



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

Bakalauro baigiamasis darbas

Besimokančiojo išivertinimas taikant kompiuterinius žaidimus

Atliko:

Lukas Rakauskas

parašas

Vadovas:

J. Asist. Oleg Mirzianov

Vilnius
2019

Turinys

Santrauka	3
Summary	4
Iyadas	5
1. Kompiuterinių žaidimų psichologija ir esamų įrankių bei tyrimų apžvalga	6
1.1. Kompiuterinių žaidimų naudotojo psichologija	6
1.2. Esamų įrankių apžvalga	7
1.2.1. Bendriniai mokomųjų kompiuterinių žaidimų pavyzdžiai ir programavimo žaidimų kategorijos	7
1.2.2. CodinGame	8
1.2.3. CodeHunt	9
1.2.4. CodeCombat	10
1.2.5. Leek Wars	10
1.2.6. Apžvelgtų įrankių apibendrinimas	11
1.3. Esamų tyrimų apžvalga	11
1.4. Skyriaus apibendrinimas	13
2. Sukurtos programos apžvalga	15
2.1. Programos bendrinis aprašymas ir siekiniai	15
2.2. Naudojami ir alternatyvūs įrankiai	16
2.2.1. Unreal Engine 4	16
2.2.2. Visual Basic .NET	16
2.2.3. Unity	17
2.2.4. Blender	17
2.2.5. Adobe Photoshop	17
2.3. Sukurto įrankio apžvalga ir iškeltų siekinių realizacijos vertinimas	17
3. Atlikto tyrimo apžvalga	20
3.1. Tyrimo bendrinė informacija	20
3.2. Tyrimo pristatymas	20
3.2.1. Kompiuterinio programavimo žaidimo efektyvumas ir žaidėjų motyvacija . .	21
3.2.2. Kompiuterinio programavimo žaidimo naudingumas ir vizualizacijos svarba .	22
3.3. Tyrimo rezultatų apibendrinimas ir palyginimas su jau atliktais tyrimais	24
Išvados ir rekomendacijos	26
Ateities tyrimų planas	27
Literatūros šaltiniai	28
Priedai	30
A. Apklausos atsakymai	31

Santrauka

Kompiuteriniai žaidimai vis labiau yra įtraukiami į kasdienį žmonių gyvenimą, jie naudojami pramogoms, simuliacijoms, reklamai ir mokymui. [1] Būtent švietimo srityje naudojami kompiuteriniai programavimo žaidimai yra šiame bakalauro baigiamajame darbe. Pagal Fogo [7] elgsenos modelį žaidimo sėkmingumą nulemia tai, kaip efektyviai žaidimas geba kontroliuoti žaidėjo motyvaciją, žinių lygį ir pateikti skatinamuosius veiksnius. Šiame rašto darbe analizuojant jau esamus kompiuterinius programavimo žaidimus arba sužaidybintas mokymosi platformas, buvo pastebėta, kad didžioji dalis įrankių susitelkia būtent į žaidėjų motyvaciją, kuri yra skatinama apdovanojant žaidėjus patirties taškais, pasiekimų ženkleliais ir reitinguojant žaidėjus pagal jų rezultatus. Taip pat buvo nagrinėjami ir apibendrinami jau atlikti tyrimai, kurie vertino kompiuterinių žaidimų efektyvumą, integruojant juos į studijų programas. Tyrimai parodė, kad žaidimai skatina studentus aktyviau dalyvauti dėstomuose moduluose, [2][15] taip pat yra pažymėtinas 10,8% padidėjęs studentų, pasiekiančių slenkstinį modulio įvertinimą, kiekis. [27] Pagrindinis šio bakalauro darbo tikslas buvo sukurti žaidybinius elementus naudojantį įrankį, kuris didintų C# programavimo kalbą studijuojančių asmenų mokymosi motyvaciją ir praplėstų jų savęs įsivertinimo galimybes. Kuriant programavimo žaidimą buvo atsižvelgiama į besimokančiojo savianalizę svarbius aspektus ir išnagrinėtų įrankių struktūrą. Siekiant įvertinti sukurto įrankio naudingumą studentų savitikros procese ir bendrą besimokančiųjų nuomonę apie kompiuterinius programavimo žaidimus, buvo atliktas sukurto žaidimo žaidėjus analizuojantis tyrimas. Įvertinus gautus tyrimo rezultatus, buvo pastebėta, kad tiriamoji imtis yra linkusi naudoti kompiuterinius programavimo žaidimus kaip papildomą mokymosi įrankį, leidžiantį pasitikrinti ir įsivertinti turimas žinias. Taip pat tyrimo rezultatai parodė, kad apklaustųjų nuomone, programinio kodo vizualizavimas žaidimo pavidalu padeda geriau suprasti programavimo kalbą. Pažymėtina yra ir tai, kad 89,47% apklaustųjų teigia, kad geriausiųjų žaidėjų lentelė padidintų jų motyvaciją progresuoti žaidime.

Apibendrinant šiame bakalauro darbe atliktą tyrimą ir jau esamus tyrimus kituose universitetuose [2][15], buvo pastebėta, kad kompiuterinių programavimo žaidimų naudojimas studentų edukacinėje veikloje yra vertinamas teigiamai, nepriklausomai nuo to, ar žaidimas yra tiesiogiai integruotas į dėstomą universitetinį modulį, ar pateikiamas kaip papildomas savitikros įrankis. Visgi, pažymėtina yra tai, kad tyrimo metu, apklausos dalyviai aktyviai išreiškė norą išbandyti sužaidybintą universitetinio lygio modulį. Ateities tyrimams ir programavimo žaidimų pritaikymams yra rekomenduojama atkreipti papildomą dėmesį į mažai motyvuotus ir minimaliai žaidime progresuojančius asmenis, nes įvardintais aspektais pasižymintys studentai, buvo linkę neigiamai vertinti kompiuterinių programavimo žaidimų naudą.

Summary

Student Self-assessment Using Computer Games

Computer games are becoming more involved in people's daily lives as they are used for entertainment, real life simulations, advertising and learning. [1] Programming games that are used in the area of education, their effectiveness and efficiency are analysed in this bachelor thesis. According to the Fogg's behavioral model [7], game success is determined by how effectively it is able to control the player's motivation, knowledge and triggers. While analysing games, that teaches users programming, or learning platforms that uses gamified elements, it has been observed that most of the tools focus specifically on players' motivation. The main motivational aspects used in educational games are players' experience points, achievements, badges and leader boards. Studies that have evaluated the effectiveness of computer games and their integration into the curriculum, was also observed and summarized. Studies have shown that the games encourage students to actively participate in the taught modules [2][15], there are also 10,8% increased amount of students that have been able to reach the threshold number of evaluation points. [27] During this bachelor work, a coding computer game, which supports C# programming language run-time compilation, was created. The purpose of this tool was to increase student's self-motivation and to expand their self-assessment capabilities. While developing the programming game, aspects relevant to the learner's self-analysis were considered and the structure of the already existing tools were examined. In order to evaluate the usefulness of the created tool in the student self-assessment process and the general learners opinion about computer programming games, an additional study was conducted. After evaluating the results of the study, it was observed that students tend to use computer programming games as an additional learning tool to test and evaluate their knowledge. Also, the survey results showed that students think that software visualization in the form of a game helps them to better understand the programming language. It is also noteworthy that 89.47% of the respondents say that leaderboard would increase their motivation to progress in the game.

Summarising this bachelor thesis study and existing researches in other universities [2][15], it has been observed that the use of coding games in educational activities are evaluated positively by the students, regardless of whether the game is directly integrated into the module or if it is used as a secondary tool. However, it is important to notice that the research participants expressed a desire to try the gamified university-level module. Future research, regarding coding games and applications are recommended to pay extra attention to the people with little motivation and minimal desire to progress in the game, because they tend to negatively evaluate the benefits of computer games and do not find this learning method useful.

Ivyadas

Pagal Oksfordo anglų kalbos aiškinamąjį žodyną¹ terminas „kompiuterinis žaidimas“ (ang. video game) yra apibūdinamas kaip „žaidimas, žaidžiamas elektroniniu būdu manipuliuojant kompiuterio programos vaizdus monitoriuje ar kitame ekrane“. Vienas pirmųjų kompiuterinių žaidimų yra Čarlso Adamso ir Džono Gilmore 1949 m. sukurtas „The Bouncing Ball Program“, kuris apibūdinamas kaip programa, kurioje simuliuojamas šokinėjantis kamuolys esantis dėžėje su skylė apačioje, o vartotojas manipuliuodamas ekrano vaizdus siekia pataikyti kamuolį į skylę. [19] Visgi per paskutinius šešis dešimtmečius kompiuteriniai žaidimai ypač pasikeitė ir tapo itin populiarūs. Dabar jie naudojami įvairiuose kasdienio gyvenimo sektoriuose, tokiuose kaip rinkodara, politika, menai, gynyba ar mokymas. [1] Būtent švietimo sektorius dažnai susilaukia visuomenės dėmesio ir yra nuolat stengiamasi tobulinti šią sritį. Tradiciniai mokymo metodai remiasi rašytinėmis knygomis ir žodinėmis paskaitomis, tačiau spartus technologinis vystymasis suteikia naujų galimybių interaktyviai mokytis patiems, pavyzdžiui, naudojant kompiuterinius žaidimus. [5][26] Šiame rašto darbe bus analizuojami įvairūs mokomieji kompiuteriniai žaidimai bei žaidybinius elementus pasitelkiančios programos. Mokomaisiais kompiuteriniais žaidimais bus laikomi įrankiai, kurie sujungia mokymo, tobulėjimo, saviugdų, komunikacijos ir informacijos aspektus kartu su pramoginiais laisvalaikio elementais ir naudoja kompiuterinių žaidimų technologijas bei vizualizaciją. [1] Mokymosi būdas, kai yra naudojami kompiuteriniai žaidimai, tarp daugelio pedagogų ir tyrėjų sukelia įvairias diskusijas, kurios bus vertinamos ir analizuojamos šiame rašto darbe. [3][16]

Šio bakalauro baigiamuoju darbu yra siekiama sukurti ir ištestuoti universitete įgyjamų žinių įsivertinimo įrankį, integravus sužaidybinių aspektus, motyvacijai mokytis padidinti. Iškeltam tikslui pasiekti, pirmiausia rašto darbe bus nagrinėjami svarbiausi sužaidybinių aspektai bei jų realizaciją jau esamuose mokymosi įrankiuose. Atsižvelgiant į sėkmingam mokymuisi bei savianalizei svarbius aspektus ir išnagrinėtų įrankių struktūra, rašto darbe yra iškeliami reikalavimai kuriamam programavimo žaidimui. Taip pat, siekiant įvertinti sukurto įrankio naudingumą studentų savitikros procese, šiame rašto darbe yra aprašytas atliktas tyrimas, kuris analizavo sukurtą žaidimą žaidusių asmenų atsiliepimus. Tyrimo metu buvo siekiama ne tik įvertinti sukurto įrankio svarbą besimokančiojo asmens savitikros procese, tačiau ir įvertinti bendrą studentų nuomonę apie kompiuterinius programavimo žaidimus. Rašto darbo pabaigoje pateikiamos išvados apie analizuotus įrankius, tyrimus bei svarbiausius aspektus, kuriant mokomąjį kompiuterinį žaidimą. Taip pat yra pateikiamos rekomendacijos ateities tyrimams ir sukurto įrankio pritaikymui bei plečiamumui.

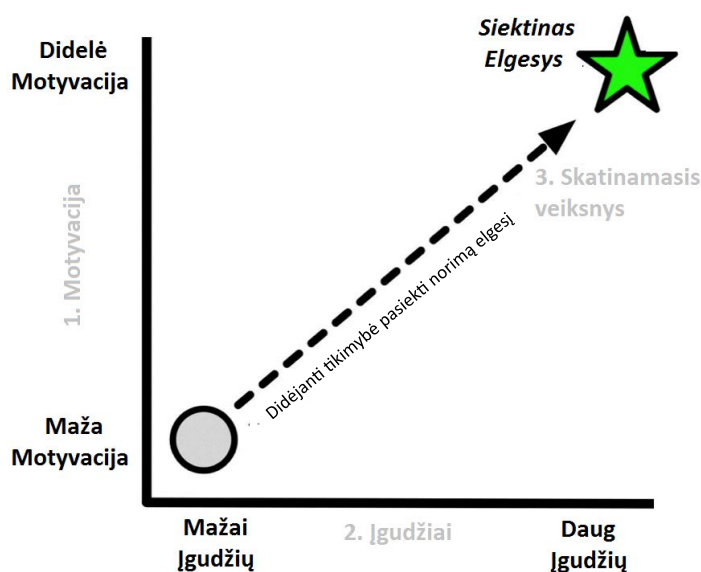
¹Oxford English Dictionary, <http://www.oed.com>

1. Kompiuterinių žaidimų psichologija ir esamų įrankių bei tyrimų apžvalga

Šiame skyriuje aprašomi pagrindiniai žmogaus psichologiniai faktoriai, turintys įtaką kompiuterinio žaidimo patrauklumui. Taip pat pristatomi ir analizuojami jau sukurti programiniai įrankiai bei atlikti tyrimai. Skyriaus pabaigoje apibendrinamos visos minėtos sistemos ir išskiriami pagrindiniai reikalavimai sėkmingam mokomajam kompiuteriniam žaidimui.

1.1. Kompiuterinių žaidimų naudotojo psichologija

Didžioji dalis kompiuterinių žaidimų, nepriklausomai ar jie yra mokomojo pobūdžio ar ne, siekia kontroliuoti konkrečias vartotojo emocijas, įsisavinamą informaciją bei naudoja įvairius veiksnius norimai vartotojo elgsenai pasiekti. Siekiant sukurti efektyvų mokomąjį kompiuterinį žaidimą ir bent minimaliai kontroliuoti vartotojo elgseną bei emocijas, reikia suprasti pagrindinius žmogaus psichologijos dėsnius žaidžiant žaidimus. Fogo elgsenos modelis (ang. Fogg's Behaviour Model) teigia, kad yra 3 pagrindiniai faktoriai turintys įtaką vartotojo elgsenai: motyvacija, įgūdžiai ir skatinamasis veiksnys (trigeris). [7] Fogas pateikia diagramą (1 pav.) kuri iliustruoja, kad norint pasiekti tam tikrą vartotojo elgseną reikalingas pakankamas subjekto motyvacijos ir įgūdžių lygis bei efektyvus skatinamasis veiksnys. Teisingai sukurtas mokomasis kompiuterinis žaidimas turi kontroliuoti vartotojo motyvaciją pasitelkiant apdovanojimus, o įgūdžių lygį - naudojant ankstesnę praktiką ir paskatinimus. [15] Esant teisingam motyvacijos ir įgūdžių balansui žaidėjas intuityviai sieks progresuoti žaidime, tad šioje situacijoje yra reikalingas papildomas trigeris, kuris tik padidintų vartotojo norą progresuoti. Kompiuterinio žaidimo kontekste skatinamasis veiksnys gali būti ir paprastas informacinis pranešimas teigiantis, kad vartotojas išsprendęs esamą problemą įgis naują apdovanojimą. Visgi, pasiekti tobulą balansą tarp minėtų trijų aspektų yra sudėtinga ir tam reikalingas nuolatinis sistemos vartotojų stebėjimas ir įrankio koregavimas, kur trigeris yra svarbiausias faktorius. Ne laiku atsiradęs skatinamasis veiksnys gali priversti subjektą jaustis nusivylusį ir nemotyvuotą.[7]



1 pav. Fogo elgsenos modelis

Kadangi šio bakalauro darbo metu kuriamas įrankis bus mokamojo pobūdžio, todėl taip pat svarbu išanalizuoti savęs įsivertinimo svarbą asmens mokymosi procese. Šiame rašto darbe savęs įsivertinimu bus laikomas procesas, kurio metu besimokantis asmuo stebi ir įvertina savo mąstymo kokybę bei identifikuoja strategijas, kurios gali pagerinti besimokančiojo įgūdžius ir temos supratimą. [14] Teisingai realizuotas besimokančiojo savęs įsivertinimo procesas gali didinti asmens vidinę motyvaciją, įdedamas pastangas ir norą tobulėti užsibrėžto tikslo link. [14] Savęs įsivertinimas taip pat yra vienas iš esminių elementų kognityvinio ir konstruktyvinio mokymosi teorijose. [24] Atlikti tyrimai [21][22] rodo, kad studentų aktyvumas ir mokymosi efektyvumas priklauso nuo besimokančiųjų asmenų savianalizės ir asmeninių įsitikinimų apie jų gebėjimą gerai atlikti tam tikrą užduotį. Taip pat besimokantis asmuo dažniausiai yra linkęs įvertinti naudą, kurią jis įgis išmokęs naujų įgūdžių. [22]

Apibendrinant šį poskyrį, yra pažymėtina tai, kad mokomojo pobūdžio kompiuteriniai žaidimai yra naudingi besimokančiojo asmens įsivertinimo procese, kuris yra svarbus aspektas didinant asmens vidinę motyvaciją ir norą siekti užsibrėžto tikslo. [14] Taip pat efektyvus savianalizės procesas gali gerinti asmens aktyvumą ir mokymosi kokybę. [22] Siekiant sukurti naudingą mokomąjį kompiuterinį žaidimą, kuris padėtų besimokantiems asmenims įsivertinti turimas žinias, reikia atkreipti dėmesį į 3 pagrindinius faktorius, turinčius įtaką vartotojo elgsenai žaidimo metu: motyvacija, įgūdžiai ir skatinamasis veiksnys. Pagal Fogo [7] elgsenos modelį, žaidimo metu yra svarbu, kad asmuo turėtų pakankamą motyvacijos ir žinių lygį, bei laiku pateikti papildomą skatinamąjį veiksnį, ir tik tada yra galima tikėtis siektino žaidėjo elgesio.

1.2. Esamų įrankių apžvalga

Šio poskyrio tikslai yra:

- 1) Įvertinti ar jau sukurti programavimo žaidimai turi pasikartojančią vartotojo sąsają, kurią būtų galima pritaikyti kuriant naują programavimo žaidimą.
- 2) Išskirti jau sukurtuose įrankiuose dažniausiai naudojamus motyvacinius aspektus, kurie būtų pritaikomi kuriant naują programavimo žaidimą.

Analizuojamiems įrankiams (1.2.2 - 1.2.5 poskyriai) buvo keliami du reikalavimai:

- 1) Žaidėjo turi būti reikalaujama naudoti bent vieną programavimo kalbą. Šio reikalavimo pagalba buvo išfiltruoti įrankiai, kurie moko tik bendrinių programavimo pagrindų.
- 2) Įrankį turi būti naudoję bent 50 tūkst. unikalių vartotojų. Pasirinktas minimalus naudotojų skaičius yra sąlyginai atsitiktinis, tačiau taip siekiama vertinti tik tuos įrankius, kurie turi pakankamą pasisekimą programavimo žaidimų tarpe. Šio reikalavimo pagalba, skyriaus pabaigoje atliekamos išvados bus laikomos kaip realiai pritaikomos ir efektyvios, nes bus remiamasi tik sėkmingais ir populiariais įrankiais.

1.2.1. Bendriniai mokomųjų kompiuterinių žaidimų pavyzdžiai ir programavimo žaidimų kategorijos

Mokomuosius kompiuterinius žaidimus galima apibūdinti kaip programą susidedančią iš trijų pagrindinių aspektų: multimedijos, patirties ir pramogos. [13] Alvarez savo knygoje [1] pastebi, kad kompiuteriniai žaidimai gali būti naudojami įvairiose gyvenimo sferose. Pavyzdžiui, Jungtinių Amerikos Valstijų kariuomenė 2002 m. pristatė kompiuterinį žaidimą „America's Army”², kuris

²America's Army, <https://www.americarmy.com>

jau 2004 m. buvo parsisiųstas daugiau nei 17 mln. kartų [1]. Žaidimas skyrėsi nuo kitų, tuometinių karinių žaidimų tuo, kad šis simuliuo itin realistišką karinį gyvenimą, o geriausi žaidėjai gaudavo oficialius kvietimus į JAV kariuomenę. Žaidimas taip pat buvo naudojamas apmokyti naujiems karininkams ir supažindinti juos su specifinėmis situacijomis kritinėse situacijose. Kitas sektorius, kuriame plačiai naudojami kompiuteriniai mokomieji žaidimai, yra medicina. Pavyzdžiui, Nintendo³ įmonės sukurtas žaidimas „Dr Kawashima’s Brain Training: How Old Is Your Brain“ įvertina naudotojo smegenų amžių, pateikiant įvairius uždavinius žaidimo forma. Taip pat yra ne vienas žaidimas simuliuojantis tikras medicininės situacijas. Vital Signs⁴, 3DiTeams⁵, Clinispace⁶ buvo sukurti specialiai apmokyti įvairias medikų komandas. [9] Visi minėti įrankiai atitinka Laamarto aprašytus mokomojo žaidimo reikalavimus: vartotojai įgyja patirties ir sužino papildomos informacijos susijusios su žaidimo tematika, žaidimai yra įtraukiantys ir naudoja vizualinius bei garsinius efektus informacijai pateikti.

Visgi, šiame rašto darbe bus analizuojami tik su informacinėmis technologijomis ir programavimu susiję įrankiai. Kompiuterinius programavimo žaidimus galima suskirstyti į tris kategorijas pagal jų tikslus: mokymasis programuoti, mokymasis kurti žaidimus ir algoritminio mąstymo mokymasis. [4] Keletas skirtingų internetinių platformų mokančių naudotojų programavimo kalbos ir algoritminio mąstymo jau yra sukurta ir būtent šių kategorijų žaidimai bus analizuojami šiame poskyryje.

1.2.2. CodinGame

„CodinGame“⁷ - tai 1 320 731⁸ unikalių vartotojų turinti platforma, kurioje vartotojas turi įveikti įvairius žaidimo lygius išspręsdamas vis kitas programavimo užduotis. Sistema turi pasikartojančią žaidybines rutinas - vartotojas yra supažindinamas su esama problema, pateikiama specifikacija ir reikalavimai kuriamam scenarijui, tuomet vartotojas turi pritaikyti jau turimas programavimo kalbos žinias ir išspręsti iškilusią problemą.

Įrankis palaiko virš 20 populiariausių programavimo kalbų. Žaidėjas kiekvieno žaidimo lygio pradžioje gali keisti programavimo kalbą, kurią jis nori naudoti. Esant vartotojo norui, šis funkcionalumas leidžia kiekvieną žaidimo lygį įveikti naudojant vis kitą programavimo kalbą.

Sistema motyvuoja vartotojus skirdama patirties taškus ir pasiekimų ženklukus, kurie yra matomi visiems sistemos vartotojams. Visi vartotojai yra viešai reitinguojami pagal savo pasiekimus. Taip pat žaidimo lygiai yra grupuojami pagal jų sudėtingumą, o kaip skatinamasis veiksnys naudojama žaidybinių vizualizacija. Įvairi informacija, kuri atvaizduoja, kad vartotojas yra beveik įveikęs žaidžiamą lygį, skatina žaidėją progresuoti toliau. „CodinGame“ įrankis sėkmingai įgalina 1.1 poskyryje minimą Fogo elgsenos modelį išlaikydamas vartotojo motyvaciją, kontroliuodamas reikalingą įgūdžių lygį, bei laiku pateikdamas skatinamuosius veiksmus.

2 paveikslėlio A dalyje atvaizduotas vartotojo matomas langas, kuris išlaiko tą pačią struktūrą viso žaidimo metu. Savo sprendimo vizualizaciją vartotojas mato tiek žaidimo, tiek komandinės eilutės pavidalu. Tokio tipo sistema stimuliuoja vartotojo algoritminį mąstymą, tačiau tiesiogiai neskatina mokytis programavimo kalbos ir pagal programavimo žaidimų kategorijas būtų priskirta „algoritminio mąstymo mokymosi“ kategorijai.[4]

³Nintendo www.nintendo.co.uk

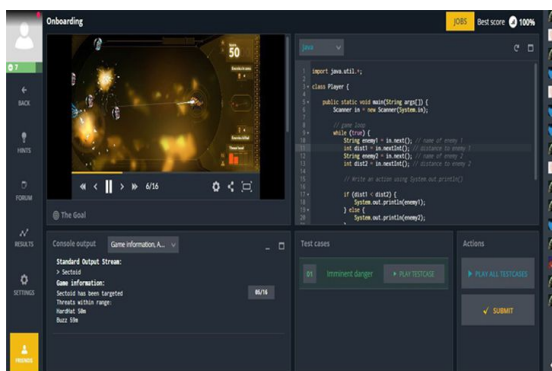
⁴Vital Signs, <http://www.breakawaygames.com/vitalsigns/>

⁵3DiTeams, https://anesthesiology.duke.edu/?page_id=825623

⁶Clinispace - Virtual sim center, <http://virtualsimcenter.clinispace.com>

⁷CodinGame, <https://www.codingame.com>

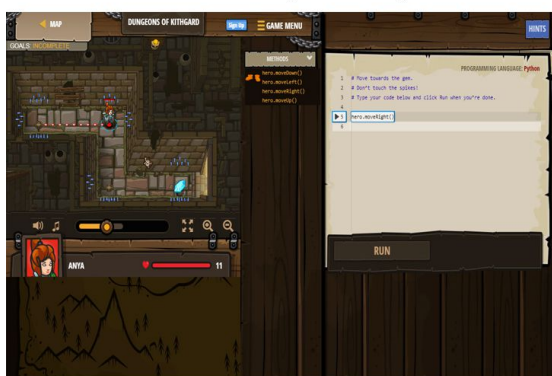
⁸Informacija tikrinta 2019 m. gruodį, <https://www.codingame.com/leaderboards/global>



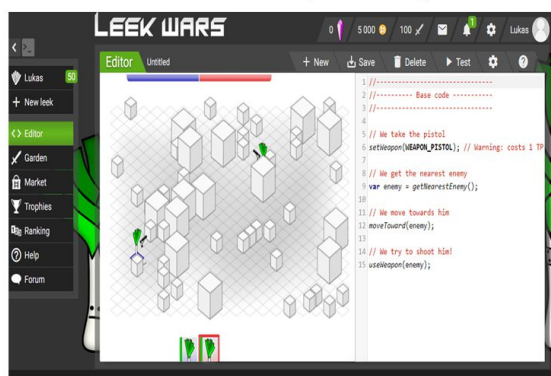
A. CodinGame vartotojo matomas langas



B. CodeHunt vartotojo matomas langas



C. CodeCombat vartotojo matomas langas



D. Leek Wars vartotojo matomas langas

2 pav. Apžvelgtų įrankių vartotojų langai

1.2.3. CodeHunt

„CodeHunt“⁹ - tai Microsoft Research įmonės sukurtas įrankis, kuris kitaip nei 1.2.2 poskyryje minėtas „CodinGame“, vartotojui pateikia tik uždavinio testus pagal kuriuos vartotojas turi nuspėti ką kuriamas scenarijus turi daryti ir tik tada „sutvarkyti“ pateiktą programos kodą.

Žaidimas leidžia vartotojams rinktis tarp Java ir C# programavimo kalbų. Šį įrankį 2016 m. duomenimis¹⁰ buvo atsisiuntę daugiau nei 350 tūkst. vartotojų. Įrankio populiarumą galima sieti su žaidime naudojamų programavimo kalbų populiarumu tarptautiniame rinkoje. Pagal Tiobe¹¹ programavimo kalbų reitingą, 2019 m. sausį Java programavimo kalba buvo pirmoje, o C# septintoje vietoje (2016 m. Java - pirmoje, C# - ketvirtoje vietoje).

Vartotojui išsprendus lygį, jis yra apdovanojimas tam tikra taškų suma, kuri yra naudojama visų sistemos naudotojų reitingavimui, kas suteikia žaidėjams motyvacijos progresuoti žaidime. [4] Taip pat, siekiant išlaikyti vartotojo motyvaciją sudėtinguose lygiuose bei kompensuoti reikiamų žinių trūkumą, programa įvertinus vartotojo įvestą programinį kodą gali pažymėti kurias programinio kodo eilutes pakeitus būtų pasiektas uždavinio sprendimas.

„CodeHunt“ išsiskiria iš kitų, šiame skyriuje apžvelgiamų, įrankių savo vartotojo sąsaja, kuri yra matoma 2 paveikslėlio B dalyje. Įrankis tiesiogiai neatvaizduoja vartotojo kodo žaidimo pavidalu, tačiau naudoja žaidybinius elementus vartotojų motyvacijai gerinti. Visgi, vartotojo matomas ekrano langas yra padalintas į du polangius - scenarijaus kūrimo (kairėje) ir žaidimo lygio testų atvaizdavimo (dešinėje). Nors įrankis ir nenaudoja akivaizdžių žaidybinių elementų žaidėjo programinio kodo vizualizacijai, tačiau vartotojo sąsaja yra pakankamai panaši į visų, šiame skyriuje

⁹CodeHunt, <https://www.codehunt.com>

¹⁰Informacija tikrinta 2019 m. sausį, <https://www.codehunt.com>

¹¹Tiobe (duomenys paskutinį kartą tikrinti 2019 m. sausį, <https://www.tiobe.com/tiobe-index/>)

apžvelgiamų, įrankių vartotojo sąsajas.

1.2.4. CodeCombat

„CodeCombat“¹² - tai nuotykių mokomasis kompiuterinis programavimo žaidimas ir mokymo programa, kuri leidžia mokytis algoritminio mąstymo ir programavimo kalbos pagrindų. 2019 m. sausio m. duomenimis¹³ nuo 2013 m. ši įrankį naudojo daugiau kaip 5 mln. unikalių vartotojų. Žaidėjai gali rinktis iš 11 skirtingų mokymosi kursų: 6 informacinių technologijų, 3 žaidimų kūrimo ir 2 puslapių kūrimo kursai. Nepriklausomai nuo pasirinkto kurso, žaidimo tematika išlieka tokia pati - žaidėjas kiekviename lygyje valdo savo veikėją ir patenka į labirintą, kur sutinka vis naują, priešiška nusiteikusia būtybę, kurią reikia įveikti, kuriant įvairius scenarijus. Pasikartojanti žaidimo lygių struktūra leidžia žaidėjui susikonsultuoti į sprendžiamą uždavinį. Žaidėjas turi pasirinkti teisingą veiksmų seką, kad įveiktų žaidimo lygį, nes logikai prieštaraujančios veiksmų sekos, pavyzdžiui, komanda nurodanti veikėjui judėti į kairę, kur iš tiesų yra kliūtis, atima iš žaidėjo jo turimas gyvybes kol pagaliau žaidėjo valdomas veikėjas miršta.

Žaidėjai gali rinktis tarp Python ir JavaScript programavimo kalbų. Visgi, įrankio kūrėjai neatmeta galimybės¹⁴ ateityje integruoti C, C++, Java programavimo kalbas į jau esamą sistemą. Žaidimas yra paremtas „Rurple“ ir „Karel“ programavimo įrankiais, kurie skirti jaunesnio amžiaus vaikams mokytis algoritminio mąstymo pagrindų.

„CodeCombat“, kitaip nei 1.2.2 poskyryje minėtas „CodinGame“, didesnis dėmesys yra skiriamas žaidybiniais elementams ir vizualiniams efektams nei programavimo uždavinių sudėtingumui. Žaidėjai yra motyvuojami skiriant patirties taškui už sėkmingai įveiktus labirintus. Taip pat žaidėjai už tam tikrus pasiekimus yra apdovanojami žaidimo metu naudojamais objektais, kurie padeda naudotojui lengviau progresuoti žaidime.

„CodeCombat“ žaidimo langas yra padalintas į dvi dalis (2 paveikslėlio C dalis), kairėje matomas labirintas, veikėjas, priešininkai ir kiti žaidybiniai elementai, o dešinėje vartotojas gali kurti norimą programos scenarijų. Ši vartotojo sąsajos struktūra yra išlaikoma viso žaidimo metu.

1.2.5. Leek Wars

„Leek Wars“¹⁵ - tai 64 893¹⁶ unikalius vartotojus turintis kompiuterinis programavimo žaidimas, kuriame žaidėjai turi sukurti savo valdomo personažo elgseną. Pagrindinis tikslas yra nugalėti kitus priešininkus mūšio lauke. Priklausomai nuo žaidėjo lygio, jis gali naudoti papildomus ginklus, gyvybes gražinančius objektus ir pan.

Žaidimo personažai yra programuojami JavaScript programavimo kalba ir papildoma programų sąsaja (ang. API), kuri suteikia papildomo funkcionalumo valdomam veikėjui. Pateikta programų sąsaja yra detalai dokumentuota, kas moko sistemos naudotojus naudotis sistemų specifikacijomis. [4]

Žaidimas motyvuoja vartotojus apdovanodamas juos patirties taškais už kiekvieną įveiktą priešininką. Patirties taškai yra naudojami gauti naujiems objektams, kurie gali būti panaudoti mūšio lauke. Taip pat pagal žaidėjų turimus patirties taškus, yra sudaroma geriausių žaidėjų lentelė, kuri skatina naudotojus, kuo geriau išstobulinti savo sukurtą scenarijų ir užimti kiek įmanoma

¹²CodeCombat, <https://codecombat.com>

¹³Informacija tikrinta 2019 m. sausį, <https://codecombat.com/about>

¹⁴<https://help.codecombat.com/frequently-asked-questions/what-programming-languages-does-codecombat-use>

¹⁵LeekWars, <https://leekwars.com>

¹⁶Informacija tikrinta 2019 m. sausį, <https://leekwars.com>

aukštesnę poziciją geriausiųjų sąrašė.

Vartotojo lango struktūra yra matoma 2 paveikslėlio D dalyje. Langas yra padalintas į du pagrindinius polangius. Vartotojas dešiniajame polangyje gali kurti norimą scenarijų savo personažui, o kairiame polangyje gali matyti savo personažo elgseną mūšio lauke. Vartotojo sąsajos išdėstymas yra panašus į 1.2.2 ir 1.2.4 poskyriuose aprašytus įrankius, kas leidžia manyti, kad toks vartotojo sąsajos išdėstymas yra vienas efektyviausių ir labiausiai priimtinių vartotojui.

1.2.6. Apžvelgtų įrankių apibendrinimas

Vertinant šiame skyriuje apžvelgtus įrankius, buvo pastebėta pasikartojanti vartotojo sąsajos struktūra (visų įrankių vartotojų matomi langai pateikti 2 paveikslėlyje). Apžvelgti įrankiai vartotojo matomą langą dalina į du pagrindinius polangius - scenarijaus kūrimo ir žaidimo vizualizacijos. Vienintelis „CodeHunt“ įrankis (1.2.3 poskyris) neturėjo tiesioginio vartotojo kodo vizualizavimo žaidimo pavidalu ir naudojo žaidybinius elementus tik vartotojo motyvacijai gerinti. Visi analizuoti įrankiai motyvuoja vartotojus skirdami patirties taškus už sėkmingai įveiktus lygius. Patirties taškai dažniausiai yra naudojami reitinguojant visus žaidėjus, kas taip pat motyvuoja naudotojus progresuoti žaidime ir užimti kuo aukštesnę poziciją geriausiųjų žaidėjų sąrašė. Papildomai žaidėjai gali būti apdovanojami žaidimo metu naudojamais objektais (1.2.4 ir 1.2.5 poskyriai). Naudotojų žinių lygis yra kontroliuojamas grupuojant žaidimo lygius pagal sudėtingumą („Coding Game“ - 1.2.2 poskyris) arba patariant žaidėjams kurias sukurto scenarijaus eilutes pakeitus, būtų įveiktas žaidžiamas lygis („CodeHunt“ - 1.2.3 poskyris).

1.3. Esamų tyrimų apžvalga

Įvairūs mokslininkai ir tyrimai parodė, kodėl kompiuteriniai žaidimai gali būti naudojami kaip mokymosi įrankis. Žaidimai yra motyvuojantys, įtraukiantys ir suteikia aiškų grįžtamąjį ryšį vartotojui. [6][20][25] Taip pat žaidimai puikiai tinka mokymui, nes visas dėmesys yra skiriamas vartotojui. [8]

Arto Vihavainen ir Jonne Airaksinen savo literatūros apžvalgoje [27] išanalizavo 226 mokslinius straipsnius, kurie aprašė įvairius mokymo metodus bei veiksmus, kurie turi įtakos dėstant informacinių technologijų kursus pradedantiesiems (ang. Computer science 1 (CS1)). Vihavainen atrinko mokslinius straipsnius pagal naudojamus raktinius žodžius, individualiai juos peržiūrėjo ir sugrupavo. Išanalizavus visą turimą informaciją autorius pateikė statistiką rodančią, kad po įvairių mokymo intervencijų, studentų, išlaikančių bent minimalius dėstomo kurso įvertinimo reikalavimus, kiekis padidėjo beveik trečdaliu lyginant su tradiciniais mokymo metodais. Iš visų analizuotų straipsnių ir mokymo metodikų, 7 naudojo kompiuterinius žaidimus ir (arba) sužaidybimo elementus dėstant informacinių technologijų kursus. Dėstymo metodikų, naudojančių kompiuterinius žaidimus, analizė parodė, kad slenkstinį įvertinimą vidutiniškai pasiekdavo 10,8% daugiau studentų.

Portugalijoje, Lisabonos universiteto informacinių sistemų ir kompiuterinės inžinerijos padalinyje, 2013 m. buvo atliktas, kurio tikslas buvo išsiaiškinti kokią įtaką mokomųjų kompiuterinių žaidimų panaudojimas turi dėstomo kurso kokybei. [2] Tyrimo metu buvo analizuojamas „Multimedijos turinio kūrimo“ (ang. Multimedia Content Production) modulis, kuris buvo stebimas du metus. Pirmaisiais metais studentai mokėsi pagal standartinę šio kurso programą, o antraisiais metais nauja studentų grupė mokėsi pagal sužaidybintą šio kurso versiją. Žaidybinė dėstomo modulio versija naudojo modifikuotą Moodle mokymosi sistemą, kuri gebėjo skaičiuoti studentų patirties

taškus, pasiektus lygius, sudaryti geriausiųjų studentų sąrašą ir skirti virtualius pasiekimų ženklelius. Tyrimo metu buvo naudojama Šeldono knygoje [23] aprašyta metodika, kuomet studentai pradeda turėdami mažiausią įmanomą įvertinimą ir kurso metu už kiekvieną įvykdytą uždavinį gauna tam tikrą taškų kiekį, kurie kurso pabaigoje konvertuojami į atitinkamą pažymį. Naudojant minėtą metodiką studentai yra skatinami mokytis iš savo klaidų, o nėra baudžiami už padarytas klaidas. [2] Tyrimo metu buvo stebimas studentų aktyvumas Moodle sistemoje, paskaitų lankymas, pažymiai ir dėstomo kurso mokomosios medžiagos parsisiuntimo kiekis. Atliktas tyrimas parodė, kad mokomoji medžiaga buvo parsisiūsta 47,7% daugiau, o aktyvumas Moodle sistemoje padidėjo net 845%. Visgi studentų aktyvumas internetinėje mokymo sistemoje yra siejamas su tuo, kad studentams buvo skiriami papildomi patirties taškai už aktyvumą (maksimaliai 10% galutinio balo). Paskaitų lankymas pirmaisiais metais buvo 81%, o antraisiais 92%, kas rodo 11% padidėjimą (pagal Mann-Whitney's U testą, $U = 51$, $Z = -3.654$, $p < 0.001$). Vertinant studentų pažymius nebuvo išvelgta pastebimo pokyčio, todėl autoriai rekomenduoja papildomus tyrimus šiam aspektui ištirti. Tyrimo metu buvo padarytos išvados, kad kompiuterinių mokomųjų žaidimų naudojimas skatina studentus skirti daugiau dėmesio dėstomam moduliui, studentai aktyviau lanko dėstomas paskaitas ir siunčiasi metodinę medžiagą. Autoriai taip pat pažymi, kad labai svarbus yra kompiuterinio žaidimo dizainas ir pasirinkti motyvaciniai aspektai. Tyrimo dalyviai geriausia atsiliepė apie uždavinius, kurie reikalavo ieškoti papildomos informacijos internete, aktyviai lankyti paskaitas ir kurti naujus produktus. Visgi uždaviniai, kurie reikalavo surasti rašybos klaidas arba programos veikimo klaidas, buvo vertinami kaip neįdomūs ir bereikšmiai.

Kitas aktualus tyrimas buvo atliktas Keiptauno universitete, Pietų Afrikos Respublikoje. [15] Mokomasis kompiuterinis žaidimas buvo pritaikytas vienam iš universitete dėstomų modulių. Sukurtas žaidimas rėmėsi 1.1 poskyryje minimu Fogo elgsenos modeliu siekiant išlaikyti vartotojo motyvaciją, įgūdžius ir skatinamuosius veiksmus reikiamam lygyje. Šio kurso sužaidybinimas naudojo vientisą istoriją, kurioje studentai siekė surasti „dingusį“ Babbado variklį (ang. Babbage Engine¹⁷). Studentai už kiekvieną išspręstą uždavinį gaudavo po užuominą kol galiausia surasdavo dingusį įrankį. Autoriai pasirinko žaidimą su konkrečiu siužetu, nes tai turėtų intriguoti ir motyvuoti studentus. [15] Taip pat studentai galėjo rinkti patirties taškus, virtualius ženkliukus bei matyti geriausiųjų studentų sąrašą. Pažymėtina, kad be pakankamai tradicinių motyvavimo būdų, autoriai taip pat įgalino ir virtualią valiutą, kurią studentai galėjo naudoti įvairiems aspektams, susijusiems su dėstomu moduliui, pirkti. Studentai galėjo leisti virtualius pinigus pirkdami papildomus bandymus atlikti testus, uždavinių užuominas bei atsiskaitymų datų pratęsimus. Autorių teigimu, taip buvo kontroliuojamas vartotojo įgūdžių lygis, padedant asmenims, kuriems sunkiau sekėsi įveikti įvairius uždavinius. Kompiuterinių žaidimų efektyvumas buvo vertinamas pagal studentų užpildytas apklausas, galutinius įvertinimus bei paskaitų lankymą. Apklausos respondentai kaip efektyviausią motyvacinį aspektą išskyrė geriausiųjų studentų sąrašą ir virtualią valiutą. Vidutiniškai 79,1% studentų atvykdavo į paskaitas, ką autoriai vertina kaip teigiamą dalyką, nes vidutiniškai Keiptauno universiteto kursai lankomi 30-60% aktyvumu. [15] Sužaidybinto kurso studentų pažymiai buvo lyginami su praeitais metais studijavusių asmenų pažymiais ir buvo išvelgtas tik nedidelis 3% įverčių pagerėjimas.

Dar vienas aktualus tyrimas buvo atliktas analizuojant studentų mokymosi efektyvumą ir žaidybinių elementų pasitelkimo naudą, mokantis C programavimo kalbos. [12] 22 tyrimo dalyviams buvo pateikta mokymosi platforma, kurioje buvo naudojami žaidybinių elementai. Studentų akademinis tikslas buvo surinkti 100 taškų. Tyrimo dalyviai mokėsi programavimo kalbos atlikdami

¹⁷Babbage Engine <https://www.computerhistory.org/babbage/>

įvairius testus, kurie buvo suskirstyti į 10 pagrindinių temų. Kurso sužaidybinimas pasireiškė skiriant apdovanojimus už pasiekimus. Motyvaciniai aspektai buvo skirstomi į 3 pagrindines grupes: geriausiųjų žaidėjų lentelė, citatų apdovanojimai ir pasiekimų ženkleliai. Geriausiųjų žaidėjų lentelė atvaizdavo studentų sąrašą pagal turimus taškus ir atliktus testus. Citatų apdovanojimo skiltis buvo sudaryta iš žymių, informacinėmis technologijomis užsiimančių, citatų, kurias studentai gaudavo už pasiektą tam tikrą taškų skaičių. Pasiekimų ženklelius reprezentavo žymių žmonių nuotraukos, kurias buvo galima gauti tik surinkus visas atvaizduojamo asmens citatas. Žymių žmonių nuotraukos buvo atvaizduojamos kiekvieno žaidėjo profilyje. Atliekamo tyrimo metu buvo analizuojamos studentų žinios prieš ir po naudojimosi sužaidybintu mokymosi kursu. Tyrimo rezultatai parodė, kad studentai buvo linkę progresuoti žaidime net ir surinkus reikalingą akademinį taškų kiekį, tačiau jų mokymosi strategija dažniausia pasikeisdavo. Studentai pasiekę akademinį taškų ribą ir toliau studijavo C programavimo kalbą, tačiau su žymiai mažesniu aktyvumu ir intensyvumu, visgi tai tik parodo, kad sužaidybinimas leido studentams mokytis ir savo malonumui. Tyrimo metu buvo nustatyta ir patvirtinta hipotezė, kad studentai patobulino ir įsisavino naujas C programavimo kalbos žinias, naudojant sužaidybintą mokymosi kursą. Akademinio požiūriu žaidybinių elementų naudojimas buvo sėkmingas, lyginant studentų žinias, buvo matomas akivaizdus progresas [12].

1.4. Skyriaus apibendrinimas

Besimokančiojo asmens mokymosi procese yra svarbu atkreipti dėmesį į asmens savęs įsivertinimo galimybes. Teisingai realizuotas besimokančiojo asmens įsivertinimo procesas gali didinti asmens įdedamas pastangas į studijuojamą modulį bei pakelti asmens motyvaciją ir norą tobulėti. [14] Siekiant sukurti efektyvų savianalizės ir programavimo mokymosi žaidimą reikia atkreipti dėmesį į žaidimo metu vartotojų turimą motyvaciją, įgūdžius bei laiku pateikti skatinamąjį veiksnį. [7]

Įvertinus 1.2 poskyryje minėtus kompiuterinius programavimo žaidimus galima pastebėti, kad visi žaidimai išlaiko panašią elementų išdėstymo struktūrą ekrane. Žaidybiniai langai yra aiškiai padalinti į scenarijaus kūrimo ir žaidimo vizualizacijos polangius, pavyzdys matomas 2 paveikslėlyje. Taip pat minėtiems įrankiams yra būdinga aiški, pasikartojanti rutina ir žaidimo struktūra, o tai neblaško vartotojo ir leidžia jam susikonsultuoti į sprendžiamą uždavinį. 1.2.2 poskyryje aprašomame „CodinGame“ įrankyje yra matomas Fogo elgsenos modelis - žaidimas siekia kontroliuoti žaidėjo motyvaciją, įvertinti vartotojo turimas žinias ir laiku pateikti skatinamuosius veiksnius. Visi šiame skyriuje minimi įrankiai motyvuoja vartotojus skirdami patirties taškus už sėkmingai įveiktus lygius. Taip pat dažnas motyvacinis aspektas yra pasiekimų ženkleliai ir virtuali valiuta, naudoti Keiptauno ir Lisabonos universitetų tyrimuose. [2][15] Vartotojo žinių trūkumas kompensuojamas pateikiant užuominas (1.2.3 poskyris - „CodeHunt“ ir Keiptauno universiteto tyrimas [15]) arba žaidimo lygiai yra skirstomi pagal jų sudėtingumą (1.2.2 poskyris - „CodinGame“). Skatinamuoju veiksmu gali būti informacinis pranešimas teigiantis, kad vartotojas išsprendęs esamą problemą įgis naują apdovanojimą [7] arba gaus naują žaidimo objektą, kurį galės vėliau panaudoti (1.2.5 poskyris - „Leek Wars“).

Šiame skyriuje aprašyti atlikti tyrimai parodė, kad mokomųjų kompiuterinių žaidimų naudojimas yra vertinamas teigiamai. Vihavaineno įvertinti moksliniai straipsniai parodė, kad dėstymo metodikos, naudojančios kompiuterinius žaidimus, padėjo 10,8% daugiau studentų pasiekti slenkstinį pasiekimų įvertinimą lyginant su standartiniais mokymo metodais.[27] Lisabonos universitete atliktas tyrimas parodė, kad dėstomo kurso medžiagos sužaidybinimas padidino studentų aktyvu-

mą lankant paskaitas 11%, kurso teorinės medžiagos parsisiuntimas padidėjo 47,7%, o aktyvumas elektroninėje mokymosi platformoje Moodle padidėjo 845%. [2] Keiptauno universitete atliktas tyrimas taip pat parodė teigiamą mokomųjų kompiuterinių žaidimų poveikį studentams. Žaidybinius elementus naudojantis kursas buvo lankomas 79,1% aktyvumu, kai tuo tarpu kiti informacinių technologijų kursai buvo lankomi 30-60% aktyvumu. [15] Visgi pažymėtina, kad Lisabonos ir Keiptauno universitetuose atlikti tyrimai neišvengė žymaus studentų pažymių padidėjimo, kaip tai yra matoma pagal Vihavaineno atliktą tyrimą.

2. Sukurtos programos apžvalga

Šio bakalauro baigiamojo darbo metu buvo sukurtas kompiuterinis programavimo žaidimas, kuris vėliau buvo naudojamas atliekant tyrimą, aprašomą 3 skyriuje. Šiame skyriuje bus aprašoma sukurta programa, panaudoti ir alternatyvūs įrankiai bei žaidimo struktūra, reikalavimai ir įgyvendinti vizualiniai aspektai.

2.1. Programos bendrinis aprašymas ir siekiniai

Pagrindinis šio bakalauro baigiamojo darbo tikslas yra sukurti žaidybinius elementus naudojantį įrankį, kuris didintų C# programavimo kalbą studijuojančių asmenų mokymosi motyvaciją ir praplėstų jų įsivertinimo galimybes. Buvo pasirinkta C# programavimo kalba, nes ji patenka į populiariausių programavimo kalbų dešimtuką¹⁸ ir yra plačiai naudojama kuriant internetinius bei mobiliems telefonams skirtus įrankius. [11]

Įvertinus 1.1 poskyryje nagrinėtą kompiuterinio žaidimo naudotojo psichologiją, buvo išskirta Fogo elgsenos modelio [7] ir besimokančiojo asmens įsivertinimo svarba. [14][22] Atsižvelgus į tai, buvo iškeltas siekinys, jog naujai kuriamas žaidimas turi siekti kontroliuoti žaidėjo turimas žinias, motyvaciją bei pateikti skatinamuosius veiksnius. Taip pat kompiuterinio programavimo žaidimo poreikis besimokančiajam asmeniui buvo grindžiamas tuo, kad sėkmingas asmens mokymasis priklauso nuo savęs įsivertinimo galimybių [22], todėl naujas mokymosi įrankis besimokančiajam asmeniui turėtų suteikti papildomas savianalizės galimybes.

Kuriant programavimo žaidimą buvo atsižvelgiama ir į 1.3 poskyryje aprašytus įrankius. Buvo pastebėta, kad nagrinėti programavimo žaidimai išlaiko panašią vizualinių elementų išdėstymo struktūrą ekrane. Žaidybinių langai yra aiškiai padalinti į scenarijaus kūrimo ir žaidimo vizualizacijos polangius. Įvertinus nagrinėtų įrankių grafines vartotojo sąsajas buvo iškeltas siekinys, kad naujai kuriamas žaidimas taip pat turi turėti aiškiai atskirtus scenarijaus kūrimo ir žaidimo vizualizacijos polangius. Taip pat vartotojo sąsaja sukurtame žaidime turi išlikti tokia pati viso žaidimo metu.

Kadangi apžvelgtuose Keiptauno [15] ir Lisabonos [2] universitetų tyrimuose sužaidybimo pritaikymas turėjo turėti stipriai apgalvotą architektūrą, kurioje elementai tarpusavyje yra stipriai integruoti ir sunkiai atskiriami, todėl vienas iš pagrindinių siekinių yra sukurti atskirą įrankį, kurį būtų galima pridėti prie jau esamo universitetinio modulio ir taip padėti besimokantiems asmenim kuo geriau suprasti dėstomą modulį. Su dėstomu moduliu tiesiogiai neintegruotas žaidimas būtų pritaikomas įvairiems universitetinio lygio kursams, taip pat žaidimą būtų galima ruošti pagal modulių dalykinius aprašus, kas padaro įrankį įvairiapusišką ir lengvai plėtojamą.

Programos turinys buvo pagrįstas Vilniaus universitete dėstomų modulių „Programavimas Windows.NET“ ir „Taikomasis objektinis programavimas“¹⁹ dalykiniais aprašais. 3 skyriuje aprašytame tyrime, buvo analizuojama būtent šiuos modulius studijavę asmenys, todėl ir žaidimas buvo grindžiamas moduliuose dėstomais aspektais. Pagrindinis programos tikslas yra suteikti naudotojui galimybę patikrinti ir įsivertinti jau turimas programavimo kalbos žinias. Vienas iš pagrindinių skirtumų tarp sukurtą žaidimo ir 1.3 poskyryje apžvelgtų įrankių yra tai, kad kuriama programa nebuvo tiesiogiai integruota su nei vienu iš minėtų Vilniaus universitete dėstomų modulių, o buvo naudojama kaip papildomas įrankis, skirtas studentams, besimokantiems C# progra-

¹⁸Tiobe (duomenys paskutinį kartą tikrinti 2019 m. sausį, <https://www.tiobe.com/tiobe-index/>)

¹⁹Vytautas Ašeris ir Donatas Kimutis - Taikomasis objektinis programavimas (tikrinta 2018-10-01) http://www.kimutis.lt/3s_TaikomasisObjektinisProgramavimas_LT.pdf

mavimo kalbos.

2.2. Naudojami ir alternatyvūs įrankiai

Įvertinus 2.1 poskyryje išskeltus siekinius, buvo ieškoma įrankio, kurio kuriamas žaidimas būtų lengvai plėtojamas ir realizuotų keliamus vizualinius reikalavimus. Taip pat įrankis turi būti pakankamai populiarus, kad tolimesnius tyrimus atliekantiems arba žaidimą prie tam tikro modulio norintiems pritaikyti asmenims, nebūtų reikalingos itin specifinės programavimo žinios. Populiarus ir žaidimams kurti pritaikytas įrankis, leistų žaidimą padaryti universaliu ir plačiai pritaikomu.

Prieš pasirenkant pagrindinį įrankį, kuriuo buvo kuriama programa, buvo papildomai išskelti šie esminiai techniniai reikalavimai:

1. Programa turi palaikyti C# programavimo kalbos kompiliavimą programos veikimo metu (ang. just-in-time compilation)
2. Žaidybiniai elementai turi būti vizualiai išraiškingi ir motyvuojantys
3. Programa turi veikti nepriklausomai nuo interneto prieigos

2.2.1. Unreal Engine 4

„Unreal Engine 4“ - tai žaidimo variklis sukurtas „Epic Games“²⁰ įmonės. Pirmoji versija pristatyta 1998 m. kartu su žaidimu „Unreal“. Žaidimo variklis turėjo integruotą scenarijų redagavimo kalbą „UnrealScript“ ir žemėlapių valdymo sistemą „UnrealEd“. [17] Dabartinė žaidimų kūrimo platformos versija 4.402 išsiskiria išraiškinga vizualine žaidimų reprezentacija, o kūrimo procesas yra pakankamai nesudėtingas bei vizualiai patrauklus. Šis įrankis buvo svarstomas kaip galimas įrankis, kuriant programavimo žaidimą, dėl to, kad jis laikomas „sėkmingiausiu žaidimų varikliu“ pagal pasaulio Gineso rekordus²¹, kas leidžia manyti, kad įrankis efektyvus žaidimų kūrime. Dabartinis žaidimo variklis skiriasi nuo savo pirmtako ir naudoja C++ programavimo kalbą žaidimų kūrimams. Kadangi programos veikimo metu turi būti palaikomas C# programavimo kalbos kompiliavimas, „Unreal Engine 4“ žaidimų variklis nebuvo tinkamas įrankis. Siekiant įgyvendinti projektą naudojant minėtą įrankį, reikėtų naudoti papildomą kompiliatorių, kuris galėtų programos veikimo metu apdoroti C# programinį kodą.

2.2.2. Visual Basic .NET

„Visual Basic .NET“ (VB.NET) - tai 2002 m. Microsoft bendrovės sukurta objektinė programavimo kalba, kuri naudoja .NET karkasą. Šis įrankis buvo svarstomas, nes jis yra suderinamas su .NET kompiliavimo platforma, geriau žinoma kaip „Roslyn“, kuri leidžia kompiliuoti C# programavimo kalbą programos veikimo metu. Kompiliatorius gali būti pasiekiamas naudojant komandinę eilutę arba aplikacijų programavimo sąsają (ang. API). Renkantis šį „VB.NET“ įrankį būtų reikalingas architektūrinis sprendimas naudojantis aplikacijų programavimo sąsają. Visgi, šis įrankis nebuvo pasirinktas dėl pakankamai skurdžios vartotojo sąsajos iš žaidybinės pusės. Įrankis nėra tiesiogiai pritaikytas kurti kompiuterinius žaidimus, todėl būtų reikalingas papildomas laikas vizualizuojant programą.

²⁰Unreal Engine, <https://www.unrealengine.com/en-US/what-is-unreal-engine-4>

²¹Guinness World Records, <http://www.guinnessworldrecords.com/world-records/most-successful-game-engine>

2.2.3. Unity

Įvertinus 2.2.1 ir 2.2.2 poskyriuose minėtus įrankius buvo ieškoma žaidimo variklio, kuris būtų sutaikomas su .NET kompiliavimo platforma. Šiam projektui puikiai tiko 2005 m. pirmą kartą pristatytas „Unity“ žaidimo variklis. Įrankis yra parašytas C ir C++ programavimo kalbomis, kurias ir naudoja programos veikimo metu, tačiau taip pat yra naudojamas papildomas apvalkalas leidžiantis vartotojui kurti žaidimą naudojant įvairius .NET įrankius. Apvalkalas palaiko didžiąją dalį 2.2.2 poskyryje minėto VB.NET įrankio funkcionalumo, tame tarpe ir „Roslyn“ kompiliatorių. Unity įrankio pagrindinė paskirtis yra žaidimų kūrimas, o tai leidžia lengvai vizualizuoti įvairius žaidybinius aspektus. Kadangi „Unity“ žaidimo variklio atitiko visus keliamus reikalavimus kuriamam programavimo žaidimui, šis įrankis ir buvo pasirinktas programos kūrimui.

2.2.4. Blender

„Blender“²² - tai nemokama ir atviro kodo trimačių kompiuterinių modelių kūrimo ir redagavimo programa. Įrankis taip pat gali būti naudojamas kuriant įvairią animacinę vaizdo medžiagą ar vaizdo efektus. „Blender“ leidžia ne tik sukurti trimatį kompiuterinį modelį, bet ir sugeneruoti modelio tinklėlį, kuris naudojamas atvaizduojant dvimatį vaizdą ant trimačio objekto. Būtent dėl gebėjimo generuoti modelio tinklėlį, šis įrankis buvo naudojamas, kuriant žaidimo vizualinius elementus. Sugeneruoti objektų tinkleliai buvo naudojami 2.2.5 poskyryje aprašomo įrankio.

2.2.5. Adobe Photoshop

„Adobe Photoshop“²³ - tai grafinių elementų kūrimo ir redagavimo įrankis. „Adobe Photoshop“ yra įvairiapusis įrankis leidžiantis redaguoti ir kurti standartines iliustracijas (JPEG, PNG formatais) bei vaizdo medžiagą. „Adobe Photoshop“ buvo naudojamas kuriant trimačių kompiuterinių modelių paviršiaus dizainą. Standartinės dvimatės iliustracijos buvo atvaizduojamos ant trimačių modelių, sukurtų 2.2.4 poskyryje minėtu įrankiu. Taip pat ši programa buvo naudojama kuriant dvimačių žaidimo atributų dizainą.

2.3. Sukurto įrankio apžvalga ir iškeltų siekinių realizacijos vertinimas

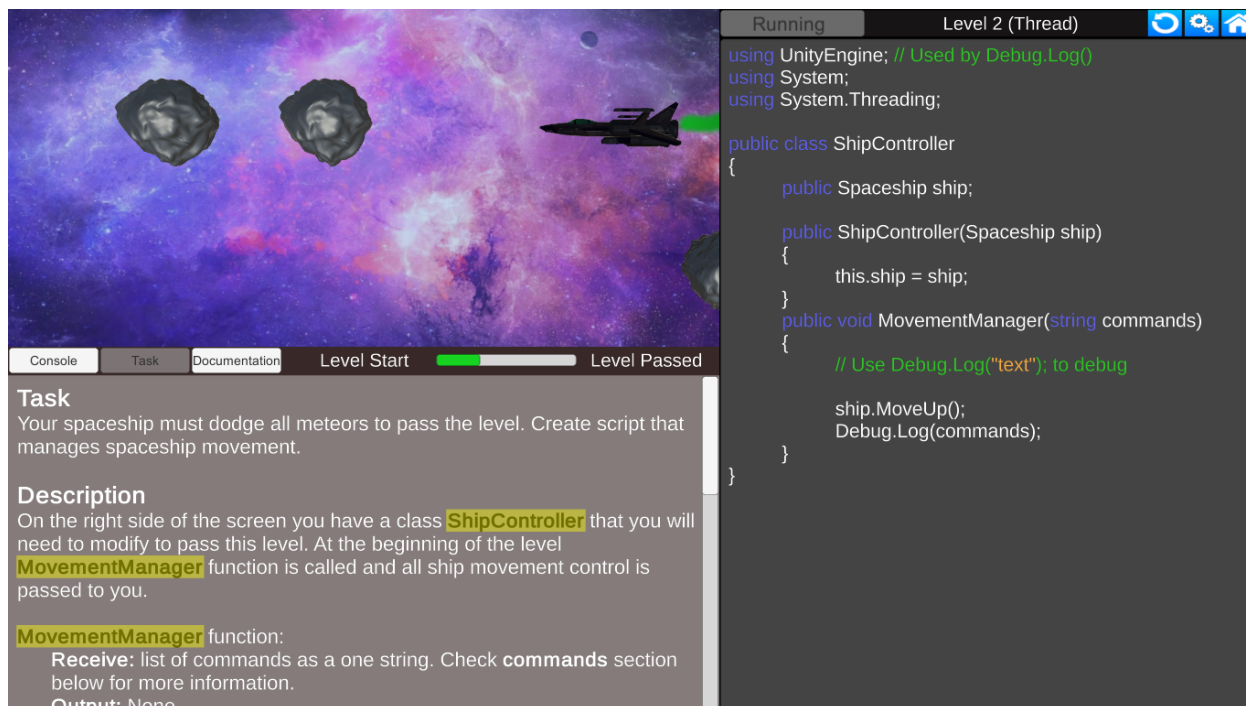
Šio bakalauro baigiamojo darbo metu buvo sukurtas programavimo žaidimas „Spacer“. Žaidimą sudaro 3 žaidimo lygiai apie skirtingus C# programavimo kalbos aspektus. Kiekvienas lygis išlaiko tą pačią vizualinę struktūrą, matomą 3 paveikslėlyje. Dešinėje ekrano pusėje žaidėjas gali kurti reikalingą C# scenarijų, kuris naudojamas žaidimo lygiui įveikti, o kairėje viršutinėje dalyje matoma scenarijaus vizualizacija. Taip pat kairėje apatinėje ekrano dalyje žaidėjas gali matyti tris skiltis: žaidimo pultą, žaidžiamo lygio sąlygą ir dokumentaciją. Siekis išlaikyti pastovią grafinę vartotojo sąsają su aiškiai suskirstytai polangiai buvo pasiektas.

Žaidimas taip pat buvo kuriamas remiantis 1.1 poskyryje aprašytu Fogo elgsenos modeliu. Vartotojo motyvacija buvo siekiama išlaikyti ją apdovanojant patirties taškais ir pasiekimų ženkleliais, kurie buvo matomi žaidimo pradiniam lange. Kadangi šis kompiuterinis programavimo žaidimas nebuvo tiesiogiai integruotas su jokių realių dėstomų kursų, visi motyvaciniai aspektai buvo matomi tik pačiam vartotojui ir pasiekimų ženkleliai ar patirties taškai nebuvo viešai skelbiami. 1.2 poskyryje nagrinėjamuose įrankiuose taip pat dažnas motyvacinis aspektas buvo

²²Blender, <https://www.blender.org>

²³Adobe Photoshop, <https://www.adobe.com/uk/products/photoshop.html>

geriausiųjų žaidėjų lentelė, kuri sukurtame žaidime nebuvo realizuota. 3 skyriuje aprašome tyrime buvo siekiama įvertinti, kaip šio motyvacinio aspekto atsisakymas paveikė žaidėjus. Pagal Fogo elgsenos modelį taip pat svarbu nuspėti žaidėjo jau turimas žinias bei sugebėjimus. Žaidėjo žinių trūkumas buvo kompensuojamas pateikiant žaidimo lygyje reikalingo programavimo kalbos aspekto dokumentaciją. Buvo naudojama „Microsoft“ puslapyje pateikta C# programavimo kalbos dokumentacija²⁴. Kaip skatinamasis veiksnys buvo naudojama progreso juosta (3 pav. žalia juosta, esanti vizualizacinio lango apačioje), kuri reprezentuoja procentinę žaidžiamo lygio įveikimo dalį.



3 pav. Sukurto kompiuterinio programavimo žaidimo „Spacer“ ištrauka

Kuriant programavimo žaidimą buvo daug dėmesio skiriama į įrankio universalumą, plėtros bei pritaikymo galimybes. Žaidimo metu vartotojo kodas buvo kompiliuojamas naudojant atviro kodo kompiliatorių „.NET Compiler Platform“, geriau žinomą kaip „Roslyn“. Įrankis taip pat turi ir kodo analizavimo aplikacijų programavimo sąsają, kuri buvo naudojama informuoti vartotojui apie jo sukurtame scenarijuje esančias klaidas. Buvo pasirinktas „Roslyn“ kompiliatorius dėl įrankio lankstumo ir pritaikymo galimybių. Kompiliatorius buvo pritaikytas taip, kad kuriant naują žaidimo lygį būtų reikalinga sukurti tik naują nuorodą, kuri aprašytų vartotojo kuriamos funkcijos antraštę, t.y. kokio tipo parametrus priima ir gražina vartotojo sukurtas metodus. Asmuo, kuriantis žaidimo lygį, naudojant pasirinktą architektūrinį sprendimą, žino kokio tipo kintamąjį gražins naudotojo kuriama funkcija, todėl lygio kūrėjui belieka vizualizuoti gautą naudotojo rezultatą. Programinio kodo vizualizavimo būdą lygio kūrėjas gali pasirinkti pats ir jis nėra priverstas naudoti tik jau esamus žaidybinius elementus. Siekis, kad sukurtas žaidimas būtų nesudėtingai plėtojamas ir pritaikomas buvo pasiektas.

Taip pat svarbu paminėti, kad žaidėjui pirmą kartą įjungus žaidimą buvo matomas pranešimas, kad 3 skyriuje aprašomo tyrimo tikslais yra renkama anoniminė informacija apie žaidėjo progresą. Žaidėjui pirmą kartą sėkmingai įveikus naują žaidimo lygį buvo sugeneruojamas ir išsiunčiamas elektroninis laiškas su žaidėjo informacija. Laiškai buvo siunčiami standartiniu SMTP protokolu į

²⁴C# dokumentacija <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>

specialiai tyrimui sukurtą elektroninį paštą. Elektroniniame laiške buvo matomas žaidėjo unikalus identifikacinis kodas, įveikto žaidimo lygio numeris, laikas praleistas žaidžiant lygį, laikas praleistas skaitant lygio dokumentaciją ir kiek kartų žaidėjas kompiliavo savo sugeneruotą scenarijų. Visa informacija buvo naudojama tik moksliniais tikslais, 3 skyriuje aprašome tyrime.

3. Atlikto tyrimo apžvalga

Šiame skyriuje bus apžvelgiamas atlikta tyrimas, jo tikslas, uždaviniai, objektas, metodas ir dalyvių imtis. Taip pat bus analizuojami ir apibendrinti gauti rezultatai.

3.1. Tyrimo bendrinė informacija

Tyrimo objektas - kompiuterinių programavimo žaidimų įtaka universitetinio lygio studentams.

Tyrimo tikslas - išsiaiškinti ar kompiuteriniai programavimo žaidimai padeda universitetinio lygio studentams pasitikrinti ir įsivertinti jau turimas programavimo kalbos žinias.

Tyrimo uždaviniai - nustatyti pagrindinius aspektus, turinčius įtaką kompiuterinių programavimo žaidimo patrauklumui. Taip pat įvertinti gautus apklausos rezultatus ir palyginti su jau atliktais moksliniais tyrimais.

Tyrimo metodai - rezultatų, gautų iš kompiuterinio mokomojo žaidimo ir apklausos, analizė.

Tyrimo imtis - 38 Vilniaus Universiteto studentai, studijuojantys arba studijavę „Programavimas Windows.NET“ arba „Taikomas objektinis programavimas“ modulius, Matematikos ir Informatikos fakultete. Kadangi tyrimas buvo integruotas kartu su 2 skyriuje aprašytu programavimo žaidimu, buvo atsižvelgta, kad tiriamasis objektas jau žinotų arba tyrimo metu mokytųsi C# programavimo kalbą, kuri yra naudojama minėtame žaidime.

3.2. Tyrimo pristatymas

2018 - 2019 mokslo metų, rudens semestro metu, buvo atliktas tyrimas, kurio tikslas buvo išsiaiškinti ar kompiuteriniai programavimo žaidimai padeda studentams įsivertinti turimas programavimo kalbos žinias. Taip pat buvo siekiama įvertinti studentų nuomonę apie mokomuosius programavimo žaidimus. Pasiūlymai dalyvauti tyrime buvo išsiųsti 98 Vilniaus Universiteto studentams, kurie jau mokėjo arba mokėsi C# programavimo kalbą. Tyrimo metu studentams buvo pateiktas 2 skyriuje aprašytas kompiuterinis programavimo žaidimas „Spacer“, kuris žaidimo metu palaikė tik C# programavimo kalbos kompiliavimą. Būtent dėl šios priežasties tyrimo imčiai buvo keliamas C# programavimo kalbos žinojimo reikalavimas. Iš 98 studentų 45 (45,91% pradinės imties) studentai bent kartą sužaidė sukurtą programavimo žaidimą, o apklausą užpildė 41 (41,83% pradinės imties) studentą. Siekiant padidinti imties aktyvumą, apklausos publikavimo pradžioje buvo žadami prizai atsitiktiniams apklausos dalyviams, tačiau keleto dienų bėgyje buvo nuspręsta šio motyvacinio sprendimo atsisakyti. Iki motyvacinio aspekto nutraukimo, apklausą jau buvo užpildę 3 asmenys. Siekiant nustatyti ar asmenų atsakymai nebuvo vien tik išoriškai motyvuoti, apklausos pabaigoje, kai buvo turimi visi studentų atsakymai, buvo vertinamas laikas, kiek studentai užtruko pildydami apklausą. Apskaičiavus 38 asmenų (be pirmųjų trijų) apklausos pildymo laiko vidurkį, buvo gautas 235 sekundžių laiko vidurkis, tuo tarpu pirmųjų trijų studentų apklausos pildymo laikas buvo 25 - 35 sekundžių intervale. Taip pat šių studentų atsakymai į visus klausimus buvo pasikartojantys, į visus suminės skalės tipo klausimus buvo atsakoma pasirenkant neutralų atsakymo variantą. Asmenų, užpildžiusių apklausą, bet nežaidusių „Spacer“ programavimo žaidimo nebuvo. Taigi, atlikus apklausos duomenų filtravimą buvo gauta tyrimo imtis, susidaranti iš 38 asmenų, kurie įveikė bent vieną kompiuterinio programavimo žaidimo „Spacer“ lygį ir užpildė apklausą.

Studentai rudens semestro metu, galėjo žaisti „Spacer“ programavimo žaidimą savo pasirinktu metu. Studentams buvo duotas apytiksliai dviejų mėnesių intervalas, dėl dviejų priežasčių. Pirmą priežastį - dalis apklaustųjų semestro metu dar tik mokėsi C# programavimo kalbos Vilniaus universitete „Programavimas Windows.NET“ arba „Taikomasis objektinis programavimas“ dėstomuose moduluose, todėl ne iškart galėjo pilnavertiškai naudotis žaidimu. Antra priežastis - buvo siekiama, kad studentas būtų pats suinteresuotas žaisti programavimo žaidimą jam patogiu metu, taip nedarant įtakos tiriamojo subjekto motyvacijai, nes tai yra vienas iš esminių žmogaus psichologiniu aspektu, žaidžiant kompiuterinius žaidimus. [7]

	Ar programavimo žaidimas „Spacer“ padėjo įsivertinti Jūsų turimas programavimo kalbos žinias?	Ar manote, kad programavimo žaidimas „Spacer“ buvo naudingas jūsų edukacinėje veikloje?	Kaip manote, ar kompiuteriniai programavimo žaidimai būtų efektyvūs universitetinio lygio studijų praktinėse paskaitose?	Kaip manote, ar kompiuteriniai programavimo žaidimai yra efektyvus būdas studentui pasitikrinti turimas programavimo kalbos žinias?
Taip	35 (92%)	35 (92%)	36 (95%)	30 (79%)
Ne	3 (8%)	3 (8%)	2 (5%)	8 (21%)

4 pav. Dalis apklausos klausimų ir jų rezultatų

3.2.1. Kompiuterinio programavimo žaidimo efektyvumas ir žaidėjų motyvacija

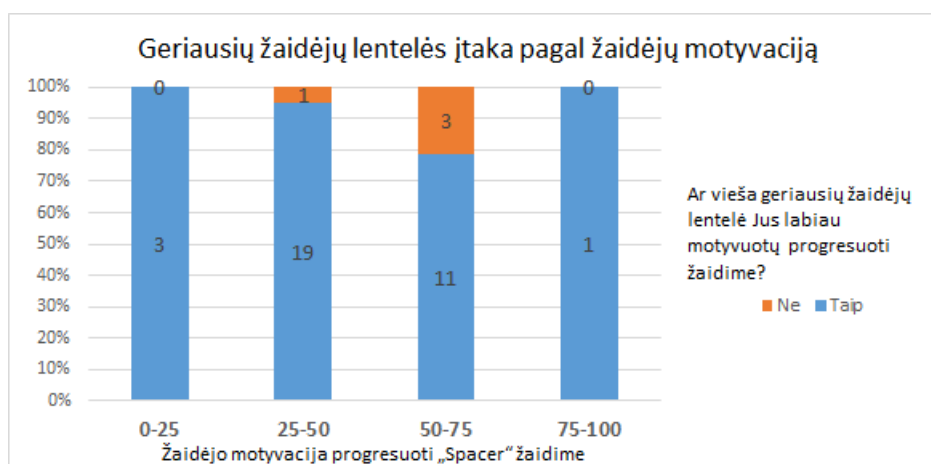
4 paveikslėlyje vaizduojama dalis susistemintų apklausos rezultatų. 92% apklausos dalyvių teigia, kad programavimo žaidimas „Spacer“ jiems padėjo įsivertinti turimas programavimo kalbos žinias. Apklausos pabaigoje buvo galima palikti papildomą komentarą. Vienas apklaustųjų apie programavimo žaidimo naudą, siekiant įsivertinti turimas žinias, komentaruose teigė, kad „tokie žaidimai panašiai kaip darbdavio paskirtos užduotys - padaryk tą naudodamas tą ir tą. Tuomet supranti ko nežinai ir ką reiktų išmokyti, todėl tai yra geras būdas pasitikrinti“. Panašią poziciją išsakė ir kitas apklaustasis: „uždaviniai, kada reikia panaudoti konkretų įrankį, gerai padeda įsivertinti turimas žinias“.

Siekiant išsiaiškinti kompiuterinių žaidimų plėtros galimybes Vilniaus universitete, apklausos dalyvių buvo klausiama ar jie mano, kad kompiuteriniai programavimo žaidimai būtų efektyvūs universitetinio lygio paskaitose. Net 95% apklaustųjų atsakė teigiamai į minėtą klausimą. Taip pat svarbu paminėti, kad 2 asmenys komentaruose papildomai išreiškė norą ir susidomėjimą, išbandyti sužaidybintą universitetinio lygio kursą. Ateities tyrimuose rekomenduojama pritaikyti kompiuterinius žaidimus viename iš jau dėstomų Vilniaus universiteto modulių ir iširti žaidimų poveikį studentams.

Taip pat 92% tyrimo imties nurodė, kad jų nuomone, programavimo žaidimas buvo naudingas jų edukacinėje veikloje. Visgi pažymėtina, kad 21% apklaustųjų nurodė, kad jų nuomone, kompiuteriniai programavimo žaidimai nėra efektyvus būdas studentui pasitikrinti turimas programavimo kalbos žinias. Siekiant įvertinti, kodėl žaidėjai nemano, kad žaidimai yra efektyvus savitikros būdas, reikėtų analizuoti vartotojų motyvaciją žaidimo metu. Prielaidą, kad būtent tokiai apklaustųjų pozicijai turi įtakos žaidėjų motyvacija, galima daryti atsižvelgus į apklaustųjų komentarus. Iš 8 (21% visos imties) asmenų, kurie mano, kad žaidimai nėra efektyvus žinių pasitikrinimo būdas, 4 paliko papildomą komentarą. Visi komentarai nurodė neigiamus atsiliepimus apie žaidimo vizualizaciją arba motyvacijos trūkumą. „Perėjus žaidimą nežinau kiek laiko išliks informacija, nes žaidimas neįtraukė“, „žaidimai turi būti gerai apgalvoti ir parinkti, kad būtų įdomūs ir efektyvūs“ - tai du iš minėtų keturių neigiamų komentarų apie sukurtą įrankį. Aiškinantis žaidėjų motyvacijos klausimą reikėtų įvertinti jau esamus motyvacinius aspektus sukurtame žaidime. 2 skyriuje yra

nurodyta, jog žaidėjai yra motyvuojami apdovanojant juos pasiekimų ženkleliais ir patirties taškais. Tokius pačius motyvacinius aspektus naudoja ir 1 skyriuje aprašomi įrankiai, tačiau esminis skirtumas, kurį galima išskirti tarp sukurto žaidimo ir jau esamų įrankių, yra tai, kad apžvelgtų įrankių naudotojai gali matyti kitų žaidėjų pasiekimus, o „Spacer“ žaidime, tokia informacija nėra pasiekama. Šią problemą atspindi 5 paveikslėlyje pateikiama diagrama. Apklaustos metu vartotojai įvertino savo motyvacijos lygį žaidimo metu. 60,52% apklaustųjų savo motyvaciją įvertino žemiau nei vidutiniu lygiu (≤ 50). Sugrupavus vartotojus pagal jų motyvacijos lygį, 5 pav. galime matyti jų pasiskirstymą atsakant į klausimą „ar vieša geriausių žaidėjų lentelė jus labiau motyvuotų progresuoti žaidime“. Diagramoje matomas ne tik bendras naudotojų suinteresuotumas dėl geriausių žaidėjų lentelės, tačiau pažymėtina ir tai, kad didžioji dalis (95%) asmenų, kurių motyvacija progresuoti žaidime buvo žemesnė nei vidutinė, nurodė, kad jie norėtų pamatyti geriausių žaidėjų lentelę šiame žaidime. Vertinant visą tyrimą, 89,47% apklaustųjų nurodė, kad geriausių žaidėjų lentelė padidintų jų motyvaciją progresuoti žaidime. Patirties taškai, apdovanojimų ženkleliai ir geriausių žaidėjų lentelė yra populiariausi motyvaciniai aspektai mokomuosiuose žaidimuose [10], todėl panašios apklaustųjų reakcijos ir buvo galima tikėtis. Kadangi žaidimas „Spacer“ buvo kuriamas, susitelkiant į vieną vartotoją ir jo galimybes patikrinti turimas žinias, geriausių žaidėjų lentelė nebuvo realizuota, todėl apklausoje buvo matomas žaidėjų išreikštas noras pamatyti būtent šį motyvacinį elementą programavimo žaidime.

Taip pat ateities tyrimuose reiktų atlikti papildomą analizę, siekiant įvertinti ar žaidėjų motyvacinis efektas nėra trumpalaikis dėl įrankio naujumo. Siekiant įvertinti šį teiginį, yra reikalingas ilgalaikis studentų stebėjimas ir vertinimas, kuomet besimokantysis asmuo naudoja sužaidybintus įrankius nebe pirmą kartą.

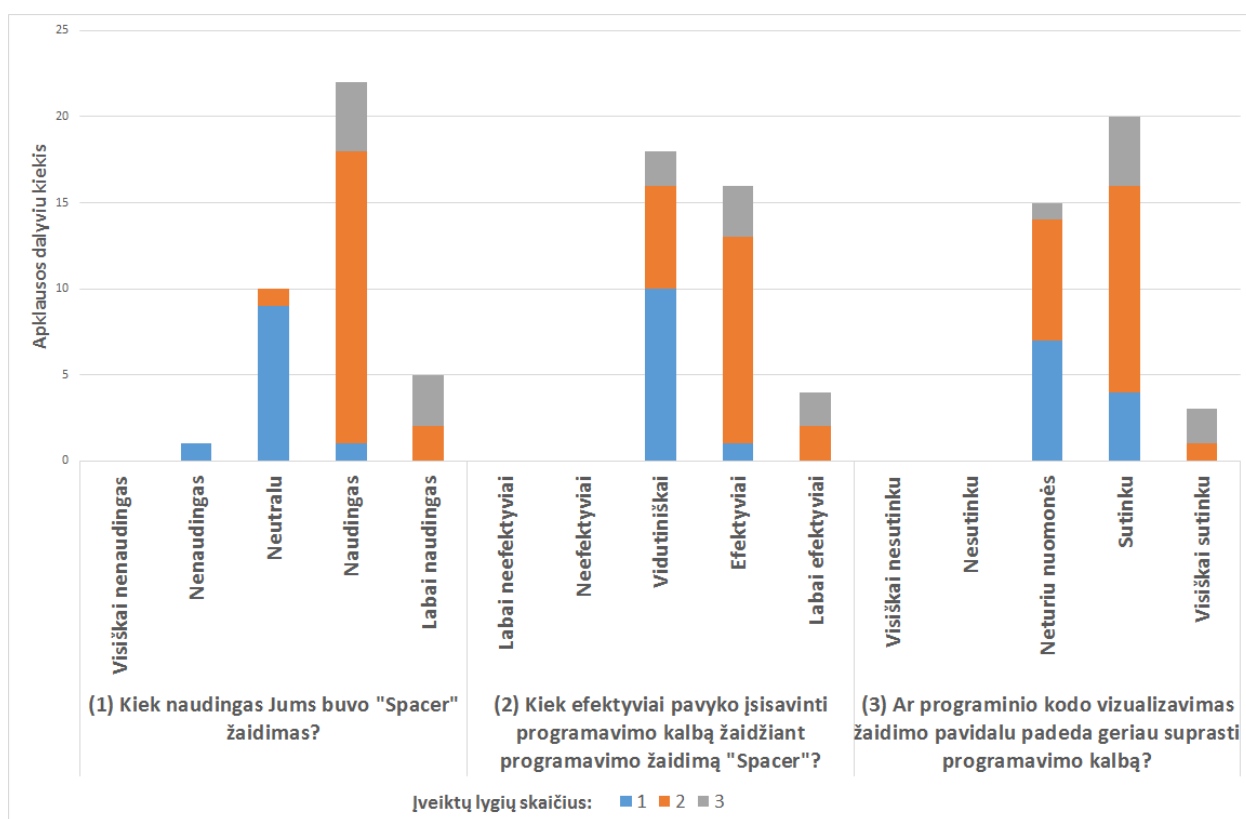


5 pav. Geriausių žaidėjų lentelės svarba pagal žaidėjų motyvacijos pasiskirstymą

3.2.2. Kompiuterinio programavimo žaidimo naudingumas ir vizualizacijos svarba

6 paveikslėlyje vaizduojamoje diagramoje galima matyti apklausos dalyvių pasiskirstymą pagal įveiktų lygių skaičių. Pagal Likerto skalę nuo 1 iki 5, kur 1 reprezentuoja neigiamą teiginį („visiškai nesutinku“, „labai neefektyvu“, „visiškai nenaudinga“), o 5 teigiamą („visiškai sutinku“, „labai efektyvu“, „labai naudingas“), studentai dažniausiai rinkosi neutralią (3) arba dalinai teigiamą (4) atsakymo variantą į klausimus, susijusius su žaidimo naudingumu ir vizualizacijos svarba. Siekiant išlaikyti šio poskyrio paprastumą, bet tuo pačiu ir informacijos pilnumą, 6 paveikslėlyje atvaizduojami klausimai bus referuojami jų eilės numeriais, nurodytais diagramoje.

Pirmojo klausimo, „kiek naudingas jums buvo „Spacer“ žaidimas“, moda ir mediana yra 4 („naudingas“) su akivaizdžiu dažnių skirtumu („nenaudingas“ - 1, „neutralu“ - 10, „naudingas“ - 22, „labai naudingas“ - 5). Visgi atsakymus į šį klausimą reiktų vertinti pakankamai kritiškai atsižvelgus į tai, kad apklausos korespondentai galimai įvertino save labiau teigiamai nei iš tiesų jiems atrodė, nes norėjo atrodyti socialiai patrauklesni. [18] Paulhus savo tyrime parodė, kad asmenys, kurie žino arba mano, kad juos bus galima identifikuoti analizuojant apklausos duomenis, yra linkę apie save pateikti labiau teigiamą ir socialiai priimtina informaciją nei visiškai anonimiškose apklausose. [18] Nors ir studentams buvo nurodyta, kad apklausa ir žaidimas yra anoniminiai, tam, kad susieti apklausą su žaidimo duomenimis, buvo prašoma apklausoje nurodyti žaidime sugeneruotą specialų 5 simbolių kodą. Būtent identifikacinio kodo nurodymas, galėjo sukelti apklausos dalyviams anonimiškumo trūkumo įvaizdį, nes asmenys negalėjo būti visiškai užtikrinti, kokie duomenys buvo renkami žaidimo metu. Siekiant teisingai įvertinti klausimo rezultatų tikrumą, reikalinga išanalizuoti ir kitus 6 pav. vaizduojamus klausimus.



6 pav. Suminės skalės klausimai pagal įveiktų žaidimo lygių skaičių

Programavimo kalbos įsisavinimo efektyvumą, žaidžiant kompiuterinį programavimo žaidimą „Spacer“ (2 klausimas), apklausos dalyviai vertino labiau kritiškai nei pačio žaidimo naudingumą (1 klausimas). Šio klausimo mediana taip pat buvo 4 („efektyvu“), tačiau moda yra 3 („vidutiniškai“). Komentaruose buvo pažymėta, kad standartiniai programavimo uždaviniai taip pat efektyviai patikrina turimas žinias kaip ir programavimo žaidimai, tačiau vienintelis skirtumas yra tai, kad žaidimai tai padaro vizualiai patraukliau nei standartiniai uždaviniai, o tai dar nereiškia, kad programavimo kalba bus įsisavinta efektyviau. Bendrinį teiginį, kad programinio kodo vizualizavimas žaidimo pavidalu padeda geriau suprasti programavimo kalbą (3 klausimas), apklausos dalyviai vertino pakankamai teigiamai. Įvertinus atsakymus į trečią klausimą, buvo gauta mediana ir moda lygi 4 („sutinku“). Pažymėtina yra tai, kad bendrinio aspektu apie kompiuterinius programavimo žaidimus apklausos dalyviai atsiliepia teigiamai ir jie mano, kad žaidimai padeda geriau

suprasti programavimo kalbą, tačiau konkretniu klausimu, apie šiame tyrime naudotą žaidimą „Spacer“, apklaustieji atsiliepija negatyviau nei bendrinio aspektu. Nuomonę apie „Spacer“ žaidimą ir jo trūkumus atspindi ir 3.2.1 poskyris. Visgi, vertinant visų trijų klausimų rezultatus, yra pažymėtina, kad apie kompiuterinius programavimo žaidimus apklausos dalyviai atsiliepija teigiamai. Nors pateiktas įrankis ir turėjo trūkumų, studentai nurodė, kad jiems žaidimas buvo naudingas ir padėjo įsisavinti programavimo kalbą.

6 paveikslėlio diagramoje yra matomas apklaustųjų pasiskirstymas pagal įveiktų žaidimo lygių skaičių. Pagal pateiktą diagramą galima išvelgti tiesioginę koreliaciją tarp įveiktų žaidimo lygių ir atsakymų pozityvumo, suminės skalės klausimuose. Žaidėjai įveikę vieną žaidimo lygį dažniausiai rinkosi neutralią poziciją atsakant į klausimus (82%, 91%, 64% - vieną žaidimo lygį įveikusių ir neutralų (3) atsakymo variantą pasirinkusių asmenų procentinė dalis atitinkamai 1, 2 ir 3 klausimuose). Taip pat du žaidimo lygius įveikę asmenys dažniausia rinkosi dalinai pozityvų (4) atsakymo variantą (85%, 60%, 60% - atitinkamai 1, 2 ir 3 klausimuose). Įvertinus žaidėjų pasiskirstymą pagal įveiktus lygius, galima pažymėti, kad kuriant kompiuterinį programavimo žaidimą reikalinga atkreipti papildomą dėmesį į mažai progresuojančius ir ne itin motyvuotus žaidėjus, nes jie išlaiko mažiau pozityvią poziciją apie programavimo žaidimus nei efektyviai progresuojantys asmenys. Taip pat remiantis apklausos duomenimis matoma, jog didesnis progresas žaidime, padeda efektyviau įsisavinti programavimo kalbą ir jausti įvairiapusišką žaidimo naudą.

3.3. Tyrimo rezultatų apibendrinimas ir palyginimas su jau atliktais tyrimais

Atliktas tyrimas parodė, kad tyrimo imtis didžiąja dalimi teigiamai atsiliepė apie kompiuterinio programavimo žaidimo įtaką jų mokymuisi ir įsivertinimo galimybėms. 92% apklausos dalyvių teigia, kad pateiktas programavimo žaidimas padėjo jiems įsivertinti turimas programavimo kalbos žinias ir buvo naudingas jų edukacinėje veikloje. Visgi yra pažymėtina, kad 21% apklaustųjų mano, kad mokytojai žaidimai nėra efektyvūs mokymosi būdas. Atlikus detalesnę šios negatyvios nuomonės analizę, paaiškėjo, kad tokiai apklaustųjų pozicijai įtakos turėjo žaidėjų motyvacijos lygis, žaidžiant „Spacer“ programavimo žaidimą. 60,52% (23 studentai) tyrime dalyvavusių asmenų nurodė, kad jie buvo mažai motyvuoti progresuoti žaidime (skalėje nuo 1 iki 100, savo motyvacijos lygį nurodė mažesnę nei 50). 95% asmenų, kurių motyvacija progresuoti žaidime buvo mažesnė nei vidutinė (≤ 50 pagal motyvacijos skalę), nurodė, kad juos motyvuotų geriausių žaidėjų lentelė. Remiantis šiais duomenimis galima daryti išvadą, kad besimokantieji asmenys yra suinteresuoti konkuruoti su kitais studentais ir taip siekti geresnių rezultatų. Taip pat buvo tiriama ir programavimo kalbos įsisavinimo efektyvumas naudojant kompiuterinį programavimo žaidimą „Spacer“. Pagal suminės skalės klausimus su penkiais atsakymo variantais, apklausos dalyviai, programavimo kalbos įsisavinimo efektyvumą dažniausia vertino „vidutiniškai“ (3 variantas) arba „efektyvu“ (4 variantas). Pažymėtina yra tai, kad bendrinio aspektu apie kompiuterinius programavimo žaidimus, apklausos dalyviai atsiliepė labiau teigiamai, nei apie pateiktą programavimo žaidimą „Spacer“. Bendrinį teiginį, kad programinio kodo vizualizavimas žaidimo pavidalu padeda geriau suprasti programavimo kalbą, vertino teigiamai (klausimo atsakymų moda ir mediana lygi 4 - „sutinku“). Taip pat svarbu paminėti, kad buvo išskirtas žaidėjų pasiskirstymas pagal įveiktus lygius, kas parodė, jog, kuo žaidėjas mažiau progresuoja žaidime, tuo negatyviau jis atsiliepija apie visą mokymo techniką, naudojant kompiuterinius programavimo žaidimus.

Lisabonos [2] ir Keiptauno [15] universitetuose atlikti tyrimai, analizavo kompiuterinių žaidimų įtaką, integruojant juos į dėstomus kursus universitete, tuo tarpu šiame skyriuje aprašytas

tyrimas analizavo besimokančiojo įsivertinimo galimybes, naudojant individualų programavimo žaidimą, nesusietą su kitais vartotojais. Visgi lyginant šios apklausos rezultatus su kitais atliktais, 1.3 poskyryje aprašytais, tyrimais, galima pastebėti, kad visi, Vilniaus, Lisabonos ir Keiptauno universitetų studentus analizuojantys tyrimai, parodė teigiamus rezultatus apie kompiuterinių mokomųjų programavimo žaidimų įtaką besimokančiam asmeniui. Taip pat pažymėtina, kad visuose tyrimuose yra išskiriama vartotojo motyvacija, kaip kertinis aspektas, siekiant teigiamos kompiuterinio programavimo žaidimo įtakos studentui.

Išvados ir rekomendacijos

Šiame bakalauro baigiamajame darbe buvo sukurtas ir ištestuotas universitete įgyjamų žinių įsivertinimo įrankis, kuriame buvo pritaikyti sužaidybinimo aspektai. Atsižvelgiant į jau esamų programavimo žaidimų naudojamus motyvacinius aspektus, sukurtame įrankyje buvo įgyvendinti žaidėjo patirties taškai bei pasiekimų ženkleliai. Atliktas tyrimas parodė, kad, naudojant šiuos motyvacinius aspektus, 60,52% sukurtų įrankio naudotojų savo motyvacijos lygį vertino žemiau nei vidutiniškai (≤ 50 , skalėje nuo 0 iki 100). Įrankyje nebuvo naudotas sužaidybinimo aspektas, kaip geriausiųjų žaidėjų sąrašas, norint patikrinti, kiek jis yra svarbus besimokantiesiems asmenims. Susisteminius tyrimo rezultatus buvo nustatyta, kad 89,47% apklaustųjų teigia, jog geriausiųjų žaidėjų lentelė padidintų jų motyvaciją progresuoti žaidime.

Apibendrinus jau esamus programavimo žaidimus ir atliktus tyrimus, bei įvertinus gautus tyrimo rezultatus, galima daryti išvadą, kad kompiuteriniai programavimo žaidimai padeda studentams įsivertinti jau turimas programavimo kalbos žinias. Atliktas tyrimas parodė, kad 92% įrankį bandžusių asmenų pateiktą programavimo žaidimą vertino kaip naudingą įrankį edukacinėje veikloje. Išanalizuoti moksliniai straipsniai ir atliktas tyrimas parodo, kad programavimo žaidimai yra naudingi studentų edukacinėje veikloje, nes studentai gali įsivertinti savo turimas žinias, o efektyvus savęs įsivertinimo procesas gali didinti besimokančiojo asmens vidinę motyvaciją, įdedamas pastangas ir norą tobulėti užsibrėžto tikslo link. Taip pat, atsižvelgiant į atlikto tyrimo rezultatus, galima daryti išvadą, kad besimokančiojo motyvacija yra kertinis aspektas, siekiant sėkmingai įgalinti programavimo žaidimus kaip galimą mokymosi techniką. Lisabonos ir Keiptauno universitetuose atlikti tyrimai parodė, kad kompiuteriniai žaidimai padidina studentų aktyvumą lankant dėstomus modulius ir siunčiantis teorinę kurso medžiagą. Vihavaineno išanalizuoti moksliniai straipsniai parodė, kad dėstymo metodikos, naudojančios kompiuterinius žaidimus, padėjo 10,8% daugiau studentų pasiekti slenkstinį pasiekimų įvertį lyginant su standartiniais mokymo metodais, o įvertinus Vilniaus universitete atliktą tyrimą, yra matoma, kad studentai yra linkę naudoti kompiuterinius programavimo žaidimus ir kaip papildomą įrankį, leidžiantį pasitikrinti ir įsivertinti turimas žinias.

Siekiant įgalinti kompiuterinius programavimo žaidimus dėstomuose moduluose ar kaip papildomą savitikros įrankį, yra rekomenduojama atkreipti dėmesį į mažai motyvuotus ir minimaliai žaidime progresuojančius asmenis. Rekomenduotina, kaip motyvacinius aspektus naudoti žaidėjų patirties taškus, pasiekimų ženklelius ir geriausiųjų žaidėjų lentelę. Naudotojo turimų žinių kontrolė ir kompensavimas taip pat yra siektinas veiksnys. Atliekant tyrimą buvo pastebėtas aktyvus tiriamosios imties noras išbandyti sužaidybintą universitetinio lygio modulį, todėl rekomenduojama pritaikyti kompiuterinius programavimo žaidimus arba žaidybinius elementus bent viename iš jau esamų universitetinių modulių.

Ateities tyrimų planas

Ateities tyrimams yra numatoma tiesiogiai integruoti kompiuterinį programavimo žaidimą į pasirinktą universitetinio lygio modulį. Tyrimo siekiamybė būtų detalizuoti, kokius veiksmus reikia atlikti kurso dėstytojui, siekiant įgyvendinti sužaidybintus įrankius. Taip pat numatoma išanalizuoti įsivertinimo įrankio kūrimo ir tobulinimo kaštus. Įvertinus laiko ir piniginius aspektus, universitetinių modulių dėstytojams būtų suteikta galimybė įvertinti papildomo savianalizės įrankio reikalingumą ir teikiamą naudą prie dėstomo modulio.

Šiame rašto darbe aprašyto tyrimo metu nebuvo galimybės įvertinti sukurto įrankio įtakos besimokančiojo pažymiams. Vihavaineno atliktas tyrimas parodė, jog programavimo žaidimai padėjo 10,8% daugiau studentų pasiekti slenkstinių pasiekimų įvertį lyginant su standartiniais mokymo metodais, tačiau Keiptauno ir Lisabonos universitetuose atlikti tyrimai žymaus pažymių pokyčio neižvelgė. Ateities tyrimams yra planuojama išanalizuoti universitete įgyjamų žinių įsivertinimo įrankio įtaką studentų pažymiams, kadangi pagal dabartinius surinktus duomenis ir jau atliktą tyrimą, negalima priimti užtikrintos išvados apie besimokančiųjų asmenų pažymių pokytį naudojant sužaidybintą įsivertinimo įrankį.

Literatūros šaltiniai

- [1] Julian Alvarez. *Serious games: Advergaming, edugaming, training*. Understanding the digital world. IDATE, Montpellier, 2008.
- [2] Gabriel Barata, Sandra Gama, Joaquim Jorge, and Daniel Goncalves. Engaging engineering students with gamification. In *2013 5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, pages 1--8. IEEE, 2013.09.11 - 2013.09.13.
- [3] Sue Bennett, Karl Maton, and Lisa Kervin. The 'digital natives' debate: A critical review of the evidence. *British Journal of Educational Technology*, 39(5):775--786, 2008.
- [4] Sébastien Combéfis, Gytautas Beresnevičius, and Valentina Dagienė. Learning programming through games and contests: Overview, characterisation and discussion. *Olympiads in Informatics*, 10(1):39--60, 2016.
- [5] Miguel de Aguilera and Alfonso Mendiz. Video games and education. *Computers in Entertainment*, 1(1):10, 2003.
- [6] Michele D. Dickey. *Engaging By Design: How Engagement Strategies in Popular Computer and Video Games Can Inform Instructional Design*. Educational Technology Research and Development, 2005.
- [7] B. J. Fogg. A behavior model for persuasive design. In Samir Chatterjee and Parvati Dev, editors, *Proceedings of the 4th International Conference on Persuasive Technology - Persuasive '09*, page 1, New York, New York, USA, 2009. ACM Press.
- [8] J. P. Gee. *Good video games and good learning*. Phi Kappa Phi Forum.
- [9] Maurits Graafland. *Serious games in surgical education*. [s.n.], [S.l.], 2014.
- [10] Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work? -- a literature review of empirical studies on gamification. In *2014 47th Hawaii International Conference on System Sciences*, pages 3025--3034. IEEE, 2014.01.06 - 2014.01.09.
- [11] Anders Hejlsberg, Scott Wiltamuth, and Peter Golde. *The C# programming language*. Microsoft .NET development series. Addison-Wesley, Harlow, 2nd ed. edition, 2006.
- [12] Maria-Blanca Ibanez, Angela Di-Serio, and Carlos Delgado-Kloos. Gamification for engaging computer science students in learning activities: A case study. *IEEE Transactions on Learning Technologies*, 7(3):291--301, 2014.
- [13] Fedwa Laamarti, Mohamad Eid, and Abdulmotaleb El Saddik. An overview of serious games. *International Journal of Computer Games Technology*, 2014(3):1--15, 2014.
- [14] James H. McMillan and Jessica Hearn. Student self-assessment: The key to stronger student motivation and higher achievement. *Educational Horizons*, 87(1):40--49, 2008.
- [15] Siobhan O'Donovan, James Gain, and Patrick Marais. A case study in the gamification of a university-level games development course. In John McNeill, Karen Bradshaw, Philip Machanick, and Mosiuoa Tsietsi, editors, *Proceedings of the South African Institute for*

Computer Scientists and Information Technologists Conference on - SAICSIT '13, page 242, New York, New York, USA, 2013. ACM Press.

- [16] Harold F. O'Neil, Richard Wainess, and Eva L. Baker. Classification of learning outcomes: evidence from the computer games literature. *Curriculum Journal*, 16(4):455--474, 2005.
- [17] Partha Sarathi Paul, Surajit Goon, and Abhishek Bhattacharya. History and comparative study of modern game engines. *International Journal of Advanced Computer and Mathematical Sciences*, 3:245--249, 2012.
- [18] Delroy L. Paulhus. Two-component models of socially desirable responding. *Journal of Personality and Social Psychology*, 46(3):598--609, 1984.
- [19] Jon Peddie. *The history of visual magic in computers: How beautiful images are made in CAD, 3D, VR and AR / Jon Peddie*. Springer, London, 2013.
- [20] Marc Prensky. Digital game-based learning. *Computers in Entertainment*, 1(1):21, 2003.
- [21] P. R. Prinrich and D. H. Schunk. Motivation in education: theory, research, and application. *Ohio: Prentice-Hall*, 1996.
- [22] D. H. Schunk. *Learning theories: An educational perspective*. Pearson, Boston, 6th ed. edition, 2012.
- [23] Lee Sheldon. *The multiplayer classroom: Designing coursework as a game*. Course Technology/Cengage Learning, Australia and Boston Mass., 2012.
- [24] Lorrie A. Shepard. The role of classroom assessment in teaching and learning. In *Handbook of Research on Teaching*, 4.
- [25] Valerie J. Shute. *Stealth assessment in computer-based games to support learning*. Computer Game and Instruction, 2011.
- [26] Kurt Squire. Video games in education. *International Journal of Intelligent Games & Simulation*, 2(1):49--62, 2003.
- [27] Arto Vihavainen, Jonne Airaksinen, and Christopher Watson. A systematic review of approaches for teaching introductory programming and their influence on success. In Quintin Cutts, Beth Simon, and Brian Dorn, editors, *Proceedings of the tenth annual conference on International computing education research - ICER '14*, pages 19--26, New York, New York, USA, 2014. ACM Press.

Priedai

Dokumentą sudaro vienas priedas (A), kuriame yra vaizduojami visi atliktos apklausos atsakymai. Geltonai pažymėti atsakymai nebuvo vertinami, nes šių apklausų pildymo laikas buvo 25 - 35 sekundžių intervale, tuo tarpu bendras apklausos pildymo vidurkis buvo 235 sekundės. Taip pat šių studentų atsakymai į visus klausimus buvo pasikartojantys, į visus suminės skalės tipo klausimus buvo atsakoma pasirenkant neutralų atsakymo variantą (daugiau informacijos 3.2 poskyryje).

A. Apklausos atsakymai

Eil. Nr.	Apklausos atlikimo laikas	Identifikacinis numeris iš žaidimo Spacer	Įveiktų žaidimų lygiu skaičius	Laikas praleistas lygiuose			Laikas praleistas dokumentacijose		
				Pirmas lygis	Antras lygis	Trečias lygis	Pirmas lygis	Antras lygis	Trečias lygis
1	294	45jhh	2	545	645		34	98	
2	197	37g31	3	500	945	1453	158	315	456
3	213	gjyk8	2	342	669		15	65	
4	191	l2c6c	2	215	749		0	12	
5	241	xusjj	1	599			175		
6	269	2otpa	1	311			30		
7	222	8dg0h	2	401	512		0	242	
8	265	16iay	3	645	969	931	123	298	215
9	291	pb118	2	537	602		136	353	
10	263	nsqca	2	427	740		162	350	
11	193	sw2u6	2	591	957		45	215	
12	270	65shl	1	365			12		
13	186	18q0t	2	563	695		166	276	
14	300	n3mv6	1	592			130		
15	263	1lnoa	3	600	909	626	66	267	35
16	177	ji6a4	2	482	824		89	211	
17	266	xgmkp	1		814			140	

18	204	vmlijm	2	612	912		145	320	
19	271	0a6cq	2	562	800		164	189	
20	247	93pkv	1	630			89		
21	212	ne4w4	2	485	855		148	250	
22	201	fgxzn	2	513	916		137	196	
23	248	lgk67	1	508			198		
24	282	9zufh	3	337	600	872	64	111	240
25	248	gvs0o	3	597	991	1100	157	313	514
26	283	kl760	2	343	866		191	357	
27	239	4wtqa	2	339	942		143	193	
28	203	tf47e	1	180			0		
29	225	8fvvf	2	321	785		12	262	
30	202	ns16i	2	531	818		128	215	
31	232	r939m	1	640			132		
32	248	86dwk	3	590	442	974	154	199	245
33	186	c7nbn	2	592	488		148	353	
34	221	1ppos	2	312	590		44	200	
35	194	c4w7a	3	652	745	972	43	155	287
36	229	ydyfk	2	531	816		178	269	
37	270	mog69	1	603			135		
38	189	hkgf3	1	432			236		
39	25	jk44i	1	530			49		
40	35	iojf3	1	420			35		
41	31	esu11	1	411			85		

Eil. Nr.	Kompiliavimų kiekis			Pažymėkite kokias programavimo žaidimų platformas esate bandę						
	Pirmas lygis	Antras lygis	Trečias lygis	Codeacademy	Robocode	Code Combat	CodinGame	CodeHunt	Code Fights	Neaktualu (Niekada nebandžiau)
1	2	3		1			1			
2	4	3	6							1
3	2	4								1
4	1	5								1
5	4									1
6	2									1
7	2	3					1			
8	4	8	5	1	1		1			
9	3	3								1
10	2									1
11	3	5								1
12	1									1
13	3	4					1			
14	2									1
15	5	6	5							1
16	3	6								1
17		4								1

18	3	2		1						
19	3	5								1
20	2									1
21	2	4								1
22	4	7								1
23	2									1
24	2	5	4							1
25	2	4	9							1
26	3	4								1
27	2	6				1	1			
28	1									1
29	1	3		1						
30	2	4								1
31	3									1
32	3	5	8							1
33	1	2								1
34	1	3								1
35	2	5	7							1
36	2	3								1
37	3									1
38	2						1			
39	4									1
40	3									1
41	3									1

Eil. Nr.	Pasirinkite Jums patogiausią programavimo kalbos mokymosi techniką.				Pakomentuokite savo pasirinkimą (neprivaloma)
	Problemy sprendimu pagrįstas mokymasis naudojant programavimo žaidimus	Problemy sprendimu pagrįstas mokymasis naudojant standartines programas	Projektiniais darbais pagrįstas mokymasis	Teorinės medžiagos analizė pagrindžiant praktiniais pavyzdžiais	
1		1			
2			1		
3		1			Jau įprates prie tokio būdo
4		1			
5		1			
6				1	
7			1		
8	1				Jau senai žaidžiu įvairius programavimo žaidimus
9			1		Dažniausia taip moko
10		1			
11		1			
12		1			
13			1		
14		1			Žaidimai labai neįprasta
15				1	Keista mokytis programuoti per žaidimą
16			1		
17				1	

18			1		
19		1			
20			1		
21				1	
22		1			
23			1		
24		1			Taip patogiausia
25		1			
26			1		
27	1			1	
28			1		Jei prieš atsiskaitymą, tai tik teorinę medžiagą peržiūriu
29				1	
30		1			
31			1		
32			1		
33		1			Projektais man labiausia įprasta. Žaidimais niekad nebuvo bandęs
34			1		
35		1			
36				1	
37				1	
38		1			
39		1			
40			1		
41	1				

Eil. Nr.	Kiek naudingas Jums buvo "Spacer" žaidimas?					Kiek efektyviai pavyko įsisavinti programavimo kalbą žaidžiant programavimo žaidimą "Spacer"?				
	Visiškai nenaudingas	Nenaudingas	Neutralu	Naudingas	Labai naudingas	Labai neefektyviai	Neefektyviai	Vidutiniškai	Efektyviai	Labai efektyviai
1				1				1		
2				1				1		
3				1				1		
4				1					1	
5			1					1		
6			1					1		
7				1					1	
8					1					1
9				1					1	
10				1				1		
11					1					1
12			1					1		
13				1					1	
14			1					1		
15				1				1		
16				1					1	
17			1					1		

18				1			1		
19				1				1	
20			1				1		
21				1				1	
22				1				1	
23				1				1	
24					1			1	
25					1				1
26				1				1	
27				1				1	
28			1				1		
29				1				1	
30				1			1		
31			1				1		
32				1				1	
33			1				1		
34				1				1	
35				1				1	
36					1				1
37			1				1		
38		1					1		
39		1				1			
40			1				1		
41			1				1		

Eil. Nr.	Ar programinio kodo vizualizavimas žaidimo pavidalu padeda geriau suprasti programavimo kalbą?					Ar programavimo žaidimas „Spacer“ padėjo įsivertinti Jūsų turimas programavimo kalbos žinias?		Ar manote, kad programavimo žaidimas „Spacer“ buvo naudingas jūsų edukacinėje veikloje?	
	Visiškai nesutinku	Nesutinku	Neturiu nuomonės	Sutinku	Visiškai sutinku	Taip	Ne	Taip	Ne
1				1		1		1	
2				1		1		1	
3			1			1		1	
4				1		1		1	
5			1				1		1
6			1			1		1	
7				1		1		1	
8					1	1		1	
9				1		1		1	
10			1			1		1	
11					1	1		1	
12				1		1		1	
13				1		1		1	
14			1			1		1	
15			1			1		1	
16			1			1		1	
17			1				1		1

18			1			1		1	
19				1		1		1	
20			1			1		1	
21				1		1		1	
22			1			1		1	
23				1		1		1	
24				1		1		1	
25					1	1		1	
26				1		1		1	
27				1		1		1	
28			1			1		1	
29				1		1		1	
30			1			1		1	
31				1		1		1	
32				1		1		1	
33			1			1		1	
34				1		1		1	
35				1		1		1	
36				1		1		1	
37				1		1		1	
38			1				1		1
39			1			1			1
40			1				1	1	
41			1				1		1

Eil. Nr.	Kaip manote, ar kompiuteriniai programavimo žaidimai būtų efektyvūs universitetinio lygio studijų praktinėse paskaitose?		Kaip manote, ar kompiuteriniai programavimo žaidimai yra efektyvus būdas studentui pasitikrinti turimas programavimo kalbos žinias ?		Skalėje nuo 1 iki 100 kiek buvote motyvuotas progresuoti žaidime?			
	Taip	Ne	Taip	Ne	0-25	25-50	50-75	75-100
1	1		1				1	
2	1		1				1	
3	1			1		1		
4	1		1			1		
5	1		1		1			
6	1		1			1		
7	1		1					1
8	1		1				1	
9	1		1				1	
10	1		1			1		
11	1		1				1	
12	1		1			1		
13	1		1			1		
14	1		1			1		
15	1			1			1	
16	1		1			1		
17		1		1	1			

18	1		1				1	
19	1		1			1		
20	1			1		1		
21	1		1			1		
22	1		1				1	
23	1		1			1		
24	1		1				1	
25	1		1				1	
26	1		1			1		
27	1		1				1	
28	1			1		1		
29	1		1			1		
30	1		1			1		
31	1			1		1		
32	1		1				1	
33	1				1		1	
34	1		1				1	
35	1		1				1	
36	1		1			1		
37	1		1			1		
38		1		1	1			
39	1		1		1			
40		1		1	1			
41	1		1			1		

Eil. Nr.	Ar vieša geriausių žaidėjų lentelė Jus labiau motyvuotų progresuoti žaidime?		Papildomi komentarai
	Taip	Ne	
1	1		Uždaviniai, kada reikia panaudoti konkretų įrankį, gerai padeda įsivertinti turimas žinias
2		1	
3	1		
4	1		
5	1		
6	1		Standartiniai programavimo uždaviniai taip pat efektyviai patikrina turimas žinias. Programavimo žaidimai tiesiog tą patį padaro vizualiai patrauklesnį, bet nebūtinai efektyvesnį.
7	1		
8	1		Butu idomu pamatyti žaidimus universiteto lygmenyje
9		1	
10	1		
11	1		Privertėte paieškoti informacijos internete
12	1		
13	1		
14	1		
15	1		Perejus žaidimą nežinau kiek laiko išliks informacija, nes žaidimas neitrauke
16	1		
17	1		

18	1		Gal ir būtų įdomu naudoti žaidimus per paskaitas
19	1		Tokie žaidimai panašiai kaip darbdavio paskirtos užduotys: (padaryk ta naudodamas ta ir ta) tuomet supranti ko nežinai ir ką reikėtų išmokyti, todėl tai geras būdas patikrinti
20	1		Svarbu kad vizualizacija būtų kokybiška
21		1	
22	1		
23	1		
24	1		
25		1	
26	1		
27	1		
28	1		Žaidimai turi būti gerai apgalvoti ir parinkti, kad būtų įdomūs ir efektyvūs
29	1		
30	1		
31	1		
32	1		
33	1		Kodo vizualizavimas padeda kai vizualizavimas pateiktas su kontekstu kodėl taip atvaizduotas kodas, tuomet logiškai galima suprasti programavimo kalbos panaudojimo galimybes ir ištraukti iš problemos sprendimą
34	1		
35	1		
36	1		
37	1		
38	1		
39		1	
40		1	
41		1	