VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

**PROGRAMŲ SISTEMŲ KATEDRA**

**Pigus robotai apsaugos sistemoms**

**Bakalauro baigiamasis darbas**

Atliko: Osvaldas Žiukas (parašas)

Darbo vadovas: Irus Grinis (parašas)

Darbo recenzentas: Petras Petraitis (parašas)

Vilnius, *2015*

# Santrauka

Šio darbo tikslas yra sukurti pigų robotą, kuris galėtu 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 web kamerą užfiksuoti nurodytoj zonoje judėjimą ir ten nusiųsti robotą žvalgybai ar apsaugai. 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 saugos sistemų efektyvumą, bei padidinti naudą, kurią gali gauti švietimo įstaigos naudodamos pigų sprendimą vaikų ugdymui ir lavinimui mokyklose.

# Summary

Goal of this project is to create cheap robot which coule be controller by user or program and would do tasks assigned to it in both security and teaching domain. Complete project consist of robot, controller, web camera and program which is designed to take input, make neccesary calculations and controll robot respectively. To reduce price of a robot and all parts for this project radio controlled car was used as a robot, and as controller was used Arduino, which controller remote controller of car. Using third part library for interpretating camera information main goal of this project is to notice movement in guarded area and send robot as scout and guard zone. This project supports multiple cameras and robots at same time. For technical part of project was used simple components and parts which dont require deep understanding in that domain and most important work was on creating program for user which would let either controll robot or set guarding mode on. Final goal is to increase efficiency of security systems and give extra motivation and guidence in children teaching in schools for cheap price.

**Turinys**

[Santrauka 2](#_Toc420320758)

[Summary 3](#_Toc420320759)

[Įvadas 5](#_Toc420320760)

[Darbo eiga 8](#_Toc420320761)

[1. Esamos sistemos aptarimas 8](#_Toc420320762)

[1.1. Programos valdymo komponetų paaiškinimai 8](#_Toc420320763)

[1.2. Platformos aptarimas 9](#_Toc420320764)

[2. Naujos sistemos aptarimas 10](#_Toc420320765)

[2.1. Programos valdymo komponetų paaiškinimai 11](#_Toc420320766)

[2.2. Valdymo rėžimai 13](#_Toc420320767)

[2.3. Platformos aptarimas 13](#_Toc420320768)

[2.4. Kameros valdymas 14](#_Toc420320769)

[2.5. Valdymo schema 15](#_Toc420320770)

[3. Roboto valdymas 16](#_Toc420320771)

[3.1. Roboto vietos nustatymas 16](#_Toc420320772)

[3.2. Atstumo skaičiavimas 16](#_Toc420320773)

[3.3. Tolimesnis roboto valdymas 17](#_Toc420320774)

[4. Vizija 17](#_Toc420320775)

[Rezultatai ir išvados 18](#_Toc420320776)

[Šaltiniai 19](#_Toc420320777)

[Sąvoku apibrėžimas 20](#_Toc420320778)

[Priedai 22](#_Toc420320779)

# Įvadas

**Robotas** – mechaninis aparatas, galintis atlikti užprogramuotas fizines užduotis. Robotas gali būti valdomas tiesiogiai žmogaus arba veikti valdomas užprogramuoto kompiuterio. Robotai gali atlikti užduotis, kurios yra pavojingos žmogui, dirbti žmonėms kenksmingoje aplinkoje. Jie taip pat gali atlikti monotoniškas, dažnai pasikartojančias užduotis, tokiu būdu pigiai pakeisdami žmogaus rankų darbą.

Robotai gali būti pritaikyti daugybėje sričių: pradedant vaikų priežiura, švietimu, ligonių slauga, 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, o sekantys kuriami mechaniniu principu, siekiant išnaudoti paprastus ir pigius sprendimus pasiekti norimiems rezultatams.

Tačiau nors ir esant didžiulei robotų pasiūlai vis dar susiduriama su brangiais siūlomais sprendimais, kurie būna neprieinami, neivestuojant 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 maža kainą. Mano darbo esmė yra pasiūlyti būtent tokį sprendimą suinteresuotai auditorijai.

Mano pradinė idėja buvo praplėsti funkcionalumą jau esamo roboto projekto etape, jo naudą švietimo įstaigose, panaudojant web kameras ir atpažįstant objektus pasitaikančius kelyje, tokias kaip kliūtis, galbūt plečiant daugiau ir žmones, tačiau nesinorėjo apsiriboti vien švietimu ir mokamąja sritimi. Karta būnant su šeima išgirdau apie įvykį, kuomet buvo apiplėštas sandėlis ir iki tol, kol atvažiavo saugos tarnyba įsibrovėliai jau buvo pasitraukę. Ir nors ten buvo kameros, kurios turėjo viską užfiksuoti, vien tik iš jų įrašų daug naudos nebuvo. Tada ir kilo mintis - o jei galima būtų padaryti atbaidimo priemonę papildomai prie apsaugos kamerų, kuri ne tik atbaidytų, bet galėtų galbūt net sulaikyti įsibrovėlį ar jį pagauti iki tol kol pasirodys saugos tarnybos ar savininkas. Jau esamos saugumo sistemoje kameros galėtų būti akys informuojančios agentą ir jį iškviečiančios į ivykio vietą. Taip kilo mintis panaudojant kameras ir pigų robotą sukurti prototipą įdėjos apie saugos agentą, praplečiantį esamas saugumo sistemas, ir pasitarnaujantį kaip papildoma apsaugos sistemos dalis.

Projektas turėjo būti atliekamas namų sąlygomis, todėl norint jį pradėti ir įgyvendinti, teko susipažinti ir išsirinkti mikrokontrolerį roboto valdymui. Kas tai yra? Mikrokontroleris (MK) – ATMega, PIC ir dar panašios mikroschemos – vienakristalės mikro skaičiavimo mašinos. [Kontroleris](http://parduotuve.inforeg.lt/index.php?route=product/category&path=59)– plokštė, kurioje įlituota MK mikroschema.

    Kontroleris, kaip jau ir sako pavadinimas, elektroninis įtaisas, kuris kažką kontroliuoja, tai yra reaguoja į pasikeitusius aplinkos parametrus. Aplinkos parametrai nuskaitomi specialiais davikliais, dar kitaip vadinamais [sensori](http://parduotuve.inforeg.lt/index.php?route=product/category&path=60)ais. Valdymo įtaisai ([motorika](http://parduotuve.inforeg.lt/index.php?route=product/category&path=61), elektroniniai moduliniai [komponentai](http://parduotuve.inforeg.lt/index.php?route=product/category&path=62))  – viskas ką galima valdyti paduodant elektros impulsus. Įvairios rėlės, šviesos šaltiniai, optoelektronika ar servo varikliai. Pavyzdžiui kondicionierius, ten stovi tokia "protinga" plokštė – kontroleris, kuri įjungia/išjungia oro šildymą / šaldymą, drėgmės reguliatorių tada, kai to reikia.

   Kondicionierius, MP3 grotuvas, GPS navigatorius, net ir mikrobangų krosnelė ar šaldytuvas – visa tai grubiai tariant – specializuoti kontroleriai. Aiškinantis apie mikrokontorlerius teko susidurti su didele jų pasiūla, tačiau man reikėjo paprasto pigaus, lengvai programuojamo ir su parengtomis pamokomis ir informacija lengvai prieinama internete. Iš visų variantų nusprendžiau pasirinkti Arduino.

Arduino – tai vienas iš kontrolerių, kurį galima priversti kažką daryti. Nuo stalinio laikrodžio su žadintuvu, šviesos efektų iki roboto. Arba šiaip pasižaisti surenkant išardant savo sugalvotas smulkmenas. Radiokonstruktorius. Prie kontrolerio plokštės galima jungti begalę periferinių įrenginių – daviklius (temperatūros, apšvietimo, slėgio, pagreičio ir t.t.), šviesos diodus, skystų kristalų (LCD) ekranėlius, servo variklius. Parašius programą priversti vienaip ar kitaip reaguoti į pasikeitimus  ar vykdyti kažkokius veiksmus. 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 kontroliuoti fizinį pasaulį. Arduino naudoja C ir C++ programavimo kalbos “mišinį”. Išmokti šią programavimo kalbą nėra sudėtinga. Dar vienas geras Arduino privalumas – tai, kad jam yra sukurti papildomi [skydeliai](http://www.savaitgalioprojektai.lt/arduino-skydeliai-shields/) (angl.k Shield), kurie suteikia galimybę Arduino prijungti prie interneto, WiFi tinklo, Bluetooth ryšio. Taip pat gali suteikti galimybę naudoti  SD atminties korteles ir dar daug visokių kitų galimybių. Su Arduino žaisti galima kaip su LEGO – dėlioji detales ir iš karto matai rezultatą. Na, bet aišku privalai laikytis tam tikrų taisyklių, kad nesudegintum mikrokontrolerio. Arduino UNO ypatingas tuom, kad jis nėra toks brangus kaip Arduino MEGA ar kita Arduino šeimos platforma, taip pat  jame yra galimybė pakeisti pagrindinę mikroschemą (mikrokontrolerį). Baigus rinktis kontrolerį, man reikėjo pasirinkti kas bus mano robotas. Buvo keli variantai: pačiam susiprojektuoti ir susirinkti, nusipirkti jau surinkta, ar ieškoti alternatyvų. Taip kilo mintis panaudoti kaip modelį patį paprasčiausią variant radijo bangomis valdoma mašinėlę, kurių pilna pardduotuvėse.

Radijo bangomis valdoma mašinėlė – tai prietaisas kurį galima įvardinti pačiu papraščiausiu robotu. Jį sudaro motoras ar motorai, ratai, platforma, gaviklis, ir kontroleris. Iš kontrolerio(pultelio) siunčiamos radijo bangos, per gaviklį lengvai ir paprastai 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](http://lt.wikipedia.org/wiki/Kompiuteris) per [USB](http://lt.wikipedia.org/wiki/USB) jungtį ir [interneto](http://lt.wikipedia.org/wiki/Internetas) pagalba leidžianti realiu laiku perteikti vaizdą. Paprasčiausios tokios kameros naudoja [CMOS](http://lt.wikipedia.org/w/index.php?title=CMOS&action=edit&redlink=1) tipo [matricą](http://lt.wikipedia.org/wiki/Matrica). 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.

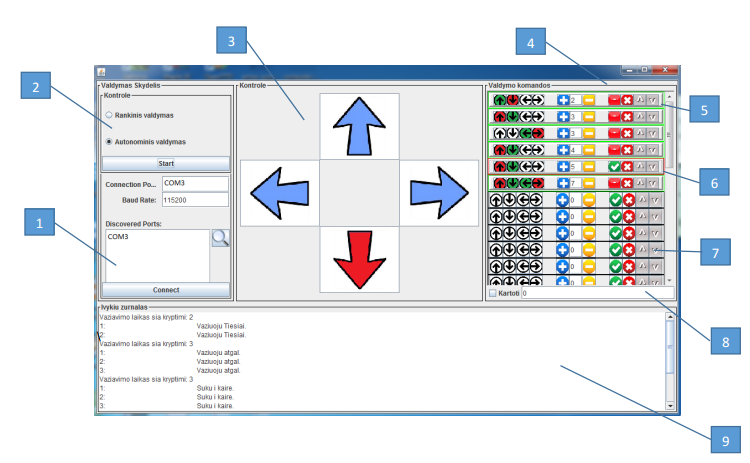
Kameros plačiai naudojamos [ICQ](http://lt.wikipedia.org/wiki/ICQ), [Skype](http://lt.wikipedia.org/wiki/Skype), [MSN](http://lt.wikipedia.org/wiki/MSN) bei kitose pokalbių programose, bei naudojamos kaip sudedamosios dalys vaizdo apdorojimo srityje. Siekaint naudoti platfforma, kuri veiktų ant skirtingų sistemų, pasirinkau Java su jos JavaFx papildiniu.

JavaFx – tai oracle produktas, modulis kuris susideda ir programinės aplinkos, savo sintaksės, struktūros ir yra skirtas kitų aukšto lygio programų kūrimui. JavaFx yra populiariausia Java kalbos aplikacijų kūrimo programa pasižyminti didekiu informacijos kiekiu internete ir lankstumu. JavaFx naudoja tą pačią sintaksę kaip ir Java tačiau vietoj SWING karkaso yra naudojama meta kalba xml. Visas funkcionalumas komponentai yra aprašomi xml kalba, o rašoma programa su aprašyta aplinka sujungiama naudojant kontrolerį kaip tarpininką, tarp komponentų ir programos logikos.

# Darbo eiga

## Esamos sistemos aptarimas

Pradedant darbus pirmiausia teko susidurti su problema dėl jau esamos aplinkos savybių. Esama platforma, nebuvo pakankamai rimta, netgi sakyčiau vaikiška, kuri tiko švietimo tikslais dėl žaismumo ir paprastumo vaikams, bet rimtesnės interesų grupės tuo nesusidomėtų. Toliau aptarsiu esamos platformos struktūrą ir galimybes (žiūrėti 1. paveiksliuką).



Paveikslėlis 1

### Programos valdymo komponetų paaiškinimai

1. Arduino susijungimo skydelis – reikalingas, kad programa galėtų tinkamai komunikuoti su prietaisu.
2. Rėžimų valdymo skydelis, nustatantis mašinos valdymo modelį:

* Automatinį radijo bangomis valdomos mašinėlės valdymą atlieka programa, sekdama vartotojo nurodytas instrukcijas.
* Rankinį atlieka pats vartotojas, valdydamas radijo bangomis valdomą mašinėlę sukurtos platformos pagalba.

1. Grafinė vartotojo valdymo dalis, rodanti realiu laiku siunčiamas valdymo kryptis.
2. Valdymo komandų rinkimosi forma. Ją sudaro komandinės eilutės, kurios viduje yra šios funkcijos:

* 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, ar vykdyti šią komandinę eilutę, ar ją ignoruoti ir nieko nedaryti.
* Pašalinimo – sunaikinama netinkama komandinė eilutė.
* Perkėlimo – esamų komandinių eilučių keitimas vietomis.

1. Aktyvi komandinė eilutė, važiavimas atgal, kurio trukmė 3 sekundės.
2. Išjungta komandinė eilutė, važiavimas atgal, kurio trukmė 5 sekundės.
3. Neapibrėžta komandinė eilutė, ji nėra vykdoma.
4. Kartojimo pasirinkimas, kiek kartų kartoti komandinės eilutės komandas.
5. Įvykių žurnalas. Žurnalo paskirtis – informuoti vartotoją apie visą svarbią informaciją.

### Platformos aptarimas

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

* Lengvai plečiama.
* Lengvai suprantamas funkcionalumas.
* Realiu 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 žurnalas pateikia ataskaitą apie nuvažiuotus atstumus, kryptis ir laiką.

Trūkumai:

* Pradinė versija neturi papildomo funkcionalumo: nežino kaip piešti liniją arba kaip pakelti/nuleisti komponentą (pasirinktas pigesnis modelis to neįtraukia).
* Norint plėsti platformą, pradedančiajam reiktų skaityti dokumentaciją.
* Valdymo panelė gali daugeliui nepatikti dėl per didelio paprastumo.

## Naujos sistemos aptarimas

Naujam funkcionalumui pirmiausia nusprendžiau atnaujinti vartotojo sąsają, pakeičiant naudota SWING karkasą į javaFx. Naudojant JavaFx scene builder program, susikūriau xml failą, kuris apibrėžė mano GUI išvaizdą ir savybes. Šis xml karkasas susidėjo iš jau esamo funkcionalumo, tad apsvarstęs galimybęs karkasą praplėčiau įdėdamas tai, ko dar nebuvo. Pirmiausia atnaujinau valdiklio paieškos mechanizmą, padarydamas kad Arduino valdiklis būtų randamas savaime, ir vartotojas turėtų tik pasirinkti jį, o ne ieškoti ar spėlioti jo indikaciją sistemoje. Ankstesnė programos versija palaikė tik vieną robotą, tad esamą sistemą aš praplėčiiau pridėdamas agentus.

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

Galimybės sistemoje turėti kelis agentus tapo reikalinga, kai nusprendžiau realizuoti atitinkamą stebėtojų kategorija, kurie gali stebėti skirtingus informacijos srautus iš skirtingų kamerų esančių tame pačiame pastate, taip padidinant visos sistemos efektyvumą. Jei būtų užfiksuota įtartina veikla keliose skirtingose vietose, programa galėtų į jas nusiųsti skirtingus robotus. Tolesnis skyrius aptars realizuotos platformos valdymą.

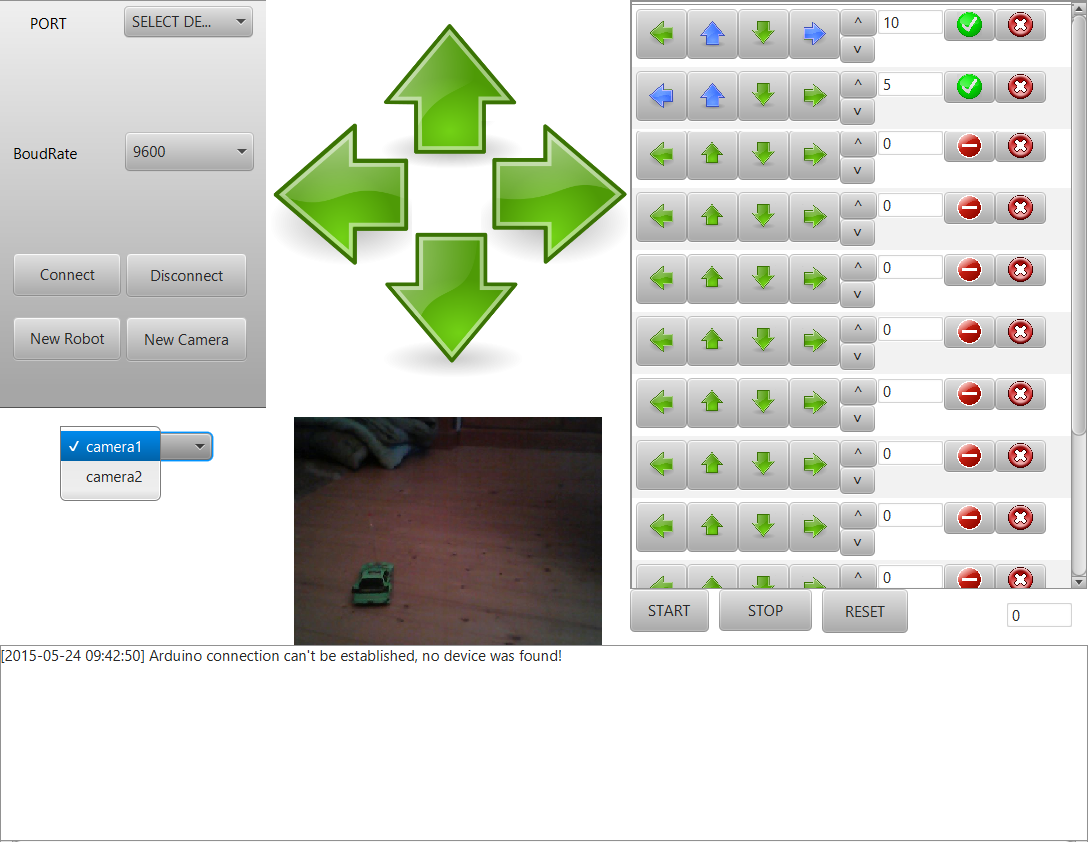
(žiūrėti 2. paveiksliuką).

4

3

2

1



5

6

7

8

Paveikslėlis 2

### Programos valdymo komponetų paaiškinimai

1. Arduino susijungimo skydelis – reikalingas, kad programa galėtų tinkamai komunikuoti su prietaisu, taip pat papildomas valdymas siekiant pridėti naują robotą ir kamerą prie sistemos.
2. Grafinė vartotojo valdymo dalis, rodanti realiu laiku siunčiamas valdymo kryptis.
3. Valdymo komandų rinkimosi forma. Ją sudaro komandinės eilutės, kurios viduje yra šios funkcijos:

* Kryptys kuriomis norima judinti radijo bangomis valdomą mašinėlę, žalia – važiuojamoji kryptis, raudona – draudžiamoji kryptis.
* Perkėlimo – esamų komandinių eilučių keitimas vietomis.
* Laiko valdiklis, skirtas nurodyti kelioms sekundėms sekti šią komandinę eilutę.
* Patvirtinimo, ar vykdyti šią komandinę eilutę, ar ją ignoruoti ir nieko nedaryti.
* Pašalinimo – sunaikinama netinkama komandinė eilutė.

1. Aktyvi komandinė eilutė, važiavimas tiesiai ir į dešinę, kurios trukmė 10 sekundžių.
2. Automatinio rėžimo valdymo panelė. Kol programa valdo mašinėlę, vartotojas negali jos valdyti.

* Start – panaudojus šį mygtuką pradedamos vygdyti vartotojo pasirinktos komandos nurodytas komandas
* Stop – panaudojus šį mygtuką nutraukiamas mašinos automatinis valdyas.
* Reset – ištrina ir atstato į pradines valdymo komandas
* Paskutinis laukelis, jei yra didesnis už 0, nurodo kiek kartų reikia kartoti ciklą.

1. Kameros valdiklis.
2. Nuo pasirinktos kameros kaip srautas vaizduojama ką kamera rodo.
3. Įvykių žurnalas. Žurnalo paskirtis – informuoti vartotoją apie visą svarbią informaciją, tiek valdymo kryptis tiek aplikacijos valdymo būsenas.

### Valdymo rėžimai

* Vartotojo valdymas: Jei vartotojas atlieka aktyvius veiksmus, valdydamas kompiuterio mygtukus, prioritetas skiriamas šiam rėžimui ir visa kita veikla sustabdoma.
* Automatinis vartotojo: Jei komandų eilutėje yra paspausta start prioritetas skiriamas komandų vygdymui.
* Automatinis stebėjimo. Egzistuojant prie sistemos pajungtai kamerai ir jei vartotojas nenurodęs tusčių komandų vygdymą – nėra jokių komandų, tačiau prašoma jas vygdyti, naudojamas stebėtojo rėžimas, kuris seka kameros informacija ir pats pasiunčia robotą į vietą.
* Pasyvus rėžimas – jei nėra tenkinamas nei vienas iš aukščiau paminėtų sąlygu programa nieko nedaro ir laukia vartotojo pasirinkimo.

### Platformos aptarimas

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

* Lengvai plečiama.
* Lengvai suprantamas funkcionalumas.
* Realiu 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 žurnalas pateikia ataskaitą apie nuvažiuotus atstumus, kryptis ir laiką.
* Atsiranda papildomas funkcionalumas ir skirtingi vartojimo rėžimai.
* Darbas su keliais robotais.
* Galimybė stebėti kamerų parodymus, ir atitinkamai reaguoti į įvykius.
* Naujoviška ir lengvai plečiama grafinė vartotojo sąsaja.
* Visi nustatymai išsaugomi nustatymų faile.

Trūkumai:

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

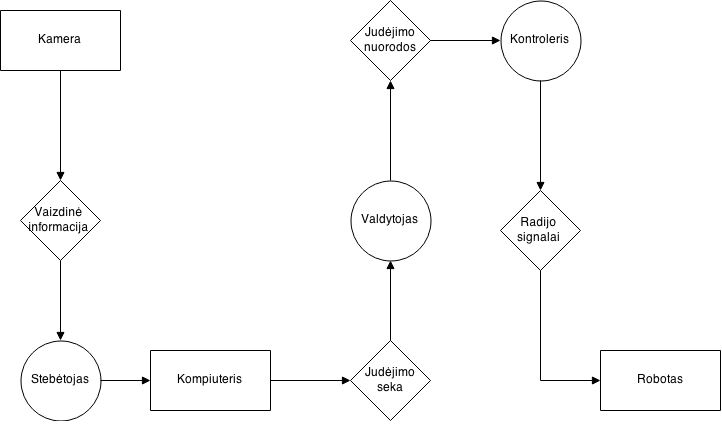
### Kameros valdymas

Pradėjus nagrinėti temą, pirmiausia pradėjau domėtis apie jos įgyvendinimo galimybes. Internete radau pavyzdžių aiškinančių kaip galima būtų realizuoti vieną ar kitą savybę kurių mano projektui gali reikti. Pirmiausia pradėjau ieškoti jau esamų sprendimų, kurie padėtų išgauti web kameros informaciją ir ją interpretuoti. Daugiausiai atsiliepimų radau apie esamą JAVA biblioteką ir api skirtą darbui su web kameromis “Webcam Capture”. Jos privalumai:

1. Paprastuma, naudojant vieną giją ą ir neblokuojanti išoriniu gijų aplinka.
2. Nereikalinga jokia papildoma programinė įranga
3. Palaikoma skirtingose platformose (Windows, Linux, Mac OS, etc) ir skirtingose kompiuterio architektūrose (32-bit, 64-bit, ARM),
4. Gali gauti kamerų vaizdus iš skirtingų šaltinių tiek iš imontuotu web kamerų tiek iš prijungtų per USB.
5. Galima gauti vaizdus iš IP ar tinklinių kamerų formatuose kaip (as MJPEG or JPEG),
6. Iškartu paruoštas judesių atpažinimo modulis
7. Visi reikalingi projekto failai yra viešai pasiekiami per MAVEN
8. Galimybė rodyti nuotraukas kaip MJPEG video srautą.
9. Paruošti SWING komponnetai atvaizduoti informacijai iš kamerų realiu laiku
10. Palaikoma daugybe skirtingų frameworkų darbui su informacija

### Valdymo schema

Šiame skyriuje bus aptariamas sistemos valdymas pažingsniui, aptariant kiekvieno komponento rolę ir rezultatus kas iš jos tikimasi. (žiūrėti 3. paveiksliuką).



Paveikslėlis 3

1. Kamera fiziškai filmuoja esamą apinką ir perduoda matoma informaciją srautu stebėtojo klasei.
2. Stebėtojas fiksuoja ir interpretuoja gaunamus duomenis, užfiksavus neatitikimą (judėjimą) perduoda informaciją pagrindinai logikos daliai atsakingai už koordinačių skaičiavimą.
3. Kompiuteris paruošia instrukcijų sąraša kuriomis robotas turės vadovautis, kad pasiektų galutinį tikslą.
4. Valdytojas konvertuoja komandas kontroleriui, ir perduoda komandas į Arduino, kuriuos valdiklio komponentus jis turi valdyti.
5. Arduinas gauna komandų seka ir kaip valdiklis perduoda viską į valdymo pultelį
6. Pultelis sugeneruoja radijo signalus ir perduoda juos į mašinėlę

## Roboto valdymas

Siekiant užtikrinti kad robotas pasieks tikslą pirmiausia reikia išspręsti kelias problemas kurios iškyla su kameromis.

### Roboto vietos nustatymas

Siekiant valdyti robotą, reikia žinoti roboto vietą erdvėje. Tam naudojama numerinė indikacija ant roboto nugaros. (žiūrėti 4. paveiksliuką).



Paveikslėlis 4

Siekiant šio tikslo buvo panaudota Asprise Java OCR(optical character recognition) biblioteka.

Ši biblioteka skirta Java ir JavaFx aplikacijoms padeda iš paveiksliukų išgauti tekstinę informaciją. Su šios bibliotekos pagalba galima nustatyti robot vietą iš kameros įrašų erdvėje. Žinant kur reikai nuvykti kameros duomenų sraute ir žinanti roboto buvimo vieta, galima suformuoti komandas skirtas nuvykti į vietą.

### Atstumo skaičiavimas

Norint apskaičiuoti atstumus iš nuotraukos reikalinga žinoti pati svarbiausią kintamąjį, objekto tikrajį dydį, kitaip ten gali buti modelis kuris tiesiog yra artii kameros. Tam mes galime panaudoti pati robotą, jau žinome kaip identifikuoti jo vietą kadre ir kaddangi robotas mūsų mes galime sistemoje nurodyti roboto dydį. Skaičiavimas atstumo nuo kameros iki roboto nėra sudėtingas, nes nuo jutiklio ir realaus objekto dydžio santykis yra toks pat kaip kameros židinio ir atstumo iki jo. Norint apskaičiuoti pirmiausia reikia apsiskaičiuoti objekto dydį pixeliais kadre, tada tada padalinti iš viso paveiklsliuko dydžio ir padauginti iš fizinio sensoriaus lešio dydžio.

Atstumas(mm) = lešio dydis(mm) \* tikrasis objecto dydis(mm) \* paveiksliuko dydys(pikseliais)

### Tolimesnis roboto valdymas

Dabar žinant apytikslį atstumą, galima siųsti ir robotą. Baigus vygdyti paruoštų komandų paketą, Galima iš naujo nustatyti kadre roboto padėtį ir vietą iki kurios jam reikia nuvykti. Siekiant kompensuoti užgaišimą skaičiavimo metu galima prie roboto būtų prijungti judesio daviklius, kad atvykęs į vietą pats robotas savarankiškai galėtų mėginti rasti problemos sukelėją kaip įsibrovėlis ir toliau imtusi vieksmų, kaip mėginimas jį pagauti sulaikyti ar panašūs jei tik yra galimybė.

## Vizija

Šiuo projektu noriu pasiekti kelis esminius tikslus, šiame skyriuje juos ir aptarsiu.

* 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 naujiausios technologijos 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 mašineles ar panašius prietaisus kuriuos galima nesunkiai valdyti valdikliu.
* Saugos – saugos sistemos yra atgyvenusios, nuo to laiko kai jos buvo pritaikytos objektų apsaugai, nebuvo didelių pasikeitimų. Robotų įtraukimas į sistemas padėtų efektyviau apsaugoti nekilnojamajį turtą ir jame esančius vertingus daiktus. Sistema skirta suformuoti valdymą robotams atsakingiems už saugą.
* Laisvalaikio pramogoms – sistema palaiko galimybę paruošti valdymo komandas robotams, tą galima panaudoti kuriant loginius veiksmus skirtus pavyzdžiui įveikti paties padarytoms trasoms ar panašioms veikloms.

# Rezultatai ir išvados

Sukurta nauja programos versija, palaiko galimybę kaip priedas prie apsaugos sistemų pridėti apsauginius robotus. Sistema palaiko galimybę sekti ir dirbti su keliomis kameromis ir robotais. Sistema taip pat veikia greičiau, tapo paprastesnė ir patruaklesnė vartotojui, turi daugiau funkcionalumo. Deja norint didžiausio efektyvumo susiduriama su bėda, kad reikalingas brangus ir patikimas robotas, kuris galėtų patekęs į vietą imtis atitinkamų veiksmų. Parašyta sistema gali tik pranešti ir nusiųsti nuorodas robotui kaip pasiekti įvykio vietą.

Rezultat ir išvad dalyje išdstomi pagrindiniai darbo rezultatai (kažkas išanalizuota, kažkas sukurta, kažkas diegta), pateikiamos išvados (daromi nagrint problem sprendimo metod palyginimai, silomos rekomendacijos, akcentuojamos naujovs). Š

# Šaltiniai

# Sąvoku apibrėžimas

portasSantrupos

# Priedai

<Config>

<Robots>

<Robot>

<Name>guardian1</Name>

<Forward>1</Forward>

<Backward>3</ Backward >

<Left>2</ Left >

<Right>4</ Right >

</Robot>

<Robot>

<Name>guardian2</Name>

<Forward>5</Forward>

<Backward>7</ Backward >

<Left>6</ Left >

<Right>8</ Right >

</Robot>

</Robots>

<Cameras>

<Camera>

<Name>camera1</Name>

<Port>COM3</Port>

</Camera>

<Camera>

<Name>camera2</Name>

<Port>COM4</Port>

</Camera>

</Cameras>

</Config>