



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Baigiamasis bakalauro darbas

Stalo žaidimo „Kauliukų šachmatai“ kūrimas
(Table Game „Dice Chess“ Development)

Atliko: 4 kurso, 1 grupės studentas

Nojus Štelbys (Parašas)

Darbo vadovas:
Lekt. Linas Būtėnas

Vilnius
2020

Turinys

Pratarmė.....	4
Anotacija.....	5
Summary.....	6
Įvadas.....	7
1. Žaidimo taisyklės.....	9
1.1 Žaidimo lenta.....	9
1.2 Žaidimo eiga.....	10
1.3 Karaliaus figūrėlė ir žaidimo tikslas.....	11
2. Analizė.....	12
2.1 Panašūs žaidimai	12
2.1.1 Šachmatai.....	12
2.1.2 „Dice King Chess“	12
2.2 Tikslinė auditorija.....	13
3. Žaidimo įgyvendinimas	14
3.1 Sistemos architektūra	14
3.2 Naudojami įrankiai	14
3.3 Duomenų bazė.....	15
3.4 Direktorijos struktūra	16
3.5 Svetainės struktūra	17
3.6 Žaidimo figūrėlės	17
3.6.1 Kauliukų klasė „DicePiece“.....	17
3.6.2 Kauliukų elgsenos iškelti iššūkiai.....	18
3.7 Figūrėlių inicijavimas.....	19
3.8 Komponentai	19
3.8.1 Žaidimo lauko langeliai	19
3.8.2 Nukirstos figūrėlės.....	20
3.8.3 Žaidimo lenta	20
3.8.4 Žaidimas.....	20
3.8.5 Žaidimo pabaiga	23

3.9	Žaidimo būsenos duomenų modelis	23
4.	Testavimas	25
	Išvados ir Rekomendacijos	26
	Literatūros sąrašas	27

Pratarmė

Kompiuteris ir internetas, jų kuriamas potencialas leidžia prikelti nepelnytai užmirštus žaidimus ir juos populiarinti. Mano pasiryžimas į virtualią erdvę perkelti mažai žinomą loginį stalo žaidimą „Kauliukų šachmatai“ nebūtų realizuotas, jei ne kelių organizacijų ir žmonių parama. Visų pirma norėčiau padėkoti Vilniaus universitetui, suteikusiam man reikalingų pamatinių žinių apie informacines technologines, skatinusiam kūrybiškai žvelgti į programavimo procesą.

Atskirai norėčiau paminėti šio diplominio darbo pradžioje man talkinusiam lektoriui Kazimierui Mickui. Žinoma, sėkmingais rezultatais nebūtų galima džiaugtis, jei ne šio darbo vadovo lektoriaus Lino Būtėno sisteminga proceso priežiūra ir nuodugnios konsultacijos.

Ne mažesnės padėkos verta įmonė UAB „Exadel“, kur atlikau praktiką. Čia susipažinau su šiam darbui svarbia „JavaScript“ biblioteka „React.js“. Patirtis, įgyta praktikos metu, leido suprasti ir pasinaudoti šios programavimo aplinkos privalumais, jos teikiamomis galimybėmis.

Anotacija

Šiame darbe gilinamasi į mažai žinomo stalo žaidimo „Kauliukų šachmatai“, sukurto dar praėjusio šimtmečio 9-ame dešimtmetyje, bet dabar nebeegzistuojančio, perkėlimo į internetinę erdvę galimybes. Darbas atliktas analizės ir testavimo būdu, pasiteikiant kelias programavimo įrankius: React.js, Node.js, MongoDB ir Material (pagrinde React.js). Sukūrus sklandžiai veikiančią kompiuterinę žaidimą „Kauliukų šachmatai“, ir jį išpopuliarinus, žmonės – įvairių amžiaus grupių atstovai – turėtų galimybę susipažinti ir lavinti įvairius smegenų gebėjimus (vizualizuoti, įsiminti, prognozuoti), žaidžiant patrauklų, įvairių gebėjimų reikalaujantį žaidimą.

Summary

In this work, the possibilities of transferring a little known table game “Dice Chess”, created in the 1990s, but currently non existent, to the online space are explored.

Before programming, similar games were analyzed, such as traditional „Chess“ and simpler version of „Dice Chess“- „Dice King Chess“. However, „Dice King Chess“ has no virtual space adaptation and was chosen to analyze its rule adaptation, because nothing like „Dice Chess“ was found on the internet.

Because the goal of moving the game to a virtual environment is to reach a larger audience, website approach was chosen and therefore architecture of the game system consists of two main parts: a page server and a database. The following tools and libraries have been selected for this: React.js, Node.js, MongoDB and Material-ui.

After implementing the idea, it turned out that the chosen „JavaScript“ library "React.js" really was the right choice for this kind of application - the breakdown of the code into components allows flexible and dynamic use and reuse of already created code parts. At the same time, dynamic upgrades of the same components allow for a smooth and dynamic gameplay.

Testing has also shown that this game is suitable for all ages - young and old alike. The many choices available over the course of the game make its results difficult to predict, which inspires for new trials in new matches.

By creating a well-functioning computer game "Dice Chess" and popularizing it, people - representatives of different age groups - would have the opportunity to familiarize themselves with and develop various brain abilities (visualize, memorize, predict) by playing an attractive, yet demanding game.

Įvadas

Per pastaruosius dešimtmečius, kompiuteriniai žaidimai paplito visame pasaulyje sukurdami naują subkultūrą, kurią sudaro įvairiausio tipo kompiuteriniai žaidimai, aptinkami tiek pačiuose kompiuteriuose, tiek nešiojamose planšetėse ar išmaniuosiuose telefonuose. Laikui bėgant, kompiuterinių žaidimų kultūra vystėsi kartu su interneto kultūra ir didėjančiu mobiliųjų žaidimų populiarumu. Daugelis kompiuterinius žaidimus žaidžiančių žmonių įvardijami kaip žaidėjai (angliškai „gamers“), tačiau tai gali reikšti bet ką, pradedant, kad žmogus žaidžia savo malonumui ir baigiant didele aistra kompiuterinių žaidimų kultūrai. Kompiuteriniams žaidimams tampant socialiai įtrauktiems, dėl daugialypės terpės ir internetinių galimybių, žaidėjai dažnai jungiasi kartu ir sudaro milžiniškus augančius socialinius tinklus. Žaidimai gali būti ir pramoga, ir varžybos, nes vis labiau populiarėja nauja tendencija, vadinama elektroniniu sportu. Kadangi laikui bėgant kompiuterinių žaidimų populiarumas taip eksponentiškai išaugo, jie turėjo, ir vis dar turi, didelę įtaką populiariajai kultūrai, todėl šiandien, kompiuterinius žaidimus galima pamatyti socialinėje žiniasklaidoje, politikoje, televizijoje, filmuose, muzikoje ir t.t.

Tačiau sprendimas lėmęs šį diplominio temos pasirinkimą yra siekis – pasinaudojant kompiuterinių žaidimų prieinamumu globaliu mastu ir žaidėjų būrimusi į socialinius tinklus, sukurtus aplink žaidžiamą žaidimą, taip žaidimą dar labiau populiarinant – populiarinti loginius lavinamuosius žaidimus. Šiame darbe ir nagrinėjama seno ir mažai žinomo stalo žaidimo „Kauliukų šachmatai“ adaptacija virtualioje aplinkoje.

Žaidimas „Kauliukų šachmatai“ buvo sugalvotas rusų fizikų, maždaug 1986 metais, ir net buvo išleista keletas egzempliorių, tačiau šiais laikais jau beveik nebeįmanoma surasti tikslų šio žaidimo taisyklių, nors egzistuoja įvairios pačių šachmatų, naudojant kauliukus, adaptacijos. Vienas iš pavyzdžių - „Dice King Chess“, kuris nagrinėjimas vėliau. „Kauliukų šachmatai“ iš pažiūros yra lengvai išmokstamas, bet sunkiai įvaldomas žaidimas – šachmatuose įprastos figūrėlės yra pakeičiamos kauliukais, žaidimo lenta yra šiek tiek didesnė nei paprastų šachmatų (9x8 langelių dydžio vietoje 8x8), ir pačių figūrėlių/kauliukų (toliau figūrėlių) yra tik po devynias kiekvienoje pusėje (18 figūrėlių iš viso vietoje 32 figūrėlių). Kadangi visos figūrėlės yra valdomos pagal tas pačias taisykles, išskyrus karaliaus figūrėlę (daugiau skiltyje žaidimo taisyklės), žaidimą išmokyti galima per penkias minutes ir greičiau, tačiau įvaldyti jį užtrunka, nes matematinė viršutinės figūrėlės plokštumos išraiška pastoviai keičiasi, priklausomai nuo padaryto ėjimo, kuris gali varijuoti, priklausomai nuo: a) pasirinktų „eiti“ langelių kiekio, b) pasirinktos ėjimo konfigūracijos. Tai šį žaidimą paverčia reikalaujančiu gerai vizualizuoti būsimas (visas galimas) figūrėlių padėties mintyse, daug mąstyti ir kurti strategijas. Pilnos žaidimo taisyklės aprašytos skiltyje „Žaidimo taisyklės“.

Pagrindinė motyvacija sukurti XX a. devintajame dešimtmetyje sukurto stalo žaidimo adaptaciją virtualioje aplinkoje, yra šio žaidimo atgaivinimas, nes niekur neįmanoma rasti „Kauliukų šachmatų“ žaidimo ar bent jo taisyklių. Negana to, turint veikiantį žaidimą, galima būtų sukurti dirbtinį intelektą šiam žaidimui, taip įgalinant žmones žaisti ir vieniems prieš kompiuterį.

Šio baigiamojo bakalauro darbo tikslas – atsižvelgiant į žaidimo „Kauliukų šachmatai“ subtilybes – sukurti programą, kuri leistų žaidėjams jį žaisti interaktyviai.

Kuriant interaktyvų žaidimą „Kauliukų šachmatai“, buvo susidurta su iššūkiais, iš esmės padiktuotais nuolatinės figūrėlės matematinės išraiškos kaitos. Taip pat buvo svarbu konkrečiam žaidėjui pateikti visas konkrečios figūrėlės teikiamas galimybes, t. y. atvaizduoti visus galimus ėjimo scenarijus.

Interaktyvaus žaidimo kūrimo eigoje ne sykį teko jį testuoti, siekiant nustatyti: a) ar jis suprantamas testuojamam žaidėjui, b) ar korektiškai atvaizduojamas žaidimo laukas, figūrėlės, jų transformacijos, kiti žaidimo elementai.

Rezultate – sklandžiai veikiantis kompiuterinis žaidimas „Kauliukų šachmatai“.

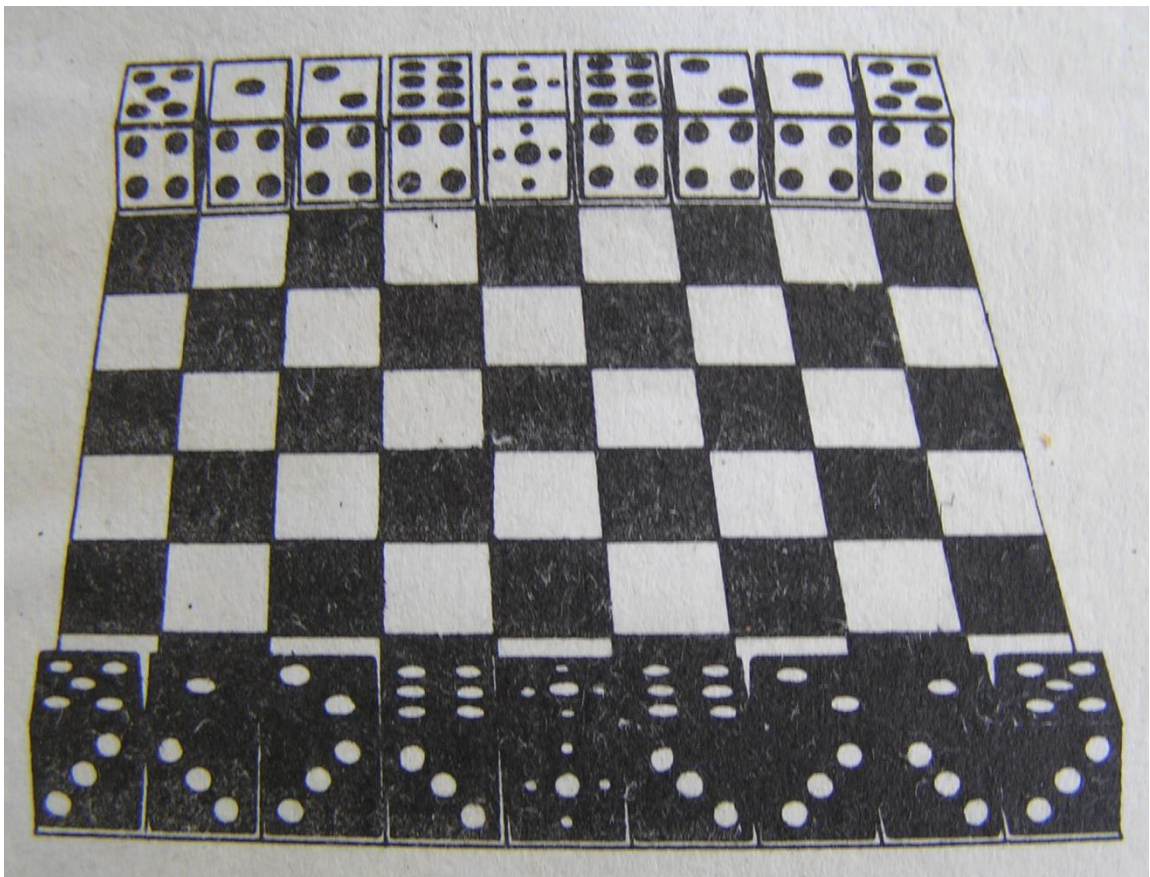
Šį rašto darbą sudaro 4 pagrindiniai skyriai („Žaidimo taisyklės“, „Analizė“, „Žaidimo įgyvendinimas“, „Testavimas“) ir išvadų skyrius. Pirmame skyriuje pristatomos žaidimo „Kauliukų šachmatai“ taisyklės, nurodant fizinę jo struktūrą ir akcentuojant idėjinę jo dalį, t.y. įvardijant, kokios dalys sudaro žaidimą, kaip jos tarpusavyje koreliuoja ir kokias galimybes turi kiekvienas žaidėjų žaidimo – vieno ėjimo – metu. Antrame skyriuje – „Analizė“ analizuojami panašūs žaidimai, jų skirtumai, galima tikslinė auditorija ir programavimo įrankiai. Trečias skyrius – „Žaidimo įgyvendinimas“ – skirtas išanalizuoti direktorijos struktūrą, žaidimo figūrėlių atvaizdavimo ir veikimo specifiką bei figūrėlių inicijavimą. Ketvirtame skyriuje – „Testavimas“ – trumpai aptariamos testavimo metu identifikuotos problemos ir vėliau rasti jų sprendimo būdai. Išvadose pateikiamos pagrindinės darbo išvados, pateikiamos rekomendacijos kaip gali būti pritaikomas šis žaidimas (ypač – mokyklinio amžiaus vaikų užklasiniam ugdyme), bei kaip jį būtų galima vystyti toliau.

1. Žaidimo taisyklės

1.1 Žaidimo lenta

Žaidimo lenta yra sudaryta iš 72 langelių, panašiai kaip šachmatų lenta, tik vietoje kvadrato sudaryto iš 8x8 langelių kraštinių, ši lenta sudaryta iš 9x8 langelių kraštinių. Dviejose kraštinėse, kurios yra priešingos viena kitos atžvilgiu, vidurinis penktas langelis yra specialiai pažymėtas- tai karaliaus figūrėlių pradinė pozicija, kurios svarba aprašyta toliau. Žaidimo figūrėlės susstatomos taip, kad kauliukų viršutinės reikšmės būtų tokios-

penki, vienas, du, šeši, karalius (būtinai turi stovėti ant karaliaus pradinės pozicijos), šeši, du, vienas, penki. Taip pat, visos kauliukų reikšmės atsuktos į priešininką turi būti ketvertai, o visos reikšmės nusuktos nuo priešininko ir žiūrinčios į lentos kraštą turi būti trejetai. Išimtis karalius- jo visos kraštinės traktuojamos kaip vienetas.



1 Pav. Pradinis žaidimo lentos sustatymas

1.2 Žaidimo eiga

Savo ėjimo metu, galite pasirinkti vieną iš kauliukų, kurį norite judinti. Pasirinktas kauliukas gali judėti tiek langelių, kiek nurodyta ant kauliuko viršutinio paviršiaus, pvz.: Jei kauliuko viršutinė reikšmė ėjimo pradžioje yra 5, tai kauliukas gali pajudėti penkis langelius, jei reikšmė buvo 3, tai kauliukas gali pajudėti 3 langelius, ir t.t.

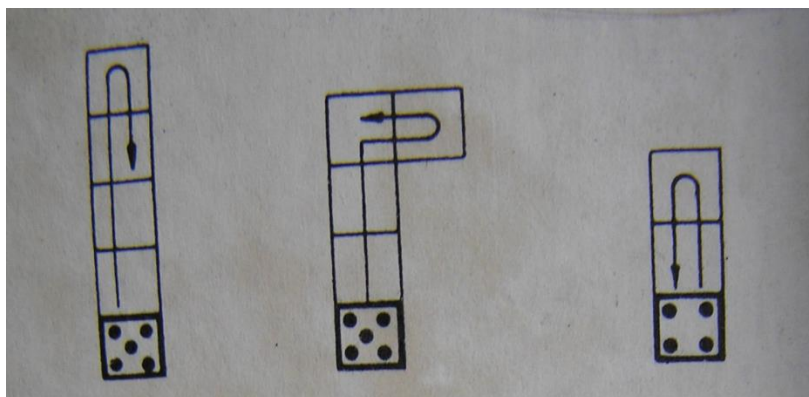
Pats kauliukas judėti gali tik judėdamas į gretimą langelį ir būtinai versdamasis per kauliuko briauną artimiausią langelio, į kurį norima judinti kauliuką, briauną, pvz.: Jei žaidimo pradžioje norite pajudinti kauliuką su reikšme 1 ant kauliuko viršaus, jis pajuda į priešais save esantį langelį versdamasis per briauną ir viršutinė reikšmė patampa 3.

PASTABA: nors kauliukui judant į kitus langelius jis verčiasi ir todėl keičiasi jo viršutinė reikšmė, kauliukas vis tiek juda tiek langelių, kokia to kauliuko reikšmė buvo ėjimo pradžioje.

Pradėjus judėti su kauliuku, turite su juo nueiti tiek langelių, kiek jis galėjo eiti ėjimo pradžioje, tai yra, kauliukas negali sustoti išnaudojęs tik dalį savo langelių ėjimo skaičiaus, ir negali nueiti toliau. Pvz.: Jei kauliuko viršutinė reikšmė buvo 5, tai pradėjus jį judinti, jis būtinai turi nueiti penkis langelius- kauliukas negali sustoti jį pajudinus tik 4 langelius ir negali kaip nors atsidurti už 6 langelių nuo pradinės pozicijos.

Taip pat, pradėjus judinti kauliuką, jis turi judėti tiesia liniją ir jį galima pradėti judinti į kairę arba dešinę nuo pradinės judėjimo trajektorijos tik vieną sykį per ėimą, Pvz.: pradėjote judinti kauliuką į priekį, kuris gali paeiti 5 langelius. Šiuo atveju galite pajudinti kauliuką visus penkis langelius į priekį, arba 4 langelius į priekį ir 1 langelį į kairę arba dešinę ir t.t. iki galimybės paeiti 1 langelį į priekį ir keturis į kairę arba dešinę.

Dėl to, kauliukas niekad negali atsidurti savo pradinėje pozicijoje ar pasisukti du ar daugiau kartų per vieną ėimą. PASTABA: žaidimo metu, patys kauliukai nėra niekad sukinėjami vietoje – kauliuko judinimas į kairėje arba dešinėje nuo pradinės judėjimo linijos esantį langelį ir yra traktuojamas kaip pasisukimas.



2 Pav. Negalimi ėjimai žaidimo metu

Negana to, kauliukas negali „praeiti“ per kitą kauliuką, nesvarbu ar priešininko, ar nuosavą.

Tam, kad numušti priešininkų kauliukus, turite su savo kauliuku baigti ėimą tame pačiame langelyje kaip priešininko kauliukas- tuo atveju priešininko kauliukas yra nuimamas nuo lentos ir jo vietoje atsistoja jūsų kauliukas. Pvz.: Jūsų kauliuko viršutinė reikšmė yra 3 ir trijų langelių atstumu

stovi priešininkų kauliukas. Laikydamosi visų judėjimo taisyklių, pajudinate kauliuką per du langelius, ir jį verčiant į priešininko langelį, prieš tai nuimant priešininko kauliuką nuo žaidimo lentos.

1.3 Karaliaus figūrėlė ir žaidimo tikslas

Yra du būdai, kaip galima laimėti žaidimą- kaip ir paprastuose šachmatuose, numušti priešininko karalių, arba su savo karaliumi atsistoti į priešininko karaliaus pradinę poziciją. Atlikus bet kurį iš šių veiksmų, žaidimas iš karto yra skelbiamas baigtu, ir tą veiksmą atlikęs žaidėjas yra skelbiamas laimėtoju.

Be to, kad karaliaus kauliukas yra viso žaidimo tikslas (karalių numušti arba nueiti su savo karaliumi į priešininko karaliaus pradinį langelį), taip pat karaliaus kauliukas ypatingas tuo, kad visi karaliaus kauliuko paviršiai traktuojami kaip vienetai, tai yra, karalius visąlaik eis tik vieną langelį.

2.Analizė

2.1 Panašūs žaidimai

2.1.1 Šachmatai

Vienas populiariausių panašių žaidimų, ir tuo pačiu žaidimas kurio pagrindu buvo sukurti „Kauliukų šachmatai“, yra be abejonės paprasti šachmatai. Pagal pasaulio šachmatų federacijos pateiktas statistikas, šachmatų žaidėjai sudaro vieną didžiausių bendruomenių pasaulyje: kasdien 605 milijonai suaugusių žaidžia šachmatų [1] ir daugiau nei 60-70% sužaistų partijų įvyksta internete, o dažniausiai internete žaidžiami šachmatai įvyksta tam dedikuotose internetinėse svetainėse - dėl lengvos prieigos nereikia nieko siųstis, užtenka prisiregistruoti paskyrą, yra net saugomas paskyros karjeros progresas.

Negana to, šachmatai buvo pirmasis žaidimas, kuriam buvo sukurtas dirbtinis intelektas, sugebėjęs įveikti geriausiai žaidžiančius šachmatų žaidėjus pasaulyje. Pirmieji bandymai sukurti tokį intelektą, prasidėjo dar vakuuminių vamzdžių kompiuterių amžiuje, apie 1950 metus. Pirmosios programos žaidė taip prastai, kad net pradedantys žaisti žmonės galėdavo jas įveikti. Tačiau tobulėjant kompiuteriams ir jiems tampant vis labiau prieinamiems, atsirado daugiau testavimo, bei informacijos rinkimo galimybių, kas įgalino spartų šachmatų dirbtinio intelekto tobulėjimą, ir po maždaug penkiasdešimt metų 1997-aisiais, šie dirbtiniai intelektai jau sugebėjo įveikti geriausiai pasaulyje žaidžiančius žmones. [2]

2.1.2 „Dice King Chess“

Panašiausia šachmatų adaptaciją į „Kauliukų šachmatai“ kurią radau, yra „Dice King Chess“, tačiau ši adaptacija turi keletą esminių skirtumų: Pirmą, naudojama mažesnė lenta (7x7 langelių kraštinės, vietoje 9x8). Antra, be karaliaus yra 7 kauliukai, išstatyti į eilę prieš karalių ir visi iš pradžių yra vienetai. Trečia, kauliukai gali judėti į bet kurį langelį, tik ne į tą iš kurio ką tik atėjo, pvz.: kauliukas su reikšme 4 ant viršutinio paviršiaus gali apsukti ratą 2x2 langelių teritorijoje ir atsirasti pradinėje pozicijoje. Ketvirta, žaidimo tikslas yra tik numušti priešininko karalių.



3 Pav. „King Dice Chess“ žaidimo lentos pradinis išdėstymas

Mažesnė lenta ir mažesnis kauliukų skaičius, nulemia mažesnį galimų ėjimų kiekį, kas lemia greitesnius raundus. Galbūt greitesni raundai ir nepadaro šios adaptacijos lengvesnės, tačiau galimybė laisviau judinti kauliukus žymiai mažiau nubaudžia už padarytas klaidas praėjusiuose ėjimuose [3][4].

Taip pat turiu patikslinti, kad tai yra tik taisyklių adaptacija kurią kai kurie žmonės žaidžia gyvai- „King Dice Chess“ neturi adaptacijos virtualioje erdvėje ir šią taisyklių adaptaciją pasirinkau analizuoti todėl, kad neradau nieko panašaus į „Kauliukų šachmatus“ internetinėje erdvėje.

2.2 Tikslinė auditorija

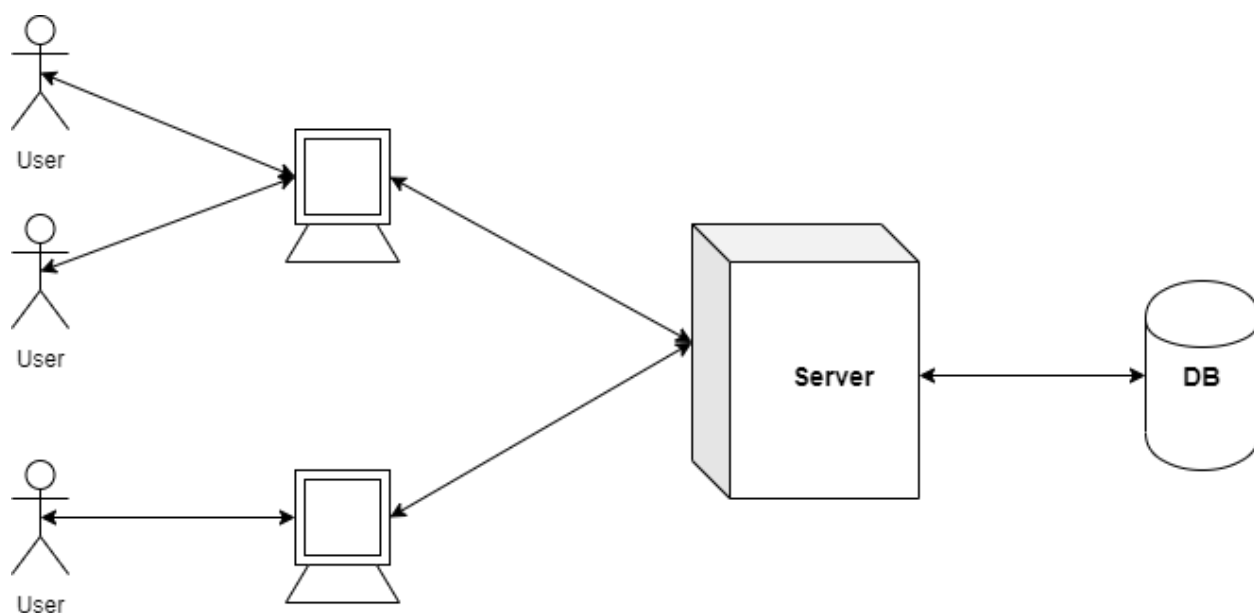
Kaip paminėta įvade, šis žaidimas lengvai išmokstamas, tačiau jį yra sunku įvaldyti, dėl figūrėlių keitimosi kiekvienu ėjimu, kas žaidimą padaro bent jau tokiu pat sudėtingu, jei ne sudėtingesniu (turint omeny ėjimų galimybių įvairovę ir matematinės figūrėlių išraiškos kaitą), nei šachmatai.

Dėl to, žaidimą žaisti gali bet kas, tačiau tikslinė auditorija būtų panaši į šachmatų – jaunimas (ypač gera terpė būtų mokyklinio amžiaus vaikų užklasinis ugdymas) ir vyresni asmenys, mėgstantys šachmatus ar kitus sunkius loginius žaidimus, nes loginį ir erdvinį mąstymą ugdantys žaidimai aktyvina smegenų veiklą. Tai liudija ir pastarųjų metų atliktų mokslinių tyrimų duomenimis: loginiai žaidimai padeda atitolinti senatvinę demenciją ir kitas su smegenų senėjimu susijusias ligas [5].

3. Žaidimo įgyvendinimas

3.1 Sistemos architektūra

Sistema susidaro iš dviejų pagrindinių dalių- puslapio serverio ir duomenų bazės. Naudotojui norint žaisti kauliukų šachmatus, pirmiausiai jis prisijungia prie puslapio naudodamasis kompiuteriu su interneto prieiga, ar kokia nors kita priemone, pajėgia naršyti internete. Jei norima žaisti per vieną priemonę, žaidimo puslapis užkraunamas, naršyklė gauna visą reikiama informacija ir komunikacija tarp naršyklės ir serverio nebeįvyksta iki kol nėra bandoma naršyti toliau puslapyje. Naudotojui norint žaisti prieš kitą žaidėją per skirtingas naršyklės, jo daromi ėjimai yra siunčiami serveriui, kuris juos perteikia duomenų basei išsaugoti ir kito naudotojo naršyklei. Naudotojui norint prisiregistruoti arba išsaugoti esamo žaidimo padėtį, informacija siunčiama serveriui, kuris šią informaciją išsaugo duomenų bazėje.



4 Pav. Architektūrinė schema

3.2 Naudojami įrankiai

Kadangi žaidimo perkėlimo į virtualią aplinką tikslas yra didesnės auditorijos pasiekimas, tai turi būti pasirinkti atitinkami įrankiai. Atsižvelgiant į šachmatų populiarumą ir panašumą į šį bakalaurinį darbą, buvo ištirti populiariausi būdai žaisti šachmatus internetu. Labai greitai paaiškėjo, kad tai yra internetiniai puslapiai, nes nereikia nieko siųsti, užtenka susikurti paskyrą ir galima iškart žaisti. Todėl sekant šiuo pavyzdžiu šis bakalaurinis darbas irgi įgyvendintas kaip internetinis puslapis. Tam pasirinkti šie įrankiai ir bibliotekos:

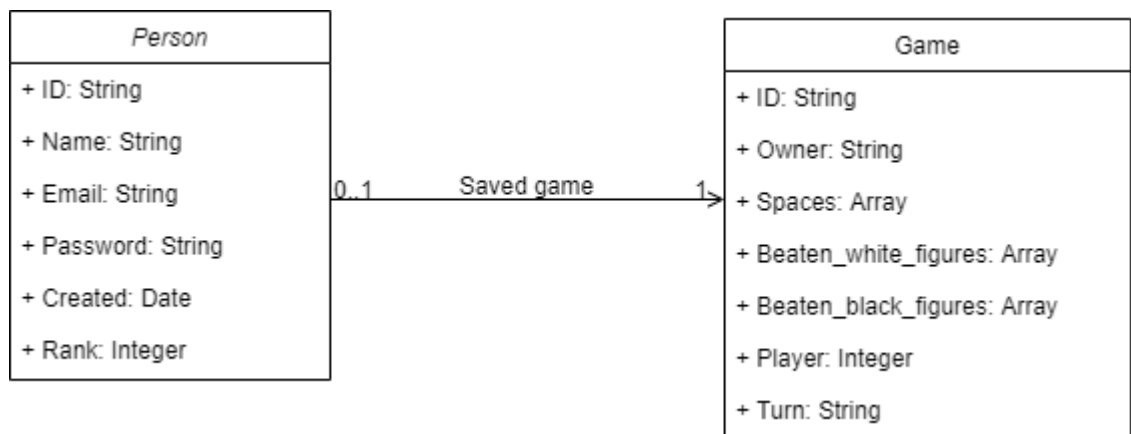
- React.js
- Node.js
- MongoDB
- Material-ui

React.js pasirinktas dėl prieš tai įgytos patirties dirbant su juo, jis gerai veikia su Node.js, kuris leidžia dinamiškai ir iškart atvaizduoti bet kokius pakeitimus svetainėje, kas yra reikalinga norint padaryti veikianti žaidimą tarp dviejų skirtingų žmonių. Taip pat, naudojant React.js, visas HTML kodas yra perkeliamas į „JavaScript“ funkcijas [6][7], kas leidžia kurti daug funkcionalumo turinčius vadinamus komponentus, kuriuos savo programoje galima pernaudoti keletą sykių iškilus poreikiui.

MongoDB pasirinktas dėl to, kad tai yra labiausiai naudojama duomenų bazė su React.js ir dėl taip pat įgytos patirties dirbant su MongoDB, ir Material-ui, palaikantis vientisą lakonišką dizainą.

3.3 Duomenų bazė

Visa informacija saugoma naudojant duomenų bazę „MongoDB“. Taip yra todėl, kad tai yra populiariausia ir dažniausiai naudojama duomenų bazė su React.js karkasu. Vartotojas turi pagrindinę informaciją: identifikacinį numerį, vartotojo vardą, el. Pašto adresą, slaptažodį, sukūrimo datą ir reitingą. Kai žaidėjas nori išsaugoti žaidimą, žaidimo informacija būna išsaugoma atskirai, susiejant žaidimo informaciją su žaidėjo ID. Apie patį žaidimą laikoma informacija yra identifikacinis numeris, išsaugojęs žaidimą naudotojas, okupuotų langelių objektų masyvas, du numuštų figūrėlių masyvai, kurio žaidėjo ėjimas. Vienas žaidėjas galėtų turėti tik iki vieno išsaugoto žaidimo, ir išsaugotas žaidimas visada turi turėti savininką.



5. Pav. No SQL duomenų bazės schema

3.4 Direktorijos struktūra

Pabaigto projekto direktorijos src aplankas yra sudarytas iš keturių kitų aplankų: komponentų, padėjėjų, figūrėlių ir piktogramų aplankų. Figūrėlių aplankale esantys failai aprašo pačių figūrėlių ypatybes ir elgsenas, piktogramų aplankas savyje laiko tik reikiamas piktogramas. Padėjėjų aplankale esantys failai naudojami paleidimo metu inicijuoti reikiamas figūrėles, o komponentų aplankas yra sudarytas iš daug kitų aplankų, kurie savyje laiko „js“ failus, vykdančius su to aplanko pavadinimu susijusius veiksmus. Svarbiausi failai yra aplankuose „game-component“ ir „game-component-helpers“. „Game.js“ savyje laiko visą žaidimo logiką ir suriša visus failus esančius „game-component-helpers“ aplanke.

```
> node_modules
> public
> src
  > components
    > about-component
    > game-component
      > Game.js
    > game-component-additions
      > Board.js
      > FFB.js
      > Not-active-space.js
      > Side-show.js
      > Space.js
      > New-board.js
      > Side-board.js
      > DicePiece.js
      > dk.js
      > Piece.js
      > Show.js
    > home-component
    > login-component
    > navigation-component
    > profile-component
    > signup-component
    > user-info-component
    > user-info-popup-component
  > icons
  > App.css
  > App.js
  > index.css
  > index.js
```

6 Pav. Projekto direktorijos struktūra

3.5 Svetainės struktūra

Svetainė turi pagrindines svetainėms būdingas dalis: pagrindinį puslapį, navigacinę puslapio juostą, registracijos puslapį, prisijungimą ir žinoma paties žaidimo puslapį. Tiek prisijungimo lango, tiek registracijos laukeliai yra patikrinami ar teisingai užpildyti ir, ar nėra kenksmingų reikšmių duomenų bazei dar prieš siunčiant užklausą į patį „backend“.

3.6 Žaidimo figūrėlės

Pagrindinė figūrėlių klasė, kurią vėliau paveldi labiau specializuotos figūrėlės, yra klasė „Piece“ faile „piece.js“. Joje laikoma informacija apie žaidėją, kuriam priklauso ši figūrėlė, nuorodą į piktogramą, atvaizduojančią tą figūrėlę, bei informaciją apie viršutinį ir į lentos pietus, bei rytus atsuktą kauliuko paviršių skaičius. Visa ši informacija yra perduodama iš labiau specializuotų poklasių „DicePiece“, „dk“ ir „Show“.

```
export default class Piece {
  constructor(player, iconUrl, dice, sideFront, sideRight){
    this.player = player;
    this.style = {backgroundImage: `url(${iconUrl})`,
  };
  this.dice = dice;
  this.sideFront = sideFront;
  this.sideRight = sideRight;
}
```

7 Pav. „Piece“ klasėje laikoma informacija apie figūrėles.

„DicePiece“ ir „dk“ poklasės reprezentuoja kauliukus ir karaliaus figūrėlę, tuo tarpu poklasė „Show“ reprezentuoja neinteraktyvių kauliukų naudojamas esybes, norint atvaizduoti žaidimo metu pasirinkto kauliuko šonines reikšmes. Todėl klasė „Show“ turi tik dvi funkcijas, kurios tik pakeičia tos esybės piktogramą pagal poreikį, arba piktogramą paslepia visai.

Tuo tarpu kitos dvi aktyvesnio naudojimo klasės, turi funkcijas „isMovePossible()“ ir „getSrcToDestPath()“, kurios apskaičiuoja, ar pasirinktas ėjimas yra tinkamu atstumu nuo pradinio taško atsižvelgiant į kauliuko reikšmę, bei langelius, kuriais buvo keliauta. Tačiau be šio funkcionalumo, kauliukų klasė „DicePiece“, turi dar daug kitų funkcijų.

3.6.1 Kauliukų klasė „DicePiece“

Ši klasė yra iš visų trijų figūrėlių klasių pati sudėtingiausia ir daugiausia vykdoma klasė. Taip yra dėl to, kad karaliaus kauliuko klasė, dėl karaliaus vienodų paviršių reikšmių, neturi apskaičiuoti kaip keisis paviršių matematinės reikšmės, ne taip, kaip paprastų kauliukų. Taip pat karalius juda visą laiką tik vieną langelį, kas lemia, kad būdų karaliui paeiti į kitą langelį yra tik vienas.

Kauliukui turint viršutinę reikšmę didesnę nei vienetas, atsiranda galimybė, kad į tą patį galutinio tikslo langelį, kauliukas gali pajudėti dviem skirtingais keliais, kas yra svarbu, nes nuo pasirinkto kelio, kinta atsivertusi viršutinė kauliuko reikšmė. Todėl ši klasė papildoma per kuriuos langelius keliaujama apskaičiuojančią funkciją, kuri apskaičiuoja alternatyvų kelią, jei toks įmanomas.

Pagrindinis šios klasės funkcionalumas ateina iš galėjimo apskaičiuoti galutiniame langelyje būsimos kauliuko paviršių reikšmes. Tam pasiekti funkcijai reikia tik kelio, kuriuo kauliukas judėjo. Funkcija išanalizuoja kiekvieną paėjimą į kitą langelį iki tikslo ir sudaro keturių krypčių žemėlapi. Šis žemėlapis yra perduodamas į kitą funkciją, kurioje jau apskaičiuojama, kaip pasikeitė kauliuko kraštinės po kiekvieno pajudėjimo, tačiau pabaigoje gražinama tik reikalingos paskutinio laukelio paviršių reikšmės.

Jei užklausa apie galutiniame langelyje atsidūrusio kauliuko reikšmes ateina kartu su papildoma reikšme vykdyti, apskaičiavusi kauliuko būsimas reikšmes, funkcija tuo pačiu ir pakeičia senas reikšmes į naujas.

3.6.2 Kauliukų elgsenos iškelti iššūkiai

Sudėtingiausia projekto dalis yra nuolatos besikeičiančios žaidimo figūrėlės ir ne tik dėl to, kad pastoviai keičiasi jų elgsena, bet ir todėl, kad figūrėlės pokytis tiesiogiai priklauso nuo vartotojo ir kraštinių kauliuko reikšmių.

Pirmiausiai buvo sukurtas tik dviejų kauliuko paviršių reikšmių išsaugojimas, viršaus ir pagal lentą į pietus žiūrinčio paviršiaus, nepasiteisino. Nors teoriškai, dviejų kauliukų paviršių užtenka sužinoti visiems paviršiams, praktiškai yra 30 skirtingų kombinacijų pagal kurias reikėtų tikrinti pasirinktą kauliuką kiekvienu ėjimu, be to kodas tampa išpūstas ir pernelyg komplikotas. Dėl to pataisiau, kad kauliukų klasės esybės savyje laikytų papildomą, į rytus pagal lentą žiūrinčio paviršiaus reikšmę.

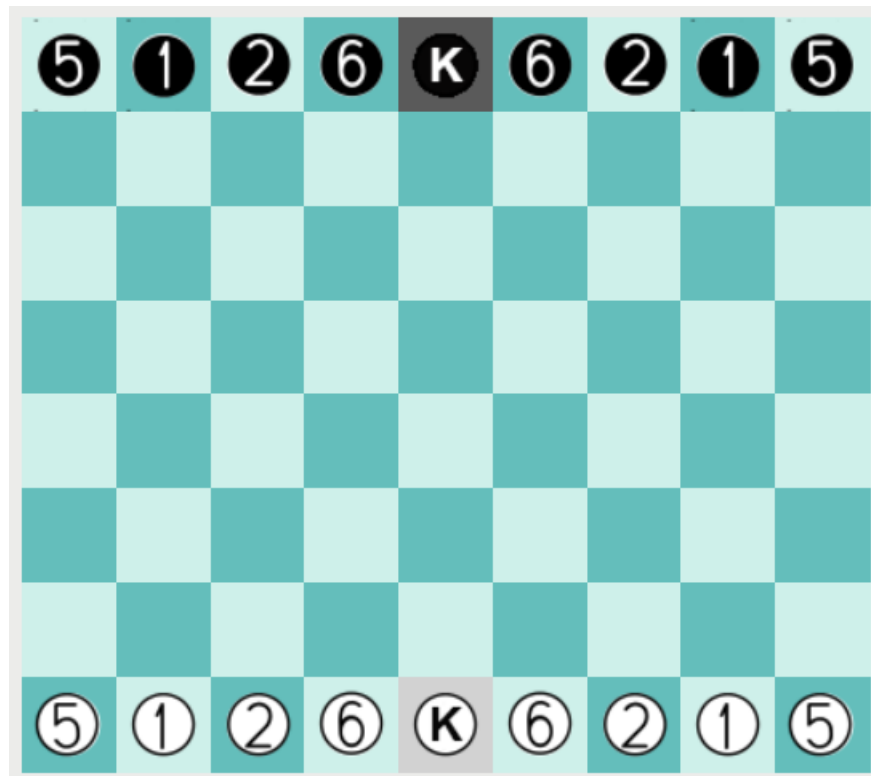
Perėjus prie trijų reikšmių laikymo, greitai atradau galimybę paprastais aritmetiniais veiksmiais sužinoti reikiamo paviršiaus reikšmę, nes priešingų paviršių reikšmė visada lygi septyniems. Paprastas ir greitas kauliuko paviršių apskaičiavimo būdas leido automatizuoti kauliuko reikšmių kitimą, skaičiuojant po vieną langelį pagal naudojamą kelią.

```
sideSwitch(top, front, sum) {  
  let temp, result = [];  
  for (let i = 0; i < sum; i++) {  
    temp = front;  
    front = top;  
    top = 7-temp;  
  }  
  result.push(top, front);  
  return result;  
}
```

8 Pav. Kauliuko reikšmių pokyčio apskaičiavimas naudojant viršutinę, į verčiamą pusę žiūrinčio paviršių reikšmes ir kiekį langelių kauliukas keliauja

3.7 Figūrėlių inicijavimas

Naudojantis vadinamų padėjėjų „board-initialiser.js“ ir „side-board-initialiser.js“ klasių funkcijų pagalba, gaunamas 72 kvadratų masyvas, iš kurių 18 yra iš anksto pradinėse pozicijose užpildyti figūrėlėmis, o likę kvadratai tušti. Be žaidimo lentos kvadratų masyvo, taip pat inicijuojamas mažesnis, devynių kvadratų masyvas, kuris bus naudojamas atvaizduoti šonines pasirinkto kauliuko reikšmes.



9 Pav. Inicijuota žaidimo lenta su figūrėlėmis

3.8 Komponentai

Visa žaidimo logika yra surišta ir valdoma „game.js“ komponento, kuriame yra naudojami visi kiti mažesni projekto komponentai, kartais juos pernaudojant daug kartų. Šis skaidymas į komponentus leidžia sukurti labai interaktyvias žaidimo dalis.

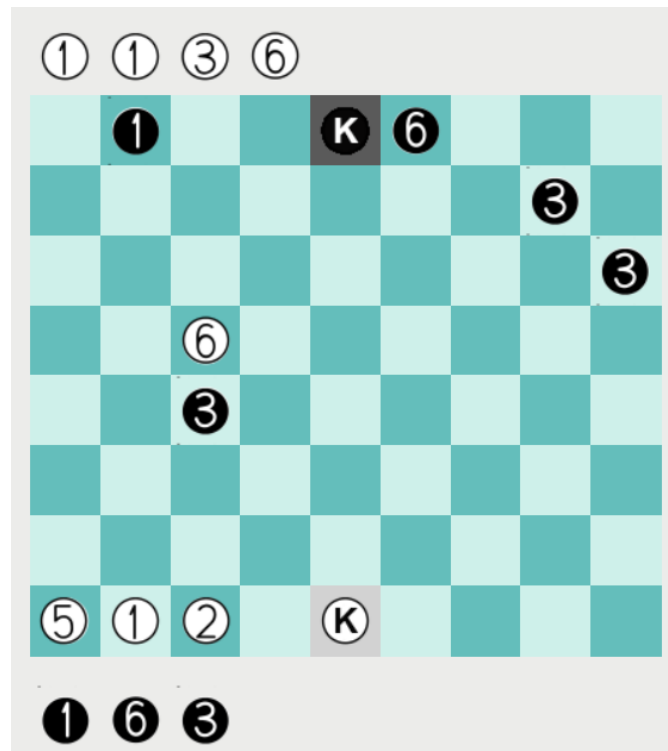
3.8.1 Žaidimo lauko langeliai

Žaidimo lauko langeliai yra patys mažiausi komponentai visame projekte. Jie yra 48 pikselių dydžio, sudaryti tik iš mygtuko, kuris savyje laiko informaciją tik apie spalvą ir jos atspalvį,

kreipimasi į žaidimo komponento funkciją įvykus paspaudimui, bei figūrėlės (jos nuotrauką), jei tokia egzistuoja ant to langelio.

3.8.2 Nukirstos figūrėlės

Kai žaidimo eigoje yra nukertamos figūrėlės, jos yra atvaizduojamos šalia žaidimo lentos, prie ją nukirtusio žmogaus. Nukirstų figūrėlių blokas gauna du masyvus- baltų nukirstų figūrėlių ir juodų, kuriuos šis komponentas tiesiog paleidžia per ciklą ir atvaizduoja dviejose linijose.



10 Pav. Nukirsti kauliukai šalia žaidimo lentos.

3.8.3 Žaidimo lenta

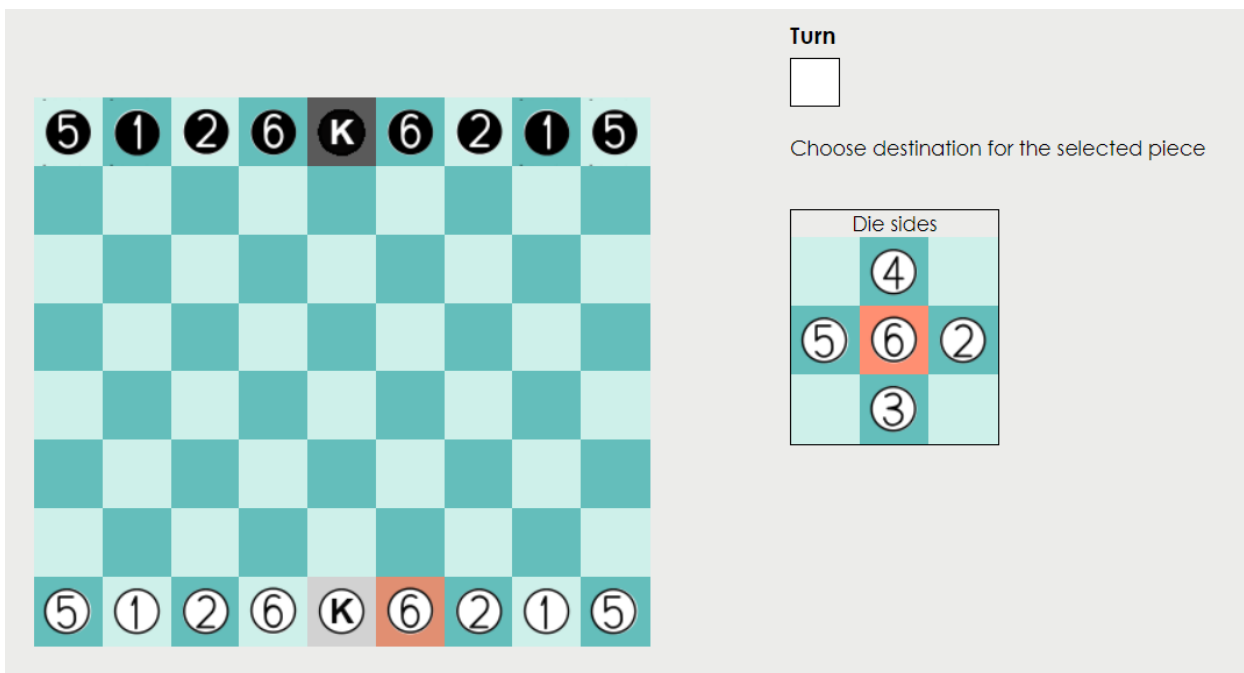
Žaidimo lentos komponento vienintelė užduotis yra pateikti tinklą sudarytą iš 72 langelių. Tai jis padaro tarpininkaudamas tarp žaidimo komponento ir langelių komponentų, naudodamas dvigubą ciklą sudaryti lentai ir tuo pačiu priskirdamas langeliui atspalvį.

3.8.4 Žaidimas

Žaidimo komponente kontroliuojamas visas žaidimo įvykių bei logikos srautas. Paspaudus ant bet kurio žaidimo lentos langelio, yra žiūrima ar langelis nėra tuščias ir ar figūrėlė priklauso tam pačiam žmogui, kurio ėjimas tuo metu vyksta. Jei tai nėra to žaidėjo figūrėlė, žaidimas perspėja apie

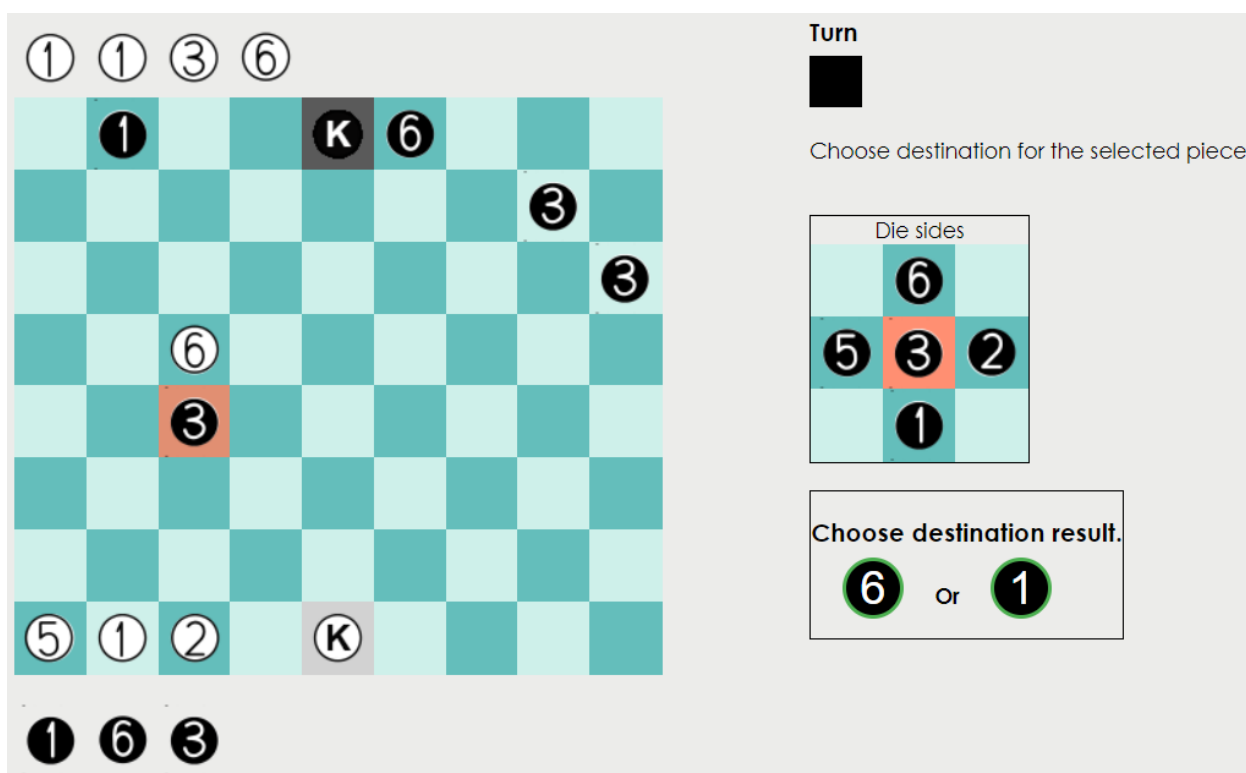
netinkamą pasirinkimą ir paprašo pasirinkti tinkamą figūrėlę, nurodydamas kokios spalvos figūrėlė turėtų būti (pagal žaidėjo, darantį ėjimą, spalvą).

Jei vis dėl to tai tinkama figūrėlė, žaidimas išsaugo paspausto langelio koordinatę atmintyje, paryškina paspaustą langelį naudodamas foninį atspalvį, parodo kauliuko šonines reikšmes ir laukia kito žaidėjo veiksmo. Žaidėjui paspaudus ant kito langelio, žaidimas patikrina ar tame langelyje nestovi nuosava figūrėlė, taip pat ar kauliukas gali paeiti į tą langelį pagal savo reikšmę ir, ar kiti kauliukai neužblokuoja kelio iki pasirinkto langelio. Jei bent vienas iš tikrinimų pasirodo esantis tiesa, žaidimas ištrina iš atminties išsaugotą pasirinkimą, ištrina prieš tai pasirinkto kauliuko foninį atspalvį, ir žaidimas iš naujo prašo pasirinkti kauliuką.



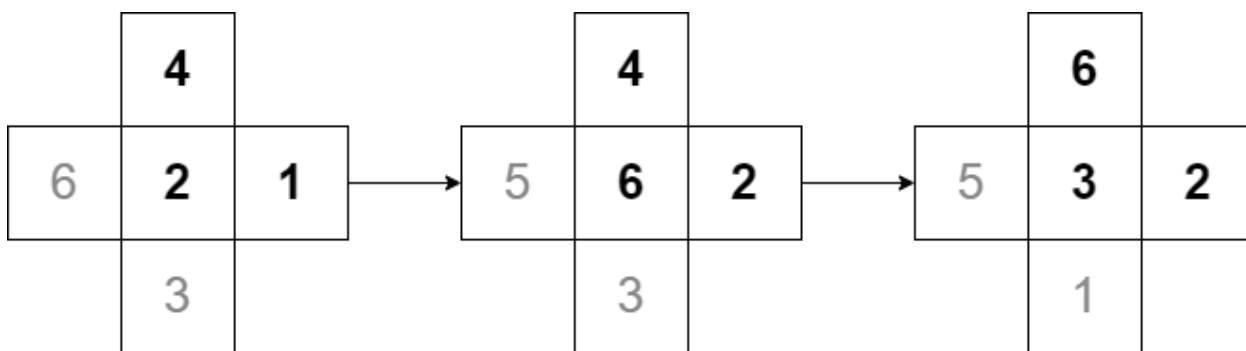
11 Pav. Pasirinktas nuosavas kauliukas paryškkinamas

Pasirinkus ir paspaudus įmanomą ir galimą ėjimą žaidimas patikrina ar galimi du skirtingi nuėjimo į langelį būdai. Jei įmanomas tik vienas kelio būdas, pavyzdžiui trys langeliai į priekį, tai žaidimas iš kart apskaičiuoja kauliuko paviršių pasikeitimus ir įvykdo pasirinktą ėjimą. Tačiau jei yra galimi du skirtingi keliai į galutinį langelį, kauliukas gali padaryti tą patį ėjimą, bet atsivertusi reikšmė galutiniame langelyje kis pagal pasirinktą kelią, todėl žaidimas šalia lentos atvaizduoja langą su dvejais galimais galutiniais kauliuko rezultatais ir laukia, kol žaidėjas pasirinks vieną iš jų. Pasirinkus vieną iš duotų variantų, žaidimas įgyvendina pasirinktą ėjimą ir perleidžiama eilė kitam žaidėjui. Paspaudus bet kur kitur, opcijos pradingsta ir žaidimas prašo iš naujo pasirinkti kauliuką ir daryti ėjimą.



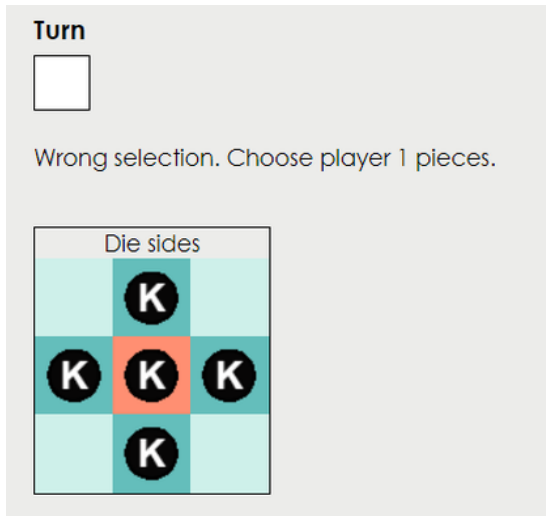
12 Pav. Atsiradęs pasirinkimo langas esant dviem skirtingam būdam patekt į galutinį langelį

Kaip anksčiau minėta, žaidimas saugo trijų kauliuko paviršių informaciją- viršutinio, priekinio ir dešinio paviršiaus reikšmes. Saugomos tik trys reikšmės, nes kitoje kauliuko pusėje esančią reikšmę visada galima apskaičiuoti iš septynių atėmus priešingo paviršiaus reikšmę- skirtingų paviršių reikšmių suma kauliukuose visada bus lygi kauliuko paviršių skaičiui plus vienas. Paprastuose šešių kraštinių kauliukuose ši reikšmė visada bus lygi septyniams. Todėl žaidimas kauliuko galutinę reikšmę apskaičiuoja patikrindamas kokia kauliuko reikšmė bus kiekviename laukelyje, per kurį figūrėlei tenka keliauti.

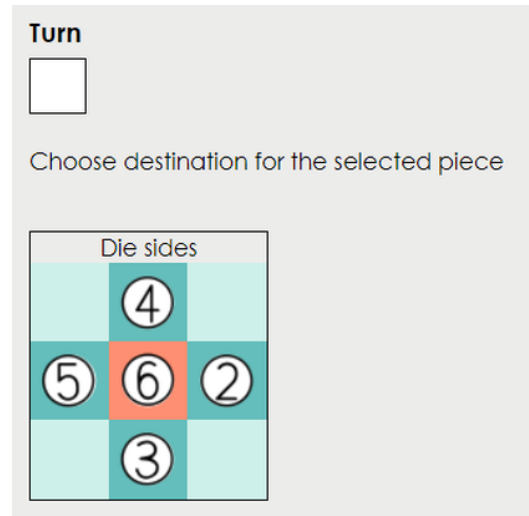


13 Pav. Kauliuko besikeičianti informacija darant ėjimą

Nesvarbu kokios spalvos figūrėlė buvo paspausta, šalia lentos esanti 3x3 parodomoji lenta nurodo paspausto kauliuko kraštines pseudo kaip kauliuko plano išsklotinę, be apatinės kauliuko reikšmės, tam kad galima būtų analizuoti žaidimo padėtį, planuoti ėjimus į priekį, bei bandyti nuspėti galimus priešininko ėjimus.



14 Pav. Balto žaidėjo ėjimo metu paspaustas priešininkų kauliukas



15 Pav. Balto žaidėjo ėjimo metu paspaustas nuosavas kauliukas

3.8.5 Žaidimo pabaiga

Po kiekvieno padaryto ėjimo, žaidimas papildomai patikrina ar nebuvo padaryti vienas iš dviejų veiksmų - pirma, ar buvo nukirsta karaliaus figūrėlė, antra, ar padariusio ėjimą žaidėjo karaliaus figūrėlė atsistojo ant priešininko karaliaus figūrėlės startinės pozicijos. Jei yra įgyvendinama viena iš šitų sąlygų, ką tik padaręs ėjimą žaidėjas yra skelbiamas laimėtoju.

3.9 Žaidimo būsenos duomenų modelis

Žaidimo būsenoje yra laikoma informacija reikalinga sklandžiai žaidimo eigai. Ją sudaro:

- „spaces“ – inicijuojama su „newBoard()“ funkcijos pagalba, savyje laiko informaciją apie langelius: 54 tuščios reikšmės ir 18 figūrėlių (kiekvienam žaidėjui po 9).
- „sideBoard“ – inicijuojama su „sideBoard()“ funkcijos pagalba, savyje laiko informaciją apie neinteraktyvius langelius, naudojamus atvaizduoti šonines pasirinkto kauliuko reikšmes.
- „WFS“ – pradeda kaip tuščias masyvas, žaidimo eigoje užpildomas baltais nukirstais kauliukais ir naudojamas jiems atvaizduoti šalia žaidimo lentos.
- „BFS“ – lygiai tas pats, kaip ir „WFS“, bet skirtas juodiems kauliukams.
- „player“ – pradeda su reikšme 1, žaidimo eigoje keičiasi tarp 1 ir 2 paeiliui, indikuojant kurio žaidėjo ėjimas tuo metu yra.
- „selectedStart“ – pradeda kaip -1 ir yra -1 jei joks kauliukas nėra pasirinktas. Jei kauliukas pasirenkamas, savyje laiko pasirinkto kauliuko koordinatas iki kol atliekamas veiksmas su pasirinktu kauliuku. Po to grįžta atgal į -1 padėtį.

- „selectedEnd“- tas pats, kaip ir „selectedStart“, tik laiko informaciją apie pasirinkto galutinio laukelio koordinatės.
- „status“- naudojamas įspėjimui ar instrukcijoms atvaizduoti.
- „turn“ – pagal „player“ būseną savyje laiko žaidėjo spalvą, šalia lentos esančiam indikatoriui.
- „option1“, „option2“- savyje laiko kauliuko reikšmės pakitimo pasirinkimo galimybes, esant dviem būdam nukeliauti į pasirinktą langelį.
- „path“, „pathAlt“- esant pasirinkimo galimybei, savyje laikinai laiko abiejų pasirinkimų naudojamus kelius į tikslą.
- „open“- pradeda „false“ pozicijoje, naudojamas paslėpti arba parodyti pasirinkimo langą.
- „victory“- pradeda „false“ pozicijoje, naudojamas paslėpti arba parodyti langelį, skelbiantį žaidimo nugalėtoją.

4. Testavimas

Programavime, testavimas vyksta naudojant vadinamus bandomuosius atvejus (angliškai „test cases“). Bandomasis atvejis yra įvesties, vykdymo sąlygų, testavimo procedūros ir laukiamų rezultatų, apibrėžiančių vieną testą, kuris turi būti atliktas tam tikram programinės įrangos testavimo tikslui pasiekti, pavyzdžiui, įvykdyti tam tikrą programos kelią ar patikrinti konkrečios specifikacijos atitikimą konkrečiam reikalavimui. [8] Dažniausiai testavimo atvejai yra vykdomi metodiškai, o ne atsitiktinai. Formaliai apibrėžti bandymo atvejai leidžia tuos pačius testus pakartotinai atlikti su paeiliui esančiomis programinės įrangos versijomis, leidžiant atlikti veiksmingą ir nuoseklų regresijos testą.

Tačiau po pirminių žaidimo testavimo bandomųjų atvejų, greitai paaiškėjo, kad tik paprastų testavimo atvejų neužtenka, nes žaidimas turi daugybę ėjimo galimybių, todėl jų visų aprašyti ir nuspėti visus galimus variantus yra beveik neįmanoma. Todėl buvo pasirinktas kitoks testavimo būdas - naudojamo testavimas (angliškai „Usability testing“).

Naudojamumo testavimas yra į vartotoją orientuotas metodas, naudojamas norint įvertinti produktą, išbandant jį su vartotojais. Tai gali būti vertinama kaip nepakeičiama naudojimo praktika, nes ji tiesiogiai nurodo, kaip realūs vartotojai naudoja sistemą. [9] Tiesioginiai atsiliepimai apie sistemą ir joje surastas klaidas ar nusiskundimus, leidžia greitai šias klaidas pašalinti, ar pataisyti ne tokias patogas sistemos dalis.

Kadangi tarp šeimos narių buvo dažnai žaidžiami kauliukų šachmatai, testavimas buvo atliktas su 2 grupėmis žmonių: a) iš artimųjų tarpo (6-ųjų ir 7-ųjų metų mergaitės ir žmonės virš 45 metų), b) tuos, kurie niekada nebuvo matę šio žaidimo (atitinkamai – jaunesni ir vyresni). Abiejose testavimo grupėse jau pirmu testavimo metu iškilo akivaizdi problema – ne taip, kaip realybėje, virtualioje adaptacijoje nebuvo įmanoma sužinoti, kokios yra kitos kauliuko kraštinės. Nors žaidimo pradžioje pagal taisykles yra aišku, kur kokia reikšmė turėtų būti kauliuko šoniniuose paviršiuose, praėjus keliems ėjimams, buvo užmiršamos šios reikšmės, todėl planuoti į ateitį ir kurti strategijas tapdavo beveik neįmanoma. Tai išspręsta šalia žaidimo lentos atvaizduojant paspausto kauliuko šoninių paviršių išsklotinę.

Kita iškilusi problema – esant dviem ėjimams tuo pačiu kauliuku į tą pačią vietą, žaidimas parinkdavo ir įvykdydavo pirmą apskaičiuotą kelią, nors ir žaidėjas norėjo paeiti kitaip. Tai išspręsta atvaizduojant pasirinkimo langą iškilus tokiems atvejams.

Taip pat po pirmo testavimo iškilo nusiskundimų dėl prastai išsiskiriančių juodų/ baltų figūrėlių nuo juodų/baltų šachmatų lentos langelių, nes juodos ir baltos spalvų kontrastas po ilgo žaidimo išsivargindavo akis.

Todėl buvo pakeista juoda/ balta lentos spalva į ne taip akį rėžiančius, šviesesnį bei tamsesnį melsvus atspalvius.

Antras testavimas (testojamųjų imtis – jau žaidę ir naujokai, įvairių amžiaus grupių atstovai) parodė, kad žaidimas vyksta sklandžiai bei sėkmingai užbaigiamas.

Išvados ir Rekomendacijos

Igyvendintas užsibrėžtas tikslas, t. y. sukurtas internetinis kompiuterinis žaidimas „Kauliukų šachmatai“ kartu su jį palaikančia svetaine ir viskas patalpinta į ant universiteto išteklių sukurtą virtualią mašiną, taip padarant svetainę pasiekiamą visiems. Ją galima aplankyti adresu <http://193.219.91.103:16292/>. Taip pat įsitikinta, kad:

1. Pasirinkta „JavaScript“ biblioteka „React.js“ tikrai buvo tinkamas pasirinkimas tokio pobūdžio aplikacijai – programinio kodo išskirstymas į komponentus leidžia lanksčiai ir dinamiškai naudoti ir pernaudoti jau sukurtas kodų dalis (su jos teikiamomis galimybėmis susipažinta atliekant praktiką UAB „Exadel“ įmonėje). Tuo pačiu dinamiškas tų pačių komponentų atnaujinimas sudaro galimybę sklandžiai ir dinamiškai žaidimo eigai.
2. Pasirinkta platforma, tai yra internetinė svetainė, leidžia patogiai bet kur, bet kada ir su bet kokia priemone, turinčią interneto prieigą, pasiekti žaidimą. Tai pritraukia daugiau žmonių išbandyti šį žaidimą, nes nereikia nieko įdiegti, ir tai populiarina žaidimą bei plečia žaidėjų gretas. Ateityje galima svetainės plėtra pridėdant forumą svetainėje, kas skatintų žaidėjus būti aktyviais, diskutuoti apie žaidimą ar galimas žaidimo strategijas, kas dar labiau populiarintų žaidimą, bei suteiktų potencialios papildomos informacijos žaidimo dirbtiniam intelektui kurti.
3. Žaidimo perkėlimas į virtualią erdvę turi teigiamų pranašumų: žaidimas tampa prieinamesnis visiems žmonėms, nesvarbu kur jie yra, yra pašalinama žmogiškos klaidos tikimybė, kai žaidėjas netyčia arba tyčia atlieka negalimą pagal taisyklės ėjimą su kauliuku. Ateityje galima įdiegti žaidimo eigos peržiūrą, ėjimo grąžinimą, išsaugoti esamo žaidimo būseną ateičiai.
4. Kaip testavimas parodė, šis žaidimas tinkamas įvairioms amžiaus grupėms – tiek jaunimui, tiek pagyvenusiems žmonėms. Daugybė galimų pasirinkimų žaidimo eigoje daro jo rezultatus sunkiai nuspėjamus, o tai įkvepia naujiems bandymams, t. y. naujoms žaidimo partijoms.

Papildomos svetainės funkcijos galėtų būti žaidėjų reitingavimas pagal kiekvieno jų laimėtų partijų skaičių. Taip pat būtų galima dvimatį (2D) figūrėlių atvaizdavimą pakeisti trimačiu (3D), nes svarbu vienu metu matyti ir šonines matematines kiekvienos figūrėlių reikšmes (laukiantis iššūkis: kai tai atvaizduoti, jei viena figūrėlė uždengia kitą). Šiuo metu šoninių plokštumų reikšmės žaidėjas gali išvysti šoniniame lange, esančiame šalia žaidimo lentos, paspaudęs ant pačios figūrėlės.

Literatūros sąrašas

- [1] AGON releases new chess player statistics from YouGov, 2012 August 09, Page 1 https://www.fide.com/images/stories/NEWS_2012/FIDE/120806_YouGovPressRelease.pdf
- [2] Wikipedia. *Computer chess*. https://en.wikipedia.org/wiki/Computer_chess
- [3] Edwards, D. Jerri, Xu, Huiping, Clark, O. Daniel, etc. Speed of processing training results in lower risk of dementia. [in:] *Alzheimer's & Dementia: Translational Research & Clinical Interventions (TRCI)*, 2017 November, Volume 3, Issue 4, Pages 603–611 [žiūrėta 2019 m. rugpjūčio 11 d.] [https://www.trci.alzdem.com/article/S2352-8737\(17\)30059-8/fulltext](https://www.trci.alzdem.com/article/S2352-8737(17)30059-8/fulltext)
- [4] „Youtube“ svetainės filmukas „*Dice King Chess ~Dice abstract game~*“, rodantis kaip žaidimas yra žaidžiamas https://www.youtube.com/watch?v=ZZ4_y1qmbgY
- [5] „Dice King Chess“ žaidimo aprašymas stalo žaidimų puslapyje <https://boardgames.com/boardgame/yyyyyyyyyy-dice-king-chess>
- [6] React: A JavaScript library for building user interfaces [paskutinį kartą žiūrėta 2019 m. gruodžio 5 d.]: <https://reactjs.org/>
- [7] Tutorial: intro to React (*Vadovas: įvadas į React programą*) [žiūrėta 2019 m. spalio 12 d.] : <https://reactjs.org/tutorial/tutorial.html>
- [8] Systems and software engineering – Vocabulary. Iso/Iec/IEEE 24765:2010(E). 2010-12-01. pp. 1–418.
- [9] Nielsen, J. (1994). Usability Engineering, Academic Press Inc, p 165