****

**Kauno technologijos universitetas**

Informatikos fakultetas

**T000M250 Tiriamasis projektas 3**

**Dirbtinio intelekto metodų taikymas strateginiams žaidimams** Testavimo planas

|  |  |
| --- | --- |
|  |  |
| **IFM3/2 gr.stud. Tadas Laurinaitis**  Studentas  **Magistrinio projekto vadovas**  **Dr. Tomas Blažauskas** | (parašas) (data) |
|  |  |
| **Dr. Eglė Butkevičiūtė**  Dėstytojas/ dėstytoja | (parašas) (data) |
|  |  |

**Kaunas, 2024**

Turinys

[1. Įvadas 3](#_Toc181656444)

[Šiame dokumente aprašoma numatomos testavimo veiklos apimtis, testavimo metodas, ištekliai ir tvarkaraštis. Nurodomi testuojami elementai, testuojamos funkcijos, testavimo užduotys, kas atliks kiekvieną užduotį ir rizikos, reikalaujančios nenumatytų atvejų planavimo. 3](#_Toc181656445)

[1.1. Testavimo tikslai ir objektai 3](#_Toc181656446)

[1.2. Testavimo apimtis ir tipai 3](#_Toc181656447)

[1.3. Pagrindiniai apribojimai 4](#_Toc181656448)

[1.4. Nuorodos 4](#_Toc181656449)

[2. Testavimo planavimas 5](#_Toc181656450)

[2.1. Testuojama programų sistema 5](#_Toc181656451)

[2.1.1. Testuojama vartotojo sąsaja 5](#_Toc181656452)

[2.2. Testavimo strategija 5](#_Toc181656453)

[2.2.1. Funkcinis testavimas 5](#_Toc181656454)

[2.2.2. Vienetų testavimas 5](#_Toc181656455)

[2.2.3. Integracinis testavimas 5](#_Toc181656456)

[2.2.4. Sisteminis testavimas 6](#_Toc181656457)

[2.2.5. Našumo testavimas 6](#_Toc181656458)

[2.2.6. Vartotojo priėmimo testavimas 6](#_Toc181656459)

[2.3. Testavimo ištekliai 6](#_Toc181656460)

[2.4. Testavimo rezultatai 6](#_Toc181656461)

[2.5. Testavimo įrankiai ir aplinka 6](#_Toc181656462)

[2.6. Testavimo tvarkaraštis 6](#_Toc181656463)

[3. Testavimo procedūra (vykdymas) 7](#_Toc181656464)

[3.1. Testuojama programų sistema 7](#_Toc181656465)

[3.2. Testavimo procedūros 7](#_Toc181656466)

[3.2.1. Funkcinio testavimo vykdymas 7](#_Toc181656467)

[3.2.2. Vienetų testavimo vykdymas 8](#_Toc181656468)

[3.2.3. Integracinio testavimo vykdymas 9](#_Toc181656469)

[3.2.4. Sisteminio testavimo vykdymas 9](#_Toc181656470)

[3.2.5. Našumo testavimo vykdymas 10](#_Toc181656471)

[3.2.6. Vartotojo priėmimo testavimo vykdymas 10](#_Toc181656472)

[3.3. Testavimo išteklių paskirstymas 11](#_Toc181656473)

[3.4. Testavimo rezultatų kaupimas 12](#_Toc181656474)

[Informacijos šaltinių sąrašas 13](#_Toc181656475)

# Įvadas

## Šiame dokumente aprašoma numatomos testavimo veiklos apimtis, testavimo metodas, ištekliai ir tvarkaraštis. Nurodomi testuojami elementai, testuojamos funkcijos, testavimo užduotys, kas atliks kiekvieną užduotį ir rizikos, reikalaujančios nenumatytų atvejų planavimo.

## Testavimo tikslai ir objektai

Pagrindinis testavimo tikslas – užtikrinti kad kiekvienas kuriamos sistemos komponentas veikia tinkamai tiek vienas, tiek integruotas į komponentų visumą. Pagrindinis testavimo tikslas sudarytas iš šių dalių:

1. Žaidimo logika ir mechanikos – užtikrinama kad pagrindiniai žaidimo elementai (žaidėjo judėjimas, resursų valdymas ir žaidime vykstančios sąveikos tarp žaidėjo ir žaidimo) veikia nuosekliai, žaidėjui atliekant skirtingus veiksmus.
2. Dirbtinio intelekto (DI) elgsena – užtikrinama kad DI valdomi komponentai dinamiškai reaguoja į žaidėjo veiksmus, ir kad DI priimami sprendimai yra priimtinose normose. Šio testavimo dalys: DI prisitaikomumas, reakcijos laikas ir gebėjimas suvaldyti kompleksinius žaidimo scenarijus.
3. Žaidimo grafinės sąsajos elementų (UI) funkcionalumas – testuojami žaidimo UI elementai, tokie kaip meniu, meniu valdymo mygtukai ir žaidimo UI elementai rodantys tam tikrus žaidimo rodiklius, kad užtikrinti, jog šie atitinka žaidėjų patogaus naudojimo reikalavimus ir yra tiek greitai reaguojantys, tiek rodantys tikslią informaciją.
4. Žaidimo našumo metrikos – užtikrinamas žaidimo stabilumas, greitas atsako laikas ir tinkamas kompiuterio resursų paskirstymas, ypač intesyvių žaidimo scenarijų metu.

## Testavimo apimtis ir tipai

Naudojami testų tipai ir jų apimamos sritys:

1. Funkciniai testai – šio testavimo tipo metu pagrindinis dėmesys sutelktas į kiekvieno žaidimo funkcionalumo atitikimo reikalavimams patikrinimą. Į šį testavimą įtraukiama: DI intelekto atliekamų veiksmų veikimo, žaidimo mechanikų veikimo, žaidimo reagavimo į žaidėjo veiksmus, korektiškumo užtikrinimas.
2. Vienetų testai – dėmesys sutelkiamas į atskirus komponentus, tokius kaip atskiri metodai DI modeliuose/algoritmuose ir žaidimo logikoje. Šio testavimo metu siekiama patikrinti, kad kiekviena atskira dalis veikia teisingai, prieš integruojant jas į komponentų visumą.
3. Integraciniai testai – atlikus funkcinius testus, integraciniai testai patiktrins skirtingų žaidimo komponentų sąveiką. Šie testai yra itin svarbūs, kadangi kuriamame žaidime, kaip ir daugumoje kitų projektų, svarbu kad funkciniai vienetai veiktų teisingai, tačiau dar svarbiau, kad jie veiktų teisingai iš visumos perspektyvos.
4. Sisteminiai testai – šie testai skirti patikrinti, ar visa integruota žaidimo sistema atitinka iškeltus reikalavimus. Jų metu pagrindinis dėmesys skiriamas patikrinti, ar visi atskiri komponentai funkcionuoja kaip vienas, stabilią ir malonią patirtį žaidėjams teikiantis žaidimas.
5. Našumo testai – šių testų metu bus matuojami žaidimo reagavimo laikai ir stabilumas esant įvairioms apkrovoms. Į šį testavimą taip pat įtraukiami DI „stress“ testavimas pasitelkiant kompleksiškus žaidimo scenarijus, kad patikrinti ar šis veikia be strigimų ir trikdžų.
6. Vartotojo primėmimo testai (User acceptance testing) – žaidimo kūrimui artėjant prie pabaigos, pasitelkiant realius žaidėjus, šie testai patikrins žaidimo eigos ir patogumo atitikimą naudotojams. Testuotojų atsiliepimai bus surinkti galutinių patobulinimų atlikimui.

## Pagrindiniai apribojimai

Nors testavimo plane siekiama apimti visus kritinius projekto aspektus, kai kuris žaidimo funkcionalumas yra neįtraukiamas dėl žemesnio prioriteto ar tam tikrų išorinių ribojimų. Neįtrauktas funkcionalumas:

1. Ne kritiniai UI elementai – žaidimo tematiką atspindintys ir kiti vizualiniai elementai, nesantys svarbūs pagrindinių žaidimų funkcijų veikimui.
2. Išplėstinis DI prisitaikymas, nepapuolantis į pradinę apimtį – nors pagrindinis DI veikimas ir pritaikomumas bus ištestuotas pilnai, labai sudėtingo prisitaikymo elgesio testavimas automatiniais testais nebus prioritetizuojamas.
3. Veikimas kitose platformose – testavimo planas susitelkia ties sistemomis, veikiančiomis Windows 10 ir aukštesnėse. Veikimo kitose, Unity žaidimų variklį palaikančiose sistemose, testavimas nepapuola į šiame testavimo plane aprašytą apimtį, tačiau gali būti įtrauktas į tolimesnes fazes, esančias už šio projekto ribų.

## Nuorodos

**1 lentelė.** Nuorodos

| **Dokumentas** | Data | Dokumento autorius |
| --- | --- | --- |
| Projekto planas | 2022-12-09,  atnaujintas 2020-09-15 | Tadas Laurinaitis |
| Reikalavimų specifikacija | 2023-03-11 | Tadas Laurinaitis |
| Architektūros specifikacija | 2023-04-22 | Tadas Laurinaitis |

# Testavimo planavimas

## Testuojama programų sistema

Testuojama programinė įranga – strateginis žaidimas, naudojantis DI algoritmus ir modelius bei grafinę vartotojo sąsają (UI). Šis testavimo planas padengia pagrindines žaidimo mechanikas, DI algoritmų bei modelių elgseną ir būtiną UI funkcionalumą. Išimtys – pagrindinėms žaidimo mechanikoms nebūtinas ir jose nenaudojamas UI funkcionalumas ir komponentai, tokie kaip vizualiniai elementai ir temos.

### Testuojama vartotojo sąsaja

Į šią kategoriją įtraukiama pagrindiai meniu, interaktyvūs grafiniai žaidimo valdikliai, bei kritinę žaidimui informaciją perteikiantys skaitikliai.

## Testavimo strategija

Šio projekto testavimo strategija susideda iš keletos testavimo lygių, iš kurių kiekvienas turi savo atskirus tikslus, ir kurių bendra paskirtis yra užtikrinti, kad kiekvienas žaidimo aspektas yra nuodugniai patikrintas. Testavimo lygiai:

1. Funkcinis testavimas
2. Vienetų testavimas
3. Integracinis testavimas
4. Sisteminis testavimas
5. Našumo testavimas
6. Vartotojo priėmimo testavimas

### Funkcinis testavimas

Šio testavimo metu susitelkiama į užtikrinimą, kad kiekvienas žaidimo funkcionalumas veikia pagal reikalavimus. Funkcinis testavimas bus naudojamas tiek vienetų, tiek integracinių, tiek sisteminių testavimų metu.

### Vienetų testavimas

Izoliuojamos specifinės funkcijos ir metodai, papildomas dėmesys sutelkiamas į su žaidimo ar DI logika susijusias funkcijas ir metodus. Komponentai testuojami atskirai vienas nuo kito. Pasitelkiami automatiniai testavimo įrankiai – Xunit ir Pytest, vienetų testų proceso greitumui ir rezultatų vienodumui užtikrinti.

### Integracinis testavimas

Įvertinama kaip skirtingi komponentai sąveikauja. Pagrindinis dėmesys kreipiamas į žaidimo logiką ir DI integraciją. Siekiama patikrinti, kad vienas nuo kito priklausomi komponentai koordinuoja teisingai ir kad funkcionalumo kombinacijos veikia prideramai tarpusavyje.

### Sisteminis testavimas

Testuojamas visas žaidimas, siekiama patikrinti ar visi komponentai dera ir veikia vienoje visumoje, ir ar žaidimas atitinka keliamus reikalavimus. Testavimo metu žiūrimas funkcionalumas, stabilumas ir tikslios žaidimo reakcijos į žaidėjo veiksmus.

### Našumo testavimas

Siekiama įvertinti žaidimo kaip sistemos greitį, stabilumą ir reagavimo laiką, ypač didelės apkrovos metu. Šio testavimo tikslas rasti galimas kliūtis ir delsos problemas, šitaip užtikrinant, kad žaidimai po pataisymų veiks gerai, be trikdžių ir strigimų.

### Vartotojo priėmimo testavimas

Dalyvauja realūs žaidėjai. Užtikrinama, kad galutinis produktas atitinka žaidėjų lūkesčius ir keliamus reikalavimus. Po šio testavimo bus atliekami paskutiniai patobulinimai pagal gautą testuotojų atsaką.

## Testavimo ištekliai

Testavimui bus reikalingi šie ištekliai:

1. Programinės įrangos ištekliai – Unity žaidimų variklis, Visual Studio 2022 Community, Github, git versijos kontrolės sistema.
2. Aparatūrinės įrangos ištekliai – didelion našumo kompiuteris, turintis 32GB RAM, 1TB vietos turintį SSD, Ryzen 7 5600X procesorių ir Nvidia RTX 3080 Ti vaizdo plokštę.

## Testavimo rezultatai

Testavimo rezultatai bus kaupiami toje pačioje git repozitorijoje kaip ir žaidimo kodas ir su projektu susiję dokumentai.

## Testavimo įrankiai ir aplinka

Automatizuoto testavimo įrankiai bus naudojami vienetų, integracinių ir našumo testų metu:

1. Unity testų karkasas.
2. Pytest.

Testavimas vyks Windows 10 operacinės sistemos aplinkoje.

## Testavimo tvarkaraštis

Testavimo tvarkaraštis:

Funkcinis ir vienetų testavimas – 1-3 projekto mėnuo, taip pat pagal poreikį.

Integracinis testavimas – 2 - 6 projekto mėnuo

Sisteminis testavimas – 6 - 8 projekto mėnuo

Našumo testavimas – 8 projekto mėnuo

Vartotojo priėmimo testavimas – 9 projekto mėnuo

# Testavimo procedūra (vykdymas)

## Testuojama programų sistema

Testuojama programinė įranga – strateginis žaidimas, naudojantis DI algoritmus ir modelius bei grafinę vartotojo sąsają (UI). Šis testavimo planas padengia pagrindines žaidimo mechanikas, DI algoritmų bei modelių elgseną ir būtiną UI funkcionalumą. Išimtys – pagrindinėms žaidimo mechanikoms nebūtinas ir jose nenaudojamas UI funkcionalumas ir komponentai, tokie kaip vizualiniai elementai ir temos.

## Testavimo procedūros

Šiame skyriuje aprašomos tikslesnės procedūros kiekvienam testavimo tipui, nurodytam 2.2 skyriuje. Kiekvienoje sekcijoje nurodomi testavimo tikslai, specifiniai veiksmai, laukti rezultatai ir trumpi aprašymai.

### Funkcinio testavimo vykdymas

2 lentelė. Funkcinio testavimo vykdymas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testo nr. | Funkcija | Tikslas | Testuojantis veiksmas | Laukiamas rezultatas |
| 1 | Judėjimo sistema | Patikrinti veikėjų judėjimo funkcionalumą | Veikėjams duodamas nurodymas judėti | Veikėjai juda tiksliai pagal valdikliu duodamus nurodymus |
| 2 | Resursų valdymo sistema | Patikrinti resursų sistemos funkcionalumą | 1. Veikėjams duodamas nurodymas rinkti resursus.  2. Statyti pastatus.  3. Treniruoti naujus veikėjus | Resursų kiekis kinta atitinkamai, pagal padarytus veiksmus. |
| 3 | Kovos sistema | Patikrinti kovos sistemos funkcionalumą | Veikėjams duodamas veiksmas pulti priešininkų valdomus veikėjus | Veikėjas puola priešininko valdomus veikėjus |
| 4 | Interakcija su DI | Patikrinti ar DI valdomų veikėjų veiksmai prisitaiko prie žaidėjo veiksmų | Žaidžiamas žaidimas | DI valdomų veikėjų veiksmai prisitaiko prie žaidėjo veiksmų |
| 5 | DI kelio radimo algoritmas | Patikrinti DI valdomą kelio radimą | Žaidžiamas žaidimas | DI valdomų veikėjai teisingai randa kelią į reikiamas vietas |

### Vienetų testavimo vykdymas

3 lentelė. Vienetų testavimo vykdymas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testo nr. | Komponentas | Tikslas | Testuojantis veiksmas | Laukiamas rezultatas |
| 1 | Resursų valdymo sistemos skaičiavimai | Patikrinti resursų sistemos skaičiavimus | 1. Simuliuojamas veikėjams duodamas nurodymas rinkti resursus.  2. Simuliuojamas pastatų statymas.  3. Simuliuojamas naujų veikėjų treniravimas | Resursų kiekis kinta atitinkamai, pagal padarytus veiksmus. |
| 2 | Kovos sistema | Patikrinti kovos sistemos skaičiavimus | Simuliuojamas veikėjams duodamas veiksmas pulti priešininkų valdomus veikėjus | 1. Veikėjas daro žalą priešininkų valdomiems veikėjams.  2. Veikėjas gauna žalą iš priešininkų valdomų veikėjų. |
| 3 | DI kelio radimo algoritmas | Patikrinti DI valdomo kelio radimo algoritmo skaičiavimus | Simuliuojamas priešininko veikėjo valdymo nurodymo veiksmas | DI valdomų veikėjai teisingai randa kelią į reikiamas vietas |

### Integracinio testavimo vykdymas

4 lentelė. Integracinio testavimo vykdymas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testo nr. | Modulis | Tikslas | Testuojantis veiksmas | Laukiamas rezultatas |
| 1 | Žaidimo logikos integracija su DI | Patikrinti sklandžią DI integraciją žaidimo logikoje | Simuliuojamas žaidėjo valdomų veikėjų veiksmai, žiūrimas DI atsakas. | DI intelektas tiksliai reaguoja į kintančią žaidimo būseną. |
| 2 | Resursų valdymo sistemos sąveika su grafine vartotojo sąsaja (UI) | Patikrinti ar resursų sistemos apdorojami pokyčiai atsispindi UI | Daromi veiksmai, keičiantys resursų kiekius | UI rodomi resursų kiekiai kinta atitinkamai nuo atliktų veiksmų |
| 3 | Veikėjų valdymo sistemos sąveika su grafine vartotojo sąsaja (UI) | Patikrinti ar valdomų veikėjų veiksmų pokyčiai atsispindi UI | Simuliuojamas veikėjo valdymo nurodymo veiksmas | UI rodomi veikėjų atliekami veiksmai |

### Sisteminio testavimo vykdymas

5 lentelė. Sisteminio testavimo vykdymas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testo nr. | Funkcija | Tikslas | Testuojantis veiksmas | Laukiamas rezultatas |
| 1 | Žaidimo sesija | Patikrinti žaidimo stabilumą | Sužaidžiama pilna žaidimo sesija | Žaidimas veikia be klaidų. |
| 2 | Žaidimo stadijos nekintamumas tarp sesijų | Patikrinti ar žaidimo duomenys išsaugomi baigus sesiją ir užkraunami norint toliau tęsti jau pradėtą sesiją | Išsaugomas žaidimas, uždaromas žaidimas, įjungiamas žaidimas per nauja ir užkraunama praeita žaidimo sesija | Žaidimas užsikrauna su teisinga, išsaugota sesija |

### Našumo testavimo vykdymas

6 lentelė. Našumo testavimo vykdymas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testo nr. | Aspektas | Tikslas | Testuojantis veiksmas | Laukiamas rezultatas |
| 1 | Atsako laikas | Užtikrinti greitą žaidimo atsako laiką | Matuojama įvairių žaidimo veiksmų atlikimo trukmė | Žaidimo atsako laikas atitinka reikalavimus (1 sekundė) |
| 2 | Didelė apkrova | Patikrinti žaidimo stabilumą esant didelei apkrovai | Simuliuojamas didelis kiekis žaidėjo ir DI atliekamų veiksmų su daug veikėjų | Žaidimas išlieka stabilus ir reaguojantis |
| 3 | Naudojamų sisteminių resursų kiekis | Išmatuoti CPU ir GPU apkrovą, bei sunaudojamą RAM kiekį žaidimo metu | Matuojami sisteminiai resursai ilgos sesijos metu | Sunaudojamų resursų kiekis išlyka numatytose ribose |

### Vartotojo priėmimo testavimo vykdymas

7 lentelė. Vartotojo priėmimo testavimo vykdymas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testo nr. | Aspektas | Tikslas | Testuojantis veiksmas | Laukiamas rezultatas |
| 1 | Žaidimo sesija | Patikrinti žaidimo atitikimą žaidėjų reikalavimams ir lūkesčiams | Sužaidžiama pilna žaidimo sesija | Žaidėjas patvirtina, kad žaidimas atitinka jo keliamus reikalavimus ir lūkesčius |

## Testavimo išteklių paskirstymas

8 lentelė. Ištekliai, kurių reikės kiekvienam testavimo etapui.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testavimo tipas | Žmogiškieji ištekliai | Programinė įranga | Testavimo aplinka | Apskaičiuota trukmė |
| Funkcinis testavimas | 1 programuotojas | Unity editor, rankinio testavimo įrankiai | Programuotojo kompiuteris | 1 savaitė, nuolatinis |
| Vienetų testavimas | 1 programuotojas | XUnit, Visual Studio 2022 Community, Unity editor, Pytest | Programuotojo kompiuteris | 1 savaitė, nuolatinis |
| Integracinis testavimas | 1 programuotojas | XUnit, Visual Studio 2022 Community, Unity editor | Programuotojo kompiuteris | 1-2 savaitės |
| Sisteminis testavimas | 1 programuotojas | Unity editor ir Unity žaidimų variklis | Programuotojo kompiuteris | 2-3 dienos |
| Našumo testavimas | 1 programuotojas | Unity editor ir Unity žaidimų variklis | Programuotojo kompiuteris | 1-2 dienos |
| Vartotojo priėmimo testavimas | 10 žaidėjų grupė | Žaidimo versija, paruošta žaidėjų testavimui | Žaidėjo kompiuteris | 1 savaitė testavimui, 1 savaitė rezultatų surinkimui ir analizei bei reikiamiems pakeitimams |

## Testavimo rezultatų kaupimas

9 lentelė. Testavimo rezultatų kaupimas.

|  |  |  |  |
| --- | --- | --- | --- |
| Testavimo tipas | Renkami duomenys | Saugojimo vieta | Saugojimo periodas |
| Funkcinis testavimas | Testuojamų sąlygų rezultatai, ekrano iškarpos | Programuotojo kompiuteris, Git repozitorija | Projekto laikotarpis + 1 mėnesias |
| Vienetų testavimas | Testavimo logai, kodo padengimo ataskaitos | Programuotojo kompiuteris, Git repozitorija | Projekto laikotarpis |
| Integracinis testavimas | Testavimo logai, klaidų ataskaitos | Programuotojo kompiuteris, Git repozitorija | Projekto laikotarpis + 6 mėnesiai |
| Sisteminis testavimas | Sisteminiai logai, našumo metrikos | Programuotojo kompiuteris, Git repozitorija | Projekto laikotarpis + 6 mėnesiai |
| Našumo testavimas | Našumo metrikos bei testinių duomenų rinkiniai | Programuotojo kompiuteris, Git repozitorija | Projekto laikotarpis + 1 metai |
| Vartotojo priėmimo testavimas | Žaidėjų atsiliepimai, suformuota ataskaita | Programuotojo kompiuteris, Git repozitorija | Projekto laikotarpis + 1 metai |

Informacijos šaltinių sąrašas

1. Mano pačio darbai, atlikti magistro 1 ir 2 kurso metu.