtesnės interesų grupės tuo nesusidomėtų. Toliau aptarsiu tos platformos struktūrą ir galimybes ([žiūrėti 1 pav.](#)).



1 pav. Trečio kurso projekto vartotojo sąsaja

#### 1.1. Programos valdymo komponentai

1. *Arduino* susijungimo skydelis – reikalingas, kad programa galėtų rasti ir susijungti su valdikliu.



2. Rėžimų valdymo skydelis, kuriame pasirenkama kokių valdymo modelių sistema bus kontroliuojama šiuo metu:
  - Automatinis valdymas – radijo bangomis valdomos mašinėlės valdymą atlieka programa, sekdamas vartotojo nurodytomis valdymo instrukcijomis.
  - Rankinis valdymas – mašinėlės valdymą atlieka pats vartotojas, valdydamas sukurtos platformos pagalba.
3. Grafinė vartotojo valdymo dalis, kurioje rodoma, kokias valdymo komandas šiuo metu programa perduoda radijo bangomis valdomai mašinėlei realiu laiku.
4. Valdymo komandų rinkimosi forma. Ją sudaro komandinės eilutės, kurios kiekviena turi šiuos komponentus:
  - Kryptys kuriomis norima judinti radijo bangomis valdomą mašinėlę, žalia – važiuojamoji kryptis, raudona – draudžiamoji kryptis.
  - Laiko valdiklis, skirtas nurodyti kelioms sekundėms sekti šią komandinę eilutę.
  - Patvirtinimo, ši komandinė eilutė bus vykdoma, jei bus parinktas ir pradėtas automatinis valdymas.
  - Pašalinimo – sunaikinama netinkama komandinė eilutė.
  - Perkėlimo – esamų komandinių eilučių keitimas vietomis.
5. Aktyvi komandinė eilutė – važiavimas atgal, kurio trukmė 3 sekundės.
6. Išjungta komandinė eilutė – važiavimas atgal, kurio trukmė 5 sekundės.
7. Nepanaudota komandinė eilutė – ji nėra vykdoma.
8. Kartojimo pasirinkimas, kiek kartų kartoti komandinės eilutės komandas.
9. Įvykių ataskaita. Įvykių ataskaitos paskirtis – informuoti vartotoją apie visą svarbią informaciją.

## **1.2. Platformos aptarimas**

Šiame skyriuje bus bendrai aptarti projektuojamos platformos privalumai ir trūkumai. Privalumai:

- Lengvai plečiama.
- Nesunkus programos valdymas, ir valdymo komandų skirtų robotui rašymas.
- Realio laiku galima sekti komandų vykdymą ir kryptis, kuriomis radijo nagomis valdoma mašinėlė judės.
- Galima sukurti daug komandinių eilučių šablonų, nereikia kiekvieną kartą juos pakartotinai aprašyti.
- Galima plėsti arba mažinti komandinių eilučių šablonų skaičių.
- Įvykių ataskaita pateikia ataskaitą apie nuvažiuotus atstumus, kryptis ir laiką.
- Žinant valdiklio kontaktus kuriems paruošta programa valdyti, galima prijungti ir valdyti bet kurią mašinėlę

Trūkumai:

- Norint plėsti valdymo platformą, pradedančiajam reiktų skaityti dokumentaciją.
1. Valdymo skydelis gali daugeliui nepatikti dėl per didelio paprastumo.

## **2. Naujos sistemos aptarimas ir analizė**

Šiame skyriuje bus aptariama naujoji sistema, jos savybės ir valdymas.

### **2.1.Naujos sistemos reikalavimai**

- Galimybė interpretuoti vaizdinės medžiagos informaciją.
- Galimybė vartotojui apsirašyti norimą skaičių robotų, kuriuos valdiklis gali kontroliuoti.
- Galimybė dirbti su keliomis kameromis vienu metu.

- Galimybė kurti, naudoti ir saugoti neribotą skaičių maršrutų.
- Galimybė valdyti kelis robotus vienu metu.

## 2.2. Naujos sistemos projektavimas

*JavaFx* – tai kompanijos oracle produktas, modulis kuris susideda ir programavimo aplinkos, su savo sintakse, struktūra ir yra skirtas kitų aukšto lygio programų kūrimui. *JavaFx* yra populiariausia Java kalbos aplikacijų kūrimo programa pasižyminti dideliu informacijos kiekiu skirtu padėti kuriant programas pagal savo poreikius internete ir lankstumu norint pasiekti norimų rezultatų. *JavaFx* naudoja tą pačią sintaksę kaip ir Java tačiau vietoj SWING karkaso yra naudojama meta kalba *fxml*. Visas funkcionalumas komponentai yra aprašomi *xml* kalba, o rašoma programa su aprašyta aplinka sujungiama naudojant valdiklį kaip tarpininką, tarp komponentų ir programos logikos.

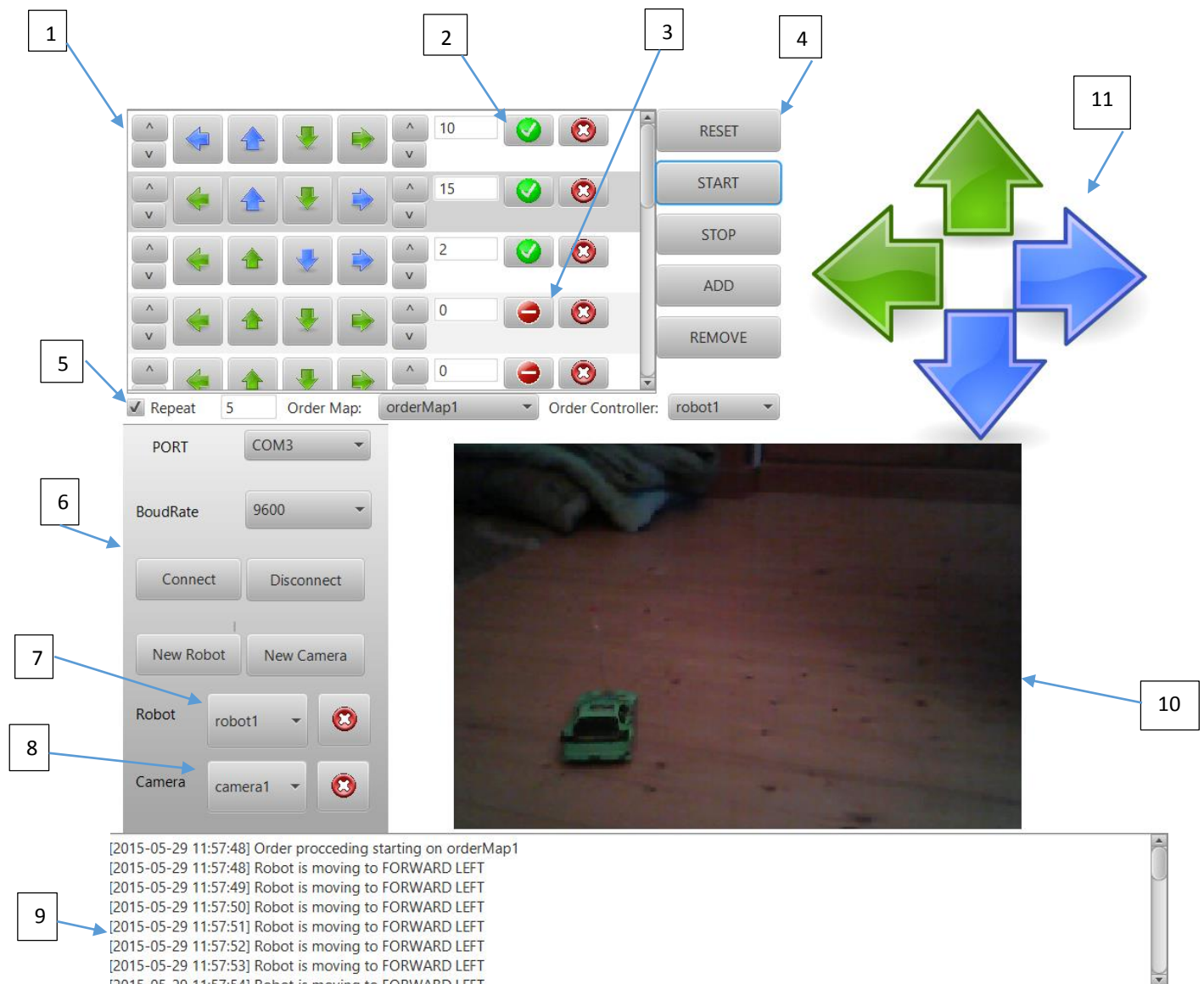
## 2.3. Naujos sistemos projektavimas

Naujoji valdymo sistemoje pirmiausia buvo atnaujinta vartotojo sąsaja, pakeičiant *SWING* grafinę aplinką į *JavaFx*. Naudojant *JavaFx scene builder program*, susikūriau *fxml* failą, kuriame buvo aprašyti visi mano sukurtos aplinkos valdikliai bei komponentai. Sekanti užduotis buvo sukurti grafinį valdiklį visai grafinei aplinkai. Grafinis valdiklis leidžia vartotojui sukurti ryšį tarp grafinės aplinkos ir programinio kodo. Galutinis *fxml* failas susidėjo iš komponentų ir savybių kurios jau buvo naudotos trečiojo kurso baigiamajame darbe, bei su praplėstomis savybėmis, kurių ankstesnis projektas neturėjo. Pirmiausia buvo atnaujintas valdiklio atpažinimo ir paieškos mechanizmas. Naujoji vartotojo sąsaja, randa sąrašą išorinių įrenginių, prijungtų prie kompiuterio ir pateikia prijungtų prietaisų sąrašą vartotojui. Ankstesnė projekto versija, negalėjo to daryti ir vartotojas turėjo, pats įvesti prietaiso identifikavimo sistemoje numerį (prietaiso adresą). Ankstesnė programos versija palaikė tik vieną robotą, tad esama sistema buvo praplėsta taip, kad galėtų joje būti įregistruoti ir dirbti keli robotai vienu metu. Tam realizuoti buvo panaudota agentų klasė.

Agentų klasę sudaro informacija apie robotus, jų statusai ir valdiklio komponentai, kurie atsakingi už jų koordinavimą. Kiekvienas agentas gali gauti valdymo komandų objektą, kuris nusako kokius judesius jis turi atlikti, pvz: penkias sekundes judėti į priekį, tada pasukti ir kitas dešimt sekundžių važiuoti į dešinę.

Siekiant plėsti sistemos galimybes buvo sukurta stebėtojų klasė. Šioje klasėje aprašyta informacija reikalinga darbui su kameromis. Vartotojas gali būti prisijungęs kelias kameras ir stebėti jų informaciją. Sekantis skyrius aptars naujosios vartotojo platformos savybes.([žiūrėti 2 pav.](#)).

## 2.4. Programos valdymo komponentų aptarimas



**2 pav. Trečio kurso projekto vartotojo sąsaja**

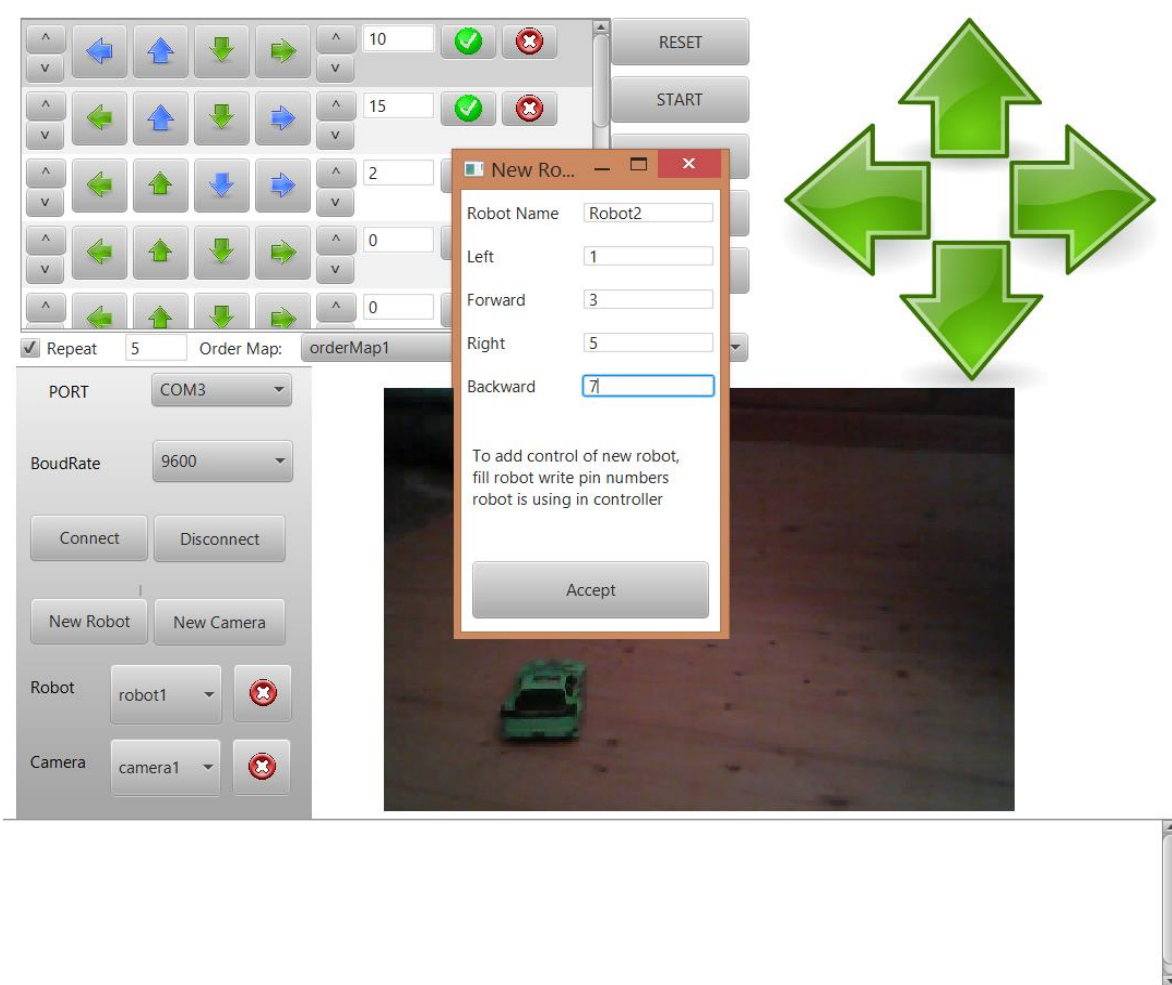
1. Komandų kontrolinis skydelis – šis skydelis rodo pasirinkto komandų sąrašo komandines eilutes. Vartotojas šiose eilutėse pagal savo poreikius pažymi kokiomis komandomis norima užprogramuoti savo roboto seką. Kiekviena eilutė yra valdymų komandų rinkinys, kurį sudaro:

- 1.1. Perkelti aprašytą judesių seką į sąrašo priekį ar atgal per vieną poziciją. Taip galima sukeisti komandines eilutes vietomis, greitai ir paprastai.
- 1.2. Kryptys kuriomis bus judinamas prijungtas robotas, mėlyna – važiuojamoji kryptis į kurią judės robotas, žalia – draudžiamoji kryptis, į kurią robotas nejudės.
- 1.3. Perkėlimo – skirto laiko komandos vykdymui didinimas ir mažinimas per vieną sekundę.
- 1.4. Laiko valdiklis, skirtas nurodyti kelioms sekundėms sekti šią komandinę eilutę.
- 1.5. Patvirtinimo, ar vykdyti šią komandinę eilutę, ar ją praleisti.
- 1.6. Pašalinimo – sunaikinama nereikalinga komandinė eilutė.
2. Varnele pažymėtos komandos dalyvaus valdant robotą. Nurodyta komanda nurodo robotui judėti dešimt sekundžių į priekį ir kairę.
3. Draudžiamuoju ženklu pažymėtos komandų eilutės nebus vykdomos.
4. Komandų kontrolinio skydelio nurodymų valdymas.
  - 4.1. Šiuo metu pasirinkta valdymo komanda bus atstatyta į pradinę būseną.
  - 4.2. Pasirinkta valdymo komanda bus vykdomas programos.
  - 4.3. Sustabdomas valdymo komanda jei šiuo metu ji yra vykdoma.
  - 4.4. Pridedamas papildomas valdymo komandų sąrašas.
  - 4.5. Pašalinamas dabar pasirinktas valdymo komandų sąrašas.
5. Komandų kontrolinio valdymo skydelis.
  - 5.1. Ar kartoti pasirinktą komandų sąrašą.
  - 5.2. Kiek kartų kartoti pasirinktą komandų sąrašą.
  - 5.3. Komandų sąrašas, kuris bus atliekamas ir rodomas vartotojui.
  - 5.4. Robotas kuris atliks nurodytas komandas. Galima nurodyti kelis, tada abu robotai vykdys tas pačias komandas.

6. Valdiklio prijungimo skydelis, skirtas prijungti *Arduino* prie programos ir leisti valdyti per *Arduino* robotus.
7. Robotų valdiklis, skirtas šalinti blogai įvestus ar nebenaudojamus robotus.
8. Kamerų valdiklis, skirtas šalinti blogai įvestas ar nebenaudojamas kameras.
9. Programos funkcijų ataskaita, nurodanti kokius veiksmus koku laiku atlikinėjo programa.
10. Kameros vaizdas realiu laiku, pasirinkus norimą kamerą, rodomas vaizdas kurį ji siunčia jei tik yra.
11. Realio laiku rodomos kryptys, kuriomis robotas juda.

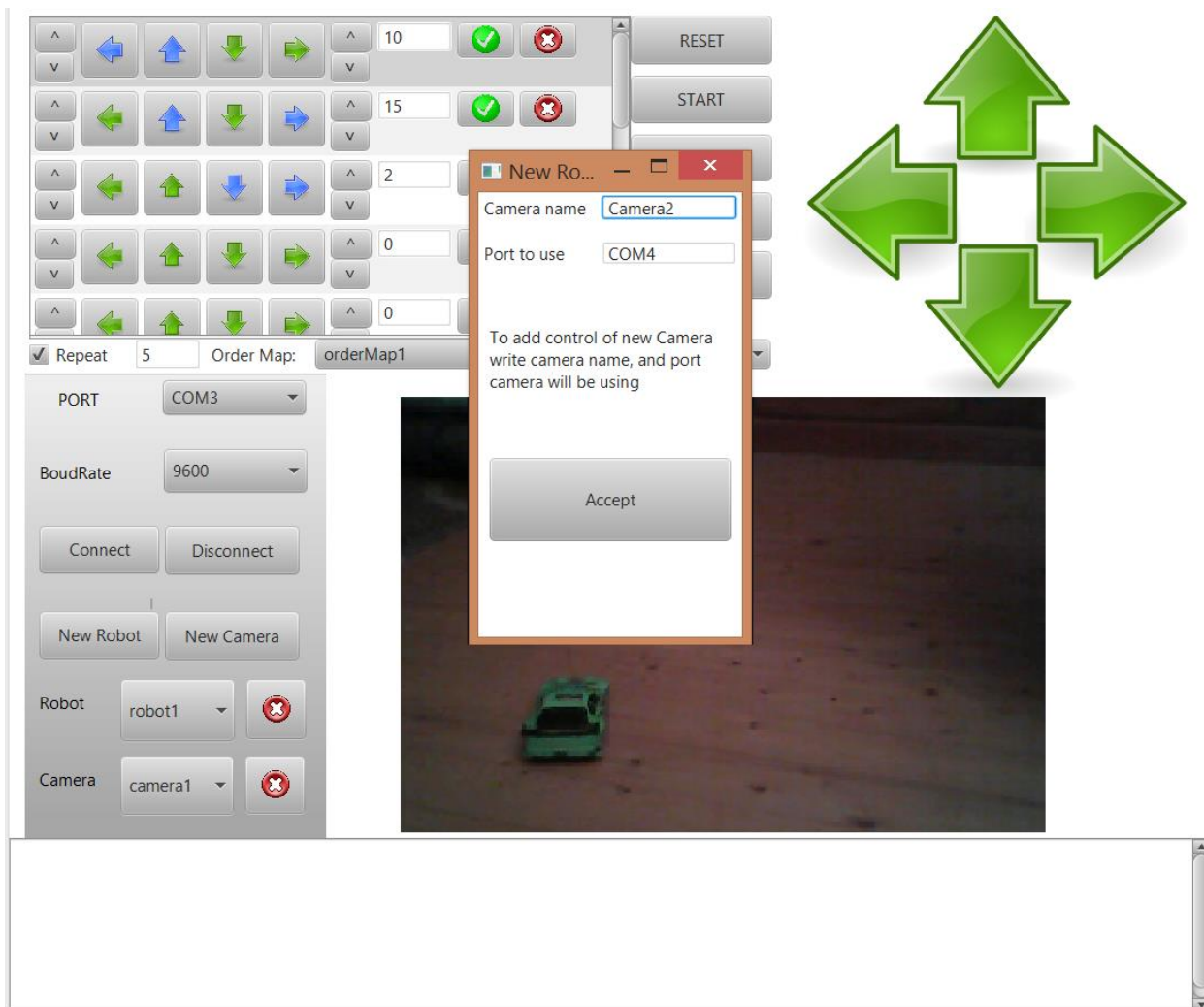
1.1. Grafinė vartotojo valdymo dalis, rodanti realio laiku siunčiamas valdymo kryptis.

1.2. Naujo roboto ir kameros pridėjimas



3 pav. Naujosios valdymo programos naujo roboto pridėjimo forma

Norint pridėti naują robotą prie aktyvių robotų sąrašo, būtina nurodyti valdiklio komponentus kurie jį valdys ([žiūrėti 3 pav.](#)). Lygiai taip pat būtina nurodyti parametrus norint prie sistemos pridėti naują kamerą ([žiūrėti 4 pav.](#))



4 pav. Naujos kameros pridėjimo forma

## 2.5. Valdymo režimai

- Vartotojo valdymas: jei vartotojas atlieka aktyvius veiksmus, valdydamas kompiuterio mygtukus, prioritetas skiriamas šiam režimui ir visa kita veikla sustabdoma.
- Automatinis vartotojo: jei komandų eilutėje yra paspausta pradėti (start) mygtukas prioritetas skiriamas komandų vykdymui.
- Automatinis stebėjimo. Egzistuojant prie sistemos pajungtai kamrai ir jei vartotojas nenurodęs jokių komandų vykdymo, tačiau prašoma vykdyti komandas, naudojamas stebėtojo režimas, kuris seka kameros informacija ir pats pasiunčia robotą į vietą.



- Pasyvus režimas – jei nėra tenkinamas nei vienas iš aukščiau paminėtų sąlygų programa nieko nedaro ir laukia vartotojo pasirinkimo.

## 2.6. Platformos vertinimas

Šiame skyriuje bus bendrai aptarti projektuojamos platformos privalumai ir trūkumai. Privalumai:

- Lengvai plečiama.
- Naujų prietaisų pridėjimas ir valdymas nėra sudėtingas.
- Realio laiku galima sekti komandų vykdymą.
- Galima sukurti daug komandinių eilučių šablonų, nereikia kiekvieną kartą juos pakartotinai aprašyti.
- Galima plėsti arba mažinti komandinių eilučių šablonų skaičių.
- Įvykių ataskaitą pateikia apie nuvažiuotus atstumus, kryptis ir laiką.
- Darbas su keliais robotais.
- Galimybė stebėti kamerų parodymus, ir atitinkamai reaguoti į įvykius.
- Naujoviška ir lengvai plečiama grafinė vartotojo sąsaja.
- Visi nustatymai išsaugomi ir juos galima panaudoti iš naujo paleidus programą.

Trūkumai:

2. Jaunesniems ypač vaikams gali būti pernelyg sudėtinga suprasti sistemos veikimą iš pat pradžių.
3. Norint plėsti programos logiką reiktų mokėti *Java* ir *JavaFx*.

## 2.7. Kameros valdymas

Pati didžiausia kliūtis šiame projekte, buvo kameros valdymas ir darbas su ja *Java* aplinkoje. Internetu rasti pavyzdžiai padėjo aiškinantis kaip galima būtų realizuoti vieną ar kitą savybę, kurių projektui gali prireikti. Pirmiausia buvo ieškomi jau esami sprendimai, kurie padėtų padėti išgauti



vaizdo kameros informaciją ir ją interpretuoti. Daugiausiai atsiliepimų radau apie esamą *JAVA* biblioteką kuri yra skirtą darbui su vaizdo kameromis. Jos privalumai:

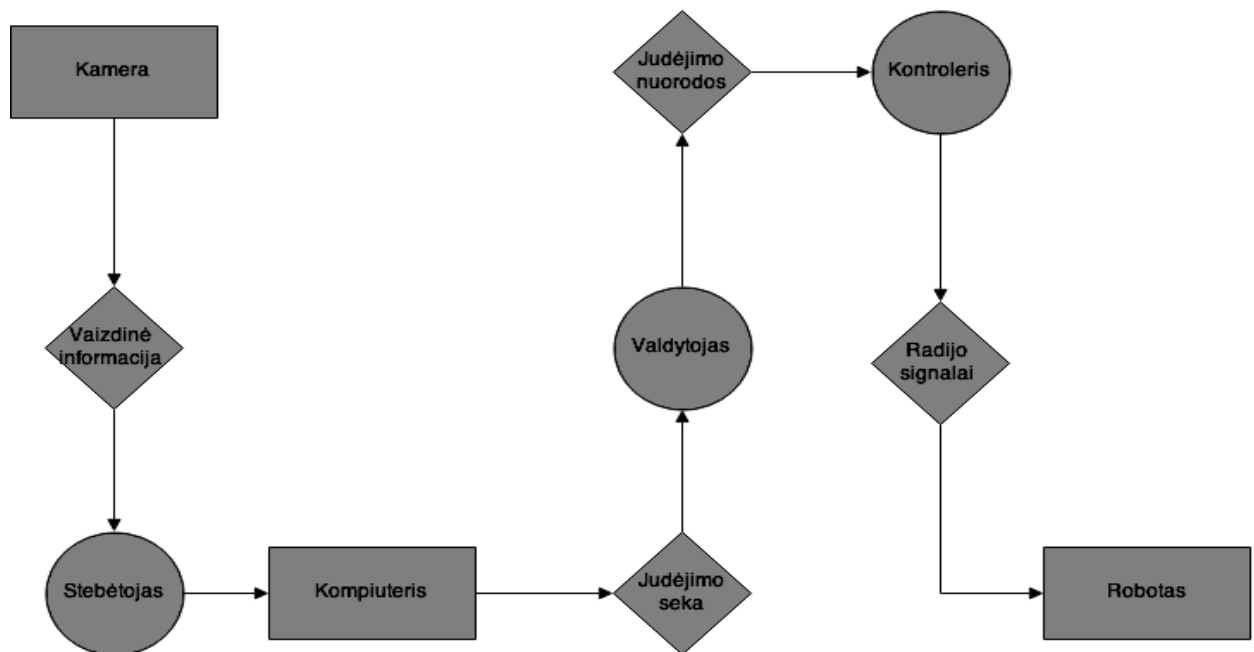
1. Paprastumas, naudojant vieną giją ir neblokuojanti išorinių gijų aplinka.
2. Nereikalinga jokia papildoma programinė įranga.
3. Palaikoma skirtingose platformose (*Windows*, *Linux*, *Mac OS*, t.t.) ir skirtingose kompiuterių architektūrose (*32-bit*, *64-bit*, *ARM*).
4. Galima gauti kamerų vaizdus iš skirtingų šaltinių tiek iš įmontuotų web kamerų, tiek iš prijungtų per *USB* vaizdo kamerų.
5. Galima gauti vaizdus iš tinklinių kamerų formatais *MJPEG* ar *JPEG*.
6. Iš karto paruoštas judesių atpažinimo modulis.
7. Visi reikalingi projekto failai yra viešai pasiekiami per *MAVEN*.
8. Paruošti *SWING* komponentai atvaizduoti informacijai iš kamerų realiu laiku.
9. Palaikoma daugybė skirtingų vaizdo apdorojimo įrankių skirtų darbui su vaizdine informacija.

## **2.8. Radijo siųstuvo valdymas**

*Arduino* plokštės valdymui šiame projekte yra naudojama *Ardulink* biblioteka. Ši biblioteka yra atvirojo kodo biblioteka skirta komunikuoti pagal paruoštas komandas tarp *Java* ir *Arduino* sutaupanti laiko rašant paties *Arduino* valdiklį programai.

## **3. Valdymo schema**

Šiame skyriuje bus aptariamas sistemos valdymas pažingsniui, aptariant kiekvieno komponento rolę ir rezultatus, kas iš jos tikimasi. ([žiūrėti 5 pav](#)).



5 pav. Naujosios sistemos valdymo schema

1. Kamera fiziškai filmuoja esamą aplinką ir perduoda matomą informacijos srautą stebėtojo klasei.
2. Stebėtojas fiksuoja ir interpretuoja gaunamus duomenis, užfiksavus neatitikimą informuojama sistema, kad reikia nusiųsti robotą.
3. Kompiuteris paruošia komandų sąrašą, kuriomis robotas turės vadovautis, kad pasiektų galutinį tikslą.
4. Valdytojas konvertuoja komandų sąrašą ir perduoda komandas į *Arduino*, kuriuos valdiklio komponentus jis turi valdyti.
5. *Arduino* gauna komandų seką ir kaip valdiklis perduoda viską į valdymo pultelį
6. Pultelis sugeneruoja radijo signalus ir perduoda juos į mašinėlę.
7. Mašinėlė gavusi radijo signalus juda nurodytomis kryptimis.

## **Rezultatai ir išvados**

Išvados: Atliekant šį projektą buvo susidurta su keliomis pagrindinėmis problemomis. Pati didžiausia buvo vaizdinės informacijos apdorojimas. Buvo pasirinktas paprastas metodas, ieškoti vaizdinėje medžiagoje šviesos šaltinio. Užfiksavus pasikeitimus, ieškoti pakankamai stipraus šviesos šaltinio ir užfiksavus tokį, nusiųsti ten robotą. Deja ši metodika nėra labai patikima ir esant keliems ar labai dideliame šviesos šaltiniui, arba esant labai šviesioje patalpoje, galima padaryti klaidų skaičiavimuose ir paieškose. Norint pasiekti geresnių rezultatų, patartina naudoti didelės raiškos kameras, nes paprastos kameros naudojamos kompiuteriuose gali nepatenkinti visų poreikių.

Rezultatai: Sukurta nauja programos versija, praplečianti trečiojo kurso projekto galimybes ir tikslus. Naujoji versija, palaiko galimybę pagal poreikius naudoti kameras, bei naudoti robotus su atitinkamais davikliais tiek metalinių objektų radimui teritorijoje, tiek ir fiksuoti pokyčius erdvėje ir reaguoti aprašytu būdu. Sistema palaiko galimybę sekti ir dirbti su keliomis kameromis ir robotais. Sistema taip pat veikia greičiau, tapo paprastesnė ir patrauklesnė vartotojui, turi daugiau funkcionalumo. Galima valdyti kelis robotus su komandomis ar iš vartotojo sąsajos vienu metu. Deja, norint plėsti funkcionalumą reikia įsigyti ir prijungti papildomus daviklius, mažai žinančiam žmogui tai padaryti gali būti sunku.

## **Šaltiniai**

[WAM+11] John-David Warren, Josh Adams, Harald Molle. Arduino Robotics. Apress 2011.

[WIL12] Don Wilcher. Electronics with Arduino. Apress 2012.

[MCR+13] Michael McRoberts. Beginning Arduino, 2nd Edition. Apress 2013.

[ARD] Arduino playground and interface interaction examples.

Prieiga per internetą: <http://playground.arduino.cc/Interfacing/Java>

[WCS] WebCam library implementation.

Prieiga per internetą: <http://webcam-capture.sarxos.pl>

[JAV8] Java ir JavaFx informacija ir dokumentacija

Prieiga per internetą: <http://docs.oracle.com/javase/8/javase-clienttechnologies.htm>

## Sąvoku ir santrupų apibrėžimas

Jutiklis – jutikliai yra aparatūros komponentai, galintys teikti kompiuteriui informaciją apie kompiuterio vietą, aplinką ir kitą svarbią informaciją susijusia su aplinkos veiksniais. Iš jutiklių informaciją gali pasiekti kompiuterio programos, per paruoštą valdiklį arba kitą informacijos perdavimo sąsają. Panaudota informacija būna naudojama atlikti įvertinti arba informuoti apie aplinkos veiksmus kurie susidarė ar atsirado jutikliams pradėjus dirbti, taip palengvinadami kasdienias užduotis.

Jutikliai yra dviejų tipų:

- Kompiuteryje integruoti jutikliai
- Jutikliai, kurie prie kompiuterio prijungti laidiniu arba balaidžiu ryšiu

Pikseliai - vienas iš daugybės taškų ar vienos spalvos laukelių, kurie, kartu kaip dvimatė matrica, sudaro matomąjį vaizdą arba visą ekrano turinį. Grafinėje aplinkoje pikselis yra mažiausias adresuojamas elementas, nusakantis tokias charakteristikas kaip spalva ir jos šviesumas. Pikselio adresą atitinka jo fizinės koordinatės erdvėje.

Klasė - aprašas, nusakantis tam tikro tipo objektų duomenų struktūrą, jos savybes, būvį ir manipuliavimo ja taisykles (elgesį).

C ir C++ - bendros paskirties programavimo kalbos, c++ viena populiariausių iš egzistuojančių programavimui skirtų kalbų. C++ yra C kalbos išplėsta versija, apibrėžianti bibliotekų rinkinį bei specialų procesorių, suteikiantį C kalbai objektinio programavimo galimybes, naudoti klases.

SD atminties kortelės - yra plačiai paplitęs atminties kortelių tipas, leidžiantis integruoti papildomas funkcijas, susijusias su duomenų saugumu. Jos pasižymi dideliu duomenų perdavimo greičiu, yra talpios, pigios ir geros kokybės.

USB jungtis - (Universal Serial Bus) tai universalioji jungtis, kuri naudojama kompiuteriuose kaip tarpininkas tarp prietaisų ir paties kompiuterio. Per šią jungtį galima prijungti įvairius išorinius įrenginius, juos valdyti ir gauti iš jų informaciją.

Xml – (angliškai Extensible Markup Language) yra W3C rekomenduojama bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba. Vienos svarbiausių savybių kuriomis

pasižymi ši kalba – yra lengvai skaitoma vartotojui bei skirta užtikrinti lengvesnį duomenų keitimąsi tarp skirtingo tipo sistemų, dažniausiai sujungtų per internetą. Pagrindinis XML kalbos vienetas yra elementas. Elementas visada turi vardą ir, be jo, gali turėti:

- norimą skaičių atributų. Atributas turi savo vardą bei reikšmę.
- kitus (dukterinius) šio elemento viduje esančius elementus.
- su elementu susijusį tekstą.

SWING – aplikacijų programavimo sąsaja skirta grafinėi vartotojo sąsajai sukurti. SWING numato įvairius galimus vartotojo sąsajos komponentus, tokius kaip mygtukai, teksto laukai, lentelės, nuotraukos, medžiai ir kitus. SWING naudojama JAVA programose. Vienas iš labiausiai atskirtinų bruožų apie SWING – nepriklausomai nuo naudojamos sistemos, komponentų išvaizda išlieka tokia pati.

GUI – (angliškai: graphical user interface) – yra grafinė sąsaja skirta bendravimui tarp vartotojo (programos naudotojo) ir interpretuotojo (kompiuterio programos). Komandoms parinkti, paleisti programoms, failų ir katalogų vardams pakeisti, parinktim parametrams stebėti ir parinkti, taip pat kitiems veiksams atlikti naudojami, monitoriuje rodomi dialogo langai, meniu punktai ir mygtukai, kuriais manipuluojama įvesties įrenginiais, tokiais kaip pelė ir klaviatūra.

JPEG – JPG (arba JPEG, angliškai: Joint Photographic Experts Group) – vaizdų išsaugojimo formatas (ir jų suspaudimo algoritmas), skirtas sukurti objektus (nuotraukas) iš nuskaitytos informacijos, paimtos iš video jutiklių. JPG labai efektyviai suspaudžia fotografinius vaizdus. Nors dalis informacijos esant didesniai suspaudimo laipsniui prarandama, formato ypatybės tokios, jog šie praradimai yra labai sunkiai pastebimi. Prarasta informacija daug labiau matoma įvairiose schemose bei diagramose, kur daug plonų linijų bei vienaspalvių plotų.

MJPEG – Video spaudimo algoritmas skirtas dideliame JPEG paveiksliukų skaičiui suspausti į vientisą vaizdo įrašą.

MAVEN – priemonė, skirta JAVA projektams valdyti bei diegimo procesams automatizuoti.

Servo variklis – išorinis prietaisas įrenginys ar motoras, palaikantis galimybes su dideliu tikslumu sukurti pagal nurodytas komandas. Servo motorai naudojami robotikoje kaip varikliai, skirti robotų judėjimui ar kaip sudedamosios dalys atliekančios svarbius valdymo veiksmus.

Projekto nuoroda GitHub

<https://github.com/AzureDragon/Bakalaurinis>