

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**P175B015 PROGRAMŲ SISTEMŲ INŽINERIJA**

**VAIDMENŲ ŽAIDIMAS**

**„ILLOGICAL“**

|  |  |
| --- | --- |
|  | **Atliko:**  Karolis Venckus IFF – 6/7  Edvinas Sadlevičius IFF – 6/7 |
|  | **Vadovas :**  Ušaniov Andrej |

Turinys

[1. Sprintai 4](#_Toc515222719)

[1. Sprintų iškarpos 4](#_Toc515222720)

[2. Sprintų Burndown grafikai 4](#_Toc515222721)

[3. Sprintų Retrospektyvos rezultatai 5](#_Toc515222722)

[2. Testavimas 5](#_Toc515222723)

[3. Priedai 7](#_Toc515222724)

# Sprintai

## Sprintų iškarpos

## Sprintų Burndown grafikai

pav. 1 Antro sprinto grafikas

pav. 2 Pirmo sprinto grafikas

## Sprintų Retrospektyvos rezultatai

Darydami trečią sprintą pagrinde dirbome su UI ir lygių sistema, dauguma dalykų, kuriuos planavome, pavyko įdėti, tačiau kaip žaidime veikėjas turėtų kautis prieš esančius priešus nepavyko sugalvoti. Ketvirtame sprinte nusprendėme padaryti daug dalykų, tačiau kaip matosi iš grafiko nespėjome, nes gauti modelį priešui ir sukurti jo AI pasirodė kiek sunkesnė užduotis nei tikėjomės.

# Testavimas

**Testavimo tikslas ir aplinka**

Norėdami patikrinti kaip žaidimas veikia, ar atitinka reikalavimus, mes žaidimą testavome. Vienetų testavimus atlikome su Unity suteiktais įrankiais, geresniam efektui, tikrinome vienas kito sukurtus skriptus ir žaidimo funkcijas. Žaidimas buvo testuojamas ant kelių kompiuterių su įdiegtais Windows 7 ir Windows 10.

Žaidimo kodas bus kompiliuojamas su standartiniu kompiliatoriumi esančiu Unity. Vartotojam nereikės kompiliuoti kodo.

**Testavimo atvejai:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testo nr. | Testo tikslas | Testo aprašymas | Tikėtini rezultatai | Gauti rezultatai |
| 1. | Patikrinti ar Editoriuje galima padėti kelis koridorius vieną ant kito | Bus bandoma padėti koridorių ant kito koridoriaus, naudojant skirtingas koridorių rūšis ir skirtingas koordinates. | Paspaudus nieko neįvyks žaidėjui, o į „Debug“ konsolę bus gauta žinutė, informuojanti, kad ta koordinatė nėra tuščia | Įvyko kaip tikėtasi |
| 2. | Patikrinti ar surinkti taškai išlieka, jei grįžtama atgal į Editorių neįveikus labirinto | Lygyje bus surinkta tam tikras kiekis taškų ir užuot ėjus iki finišo, bus išeita iš lygio atgal į Editorių. Vėl grįžus į lygį taškų skaičius turėtų būti rodomas 0 ir įveikus lygį turėtų būti išsaugomi tik to lygio taškai. | Išėjus iš labirinto neįveikus lygio, currentScore bus grąžinamas į nulį ir prie galutinių taškų nebus nieko pridėta. | Įvyko kaip tikėtasi |
| 3. | Patikrinti ar metant savo ginklą įmanoma išmesti daugiau negu vieną | Bus bandoma kuo greičiau spausti tam, kad galbūt atsirastų bent kadras, kuriame galima išmesti dar vieną ginklą, negrįžus dar senesniam. | Išmetus ginklą, išmesti dar vieno neišeis, nes ginklo būsena bus nustatyta THROWN, o išmesti ginklą galima tik tada kai jo būsena INHAND. Ginklą bus galima išmesti tik jį vėl atgal susigrąžinus. | Įvyko kaip tikėtasi |
| 4. | Patikrinti ar Meniu(iškviestos žaidimo metu), mygtukas „Go Back to Editor„ grąžina žaidėja atgal į Editorių be klaidų | Bus paspausta ant „Go Back to Editor“ mygtuko ir tada bus paspausta and „Generate“. | Paspaudus ant „Go Back to Editor“ mygtuko visas lygis ir veikėjo modelis ištrinami, grįžus atgal perkuriamas visas lygis, neturi likti nei užsilikusių monetų, koridorių, priešų. | Įvyko kaip tikėtasi |
| 5. | Patikrinti ar pridėjus ir išėmus daiktus Inventoriuje keičiasi ir duomenų sąrašas, kuriame saugoma turimų daiktų informacija | Bus pridėta daiktas, patikrinta ar išjungus ir įjungus inventorių tas daiktas liko, patikrintas žaidėjo turimų daiktų sąrašas(žaidėjas to padaryt negali), tas pats bus pakartota ištrynus daiktą. | Pridėjus daiktą ir išjungus, įjungus Inventorių, tas daiktas bus rodomas Inventoriuje, o sąraše atsiras naujas įrašas su šio daikto informacija, tas pats bus ir ištrynus tą daiktą. | Įvyko kaip tikėtasi |
| 6. | Patikrinti ar gyvybėm nukritus iki arba žemiau 0, žaidėjas gauna žinutę, kad mirė, bei perkeliamas į Editorių | Žaidėjas leidžia nukristi gyvybėm iki nulio. | Žaidėjo gyvybėms pasiekus nulį, išmetama žinutė, kad žaidėjas mirė, gyvybės grąžinamos į maksimumą ir žaidėjas atsiduria Editoriuje iš kurio vėl gali pradėti lygį. | Įvyko kaip tikėtasi |
| 7. | Patikrinti ar Inventoriuje informacija apie taškus atsinaujina išjungus Inventorių. | Surenkamas tam tikras kiekis taškų, patikrinamas Inventorius, išjungiamas lygis ir vėl įjungiamas, patikrinama kiek dabar taškų yra. | Inventoriuje turi būti rodomi turimi taškai šiame lygyje ir aplamai surinkti taškai per visą žaidimą. Kiekvieną kartą atidarius inventorių rodomi taškai turi būti atnaujinami | Įvyko kaip tikėtasi |
| 8. | Patikrinti ar palikus atvirų vietų koridoriuje, nelieka tuščių vietų, pro kurias galima išeiti iš lygio. | Sudedami koridoriai taip, kad kiekvienos rūšies koridoriai būtų palikti aklini. | Aklinos vietos užstatomos juodomis sienomis, pro kurias žaidėjas negali praeiti, iš tų juodų sienų eina priešai. | Įvyko kaip tikėtasi |
| 9. | Patikrinti ar sudedant koridorius tam tikra tvarka(raštu) sukuriamas atitinkamas kambarys | Sudedami koridoriai tokia tvarka, kokia reikia, kad vietoje jų būtų generuojamas kambarys | Vieta, kurioje buvo padarytas raštas, koridoriai ištrinami ir įdedamas kambarys jų vietoje, koridorių duomenų masyve matomas kambario id. | Neįvyko kaip tikėtasi |
| 10. | Patikrinti ar įjungus inventorių priešai vistiek gali pulti žaidėją | Būnant šalia priešo įjungiamas inventorius | Žaidėjo neturėtų būti įmanoma sužeisti, nes įjungiama inventoriaus būsena, o tai reiškia, kad viskas kas vyksta žaidime sustingdoma ir žaidėjo modelis laikinai neegzistuoja | Įvyko kaip tikėtasi |

Beveik visų testų rezultatai gavosi teigiami, nes kuriant žaidimą buvo pastoviai testuojama ir žaidimą bandė daug žmonių, klaidos buvo taisomos ir kokie nors neatitikimai šalinami. Nepavyko tik įgyvendinti kambarių generavimo idėjos, nes per mažai laiko buvo.

# Priedai

URL ir prisijungimo duomenys į projekto valdymo įrankį:

<https://app.vivifyscrum.com/boards/20811>

[dirbantisasmuo@gmail.com](mailto:dirbantisasmuo@gmail.com)

ktuslaptazodis123

URL į projekto kodo saugyklą:

<https://github.com/ConsciousTank/Projektas_KTU>

Testavimo kodai:

using UnityEngine;

using UnityEngine.TestTools;

using NUnit.Framework;

using System.Collections;

public class GeneratorTests {

[UnityTest]

public IEnumerator Adds\_Room() {

int x = 0;

int y = 0;

int rotation = 0;

int id = 0;

var generator = new GameObject().AddComponent<Generator>();

generator.SetStartingEndingPoints(1, 1, 1, 1);

yield return null;

Generator.AddRoom(x, y, rotation, id);

Assert.AreEqual(false, Generator.isEmpty(x, y));

Assert.AreEqual(rotation, Generator.checkRoomRotation(x,y));

Assert.AreEqual(id, Generator.checkRoomId(x, y));

}

[UnityTest]

public IEnumerator Checks\_If\_Room\_Is\_Empty()

{

int x = 0;

int y = 0;

var generator = new GameObject().AddComponent<Generator>();

generator.SetStartingEndingPoints(1, 1, 1, 1);

yield return null;

Generator.RemoveRoom(x, y);

Assert.AreEqual(true, Generator.isEmpty(x, y));

}

[UnityTest]

public IEnumerator Checks\_If\_Can\_Add\_In\_Player\_Room()

{

int x = 1;

int y = 1;

int rotation = 0;

int id = 0;

var generator = new GameObject().AddComponent<Generator>();

generator.SetStartingEndingPoints(1, 1, 1, 1);

yield return null;

var add = Generator.AddRoom(x, y, rotation, id);

Assert.AreEqual(false, add);

}

[TearDown]

public void AfterEveryTest()

{

var generator = new GameObject().AddComponent<Generator>();

Generator.ClearRooms();

}

}

Backlog:



