

TALLINNA ÜLIKOOL

Haapsalu kolledž

Rakendusinformaatika

Taavi Ansper, Toivo Pärnpuu, Taavi Rannu, Andero Samelselg

TUDENGIRALLI

Multimeediumi praktika aruanne

Juhendajad: Laura Hein, Mihkel Pulst, Martti Raavel, Andrus Rinde

Haapsalu 2023

SISUKORD

SISSEJUHATUS	3
1. MÄNGU KONTSEPTSIOON	4
2. MEESKONNATÖÖ ÜLEVAADE	5
2.1. Suhtluskanalid ja projekt	5
2.2. Dokumenteerimine	5
2.3. Tööjaotus	5
3. ARENDUS / MÄNGUMOOTOR	7
4. GRAAFIKA	8
4.1. Graafika laad	8
4.2. Tööjaotus	8
4.3. Tööde järjekord	11
4.4. Hinnang töö käigule	12
5. KASUTAJALIIDES	13
6. HELI	15
6.1. Helide loomine	15
6.2. Muusikaline taust	15
6.3. Helide maha mängimise lahendus	15
KOKKUVÕTE	17

SISSEJUHATUS

Käesolev aruanne kirjeldab Multimeediumi praktikumi õppeaine raames rühmatööna loodud mängu nimega Tudengiralli, mis on pikselgraafikas kujundatud 2D-platvormimäng. Aruanne kirjeldab mängu kontseptsiooni, annab ülevaate meeskonnatöö ülesehitusest, mängus kasutatud mängumootorist, mängu graafikast, kasutajaliidesest ning mängu helidest.

Tiimiliikmed:

Toivo Pärnpuu – rühmajuht ja arendaja.

Taavi Rannu – arendaja.

Taavi Ansper – disainer.

Andero Samelselg – disainer.

1. MÄNGU KONTSEPTSIOON

Tudengiralli on 2D-platvormimäng, kus mängija juhib tudengi tegelaskuju läbi tasemel olevate takistuste. Mäng on loodud pikselgraafika stiilis, pakkudes nostalgilist ja unikaalset visuaalset kogemust. Mängu keskne teema on tudengi teekond läbi ülikooli semestrite.

Mängu eesmärk on jõuda algusest finišisse võimalikult kiiresti, läbides kõik vahepunktid ja kogudes neis kõik tasemel leiduvad esemed. Finiš avaneb ainult siis, kui kõik vahepunktid on edukalt läbitud. Mõned vahepunktid nõuavad batuutide kasutamist, et tegelane jõuaks piisavalt piisavalt kõrgele, mis lisab mängule lisaväljakutseid.

Tudengiralli koosneb kahest tasemest. Esimene tase on õppetase, mis on loodud selleks, et tutvustada mängijatele mängu põhifüüsikat ja -oskusi. See sisaldab lihtsamaid takistusi ja väljakutseid, et mängijad saaksid mänguga harjuda.

Teine tase on põhiline mängutase. See on oluliselt pikem ja keerukam kui õppetase, sisaldades rohkem takistusi ja ülesandeid. Teisel tasemel peavad mängijad kasutama esimese taseme käigus omandatud oskusi, et edukalt finišisse jõuda. See tase nõuab mängijalt head ajastust, planeerimist ja osavust, pakkudes mängijatele suuremaid ja rohkem rahuldust pakkuvamaid väljakutseid.

Mängu idee tuli ühise ajurünnaku käigus. Otsustasime siduda mängu süžee ülikooli läbimisega, kuna taoline mõte tuli kahel tiimiliikmel. Otsustasime, et tegu võiks olla 2D-platvormimänguga, kuna Taavi Rannul oli varasemalt kogemusi sarnaste mängude loomisega.

2. MEESKONNATÖÖ ÜLEVAADE

Käesolev peatükk annab ülevaate, kuidas oli organiseeritud meie meeskonnatöö.

2.1. Suhtluskanalid ja projekt

Kuna meie meeskonnas puudus varasem kogemus taolise töökorraldusega, siis kogemus oli kõigi meeskonnaliikmete jaoks väärtuslik ja uus. Projekti ülesandest tulenevalt olid meie meeskonna arendussprindid kahe nädala pikkused. Selle aja jooksul pidas meeskond vähemalt kord nädalas virtuaalselt Discordi platvormil häälkõnesid, et arutada projektiseisu, jooksvaid probleeme ja küsimusi. Need kohtumised olid olulised, et hoida meeskonna kommunikatsioon selge, tagada ühine arusaam arendusprotsessist ja tagada ülesannete tõhus ja õigeaegne täitmine. Ülejäänud aja jooksul tegelesid meeskonna liikmed oma tööloiguga ning jagasid oma edenemist ja muresid Discordi tekstikanalis. Projekti käigus omandasime oskuse kirjutada tööülesandeid viisil, et oleks üheselt mõistetav, millal ülesanne on täidetud ning selleks, et kajastada progressi, täiendasime jooksvalt ülesande sisu vastavalt töö edenemisele.

2.2. Dokumenteerimine

Enda töö dokumenteerimisel lähtusime nõudest, et kõik peab olema olemas GitHubi repositooriumis. Vajalikke töid ja vaheetappe hakkasime haldama kohealt GitHub Issues all, mis sai omakorda seotud Project mooduliga.

Algselt pidasime koosolekute protokolle, kuid hiljem otsustasime need viia Github Issues alla. See võimaldas meil paremini jälgida meeskonna mõtteid, arutelusid ja ideid, lisaks oli info kõigile meeskonnaliikmetele kiiremini kättesaadav. See muutus võimaldas meil tõhustada meeskonna kommunikatsiooni ja parandas meie töökorraldust, aidates kaasa projekti üldisele edukusele.

2.3. Tööjaotus

Meie arendusmeeskond koosnes neljast inimesest, kellel olid erinevad rollid ja ülesanded, allpool olev loend annab neist täpsema ülevaate:

- **Toivo Pärnpuu.** Toivo vastutas arenduse ja projektijuhtimise eest. Tema peamine ülesanne oli luua vajalikud helifailid ja integreerida need mängu. Lisaks sellele oli ta vastutav projektijuhtimise eest, tagades, et kõik arendustööd oleksid kooskõlas ajakava ja eesmärkidega.
- **Taavi Rannu.** Taavi oli peamine mängu arendaja, kuna omas varasemat Unity kasutamise kogemust. Tema keskendus peamiselt mängutasemete koostamisele ning mängu füüsika ja takistuste loomisele. Tema tööd mõjutasid otseselt mängukogemuse kvaliteeti ja mängitavust.
- **Andero Samelselg.** Andero vastutas kasutajaliidese loomise eest. Ta kujundas kasutajaliidese ning lõi batuudi ja kogutava eseme (raamatu) disaini ning osa tegelase

animatsioonidest (jalutamine ja *dashimine*). Tema töö oli oluline, et tagada mängu kasutusmugavus ja visuaalne atraktiivsus.

- **Taavi Ansper.** Taavi oli meeskonna graafilise disaini spetsialist. Ta lõi mängu peategelase kujunduse, tasemete taustagraafika ja osa tegelase animatsioonidest (hüppamine). Tema roll oli oluline mängu visuaalse stiili ja iseloomu loomisel.

Iga meeskonnaliige andis olulise panuse projekti arengusse ja lõpptulemusse, kasutades oma erialaseid oskusi ja teadmisi. Projekti käigus esines momente, kus kõik tiimi liikmed ei saanud kaasa lüüa lubatud tegevustes, kuid üldises plaanis õnnestus need olukorrad lahendada ja tekkinud probleemid leidsid lahenduse.

3. ARENDUS / MÄNGUMOOTOR

Arendusprotsessi käigus otsustasime kasutada Unity mängumootorit. Valik põhines mitmetel kaalutlustel. Esiteks, üks meie meeskonna liikmetest, Taavi Rannu, on varem Unity mängumootoriga töötanud. Unity pakub tugevat 2D-füüsikamootorit ja kvaliteetset valgustuse süsteemi, mis on meie projekti jaoks oluline. Lisaks on Unityl laialdane valik õppematerjale ja juhendeid, mis võimaldavad kiiresti lahendada tekkinud probleeme. Sisseehitatud animeerimistööriistad muudavad disainerite töö lihtsamaks ja tõhusamaks.

Teise valikuna kaalusime GODOT mängumootorit, mis on algajatele kasutajasõbralik ja kompaktsem kui Unity või Unreal Engine. GODOT'1 on palju juhendeid, mis muudavad selle kasutamise lihtsaks.

Unreal Engine on võimekas, kuid ei vastanud meie projektinõuetele. Unreal Engine pakub 2D võimalusi, kuid need ei ole nii tugevad kui teistes mängumootorites. Lisaks on Unreal Engine keerukam platvorm, mis ei sobi meie projektiga.

Phaser ei ole tegelikult mängumootor, vaid pigem raamistik. Siiski on Phaseril oma eelised, näiteks kompaktsus ja stabiilsus. Phaseriga töötamine on selge ja ei tekita segadust. Kuid Phaseri suur puudus on animeerimise võimalused - keerulisi animatsioone on raske saavutada ja need nõuavad sageli koodimist.

Mängumootor	Unity	GODOT	Unreal Engine	Phaser
Eelistatud dimensioonid	2D/3D	2D	3D	2D
Funktsioonide kogus	Suur	Keskmine	Suur	Väike
Õppematerjalide kogus	Suurepärane	Hea	Hea	Keskpärane
Keerulisus algajale	Keskmine	Kerge	Raske	Keskmine
Animeerimislahendused	Spritesheet, Recording	Spritesheet, Recording	Spritesheet, Recording	Spritesheet
Programmeerimiskeeled	C#, <i>visual scripting (drag and drop)</i>	GDScript, C#, C++	C++, Blueprints	Javascript, Typescript
Meeskonnaliikmete varasem kogemus	Laialdane	Puudub	Puudub	Puudub

Tabel 1. Mängumootorite võrdlus

4. GRAAFIKA

Käesolev peatükk kirjeldab Tudengiralli erinevaid graafilisi elemente ja otsuseid nende taga.

4.1. Graafika laad

Graafika laadiks valisime pikslikunsti stiili, sest see sobis meie mängu 2D-platvormer stiiliga. Sama stiili kasutab ka videomäng *Super Mario Bros.* (Joonis 1), mis oli projekti alguses graafiliseks inspiratsiooniks, kuid pikslikunst on selles platvormermängude žanris üldse väga levinud stiil. Seda kasutavad näiteks 1980. aastate mängud *Metroid* ja *Castlevania*, aga ka uuemad mängud nagu *Celeste*, *Katana Zero* ja *Insignia*. Enda töös otsustasime, et skaala huvides on tegelase suurus 32x32 pikslit, mis lubas piisavalt suure ala elementide väljajoonistamiseks, samas hoides asju lihtsana ja loomulikus suurustes.



Joonis 1. Kuvatõmmis esialgseks inspiratsiooniks olnud mängust *Super Mario Bros.*

4.2. Tööjaotus

Mängusisest graafikat tegi peamiselt Taavi Ansper, kes lõi tegelase (Joonis 2), esiplaani (Joonis 3) ja taustaplaani plaadikomplektid (Joonis 4) ja koostas kaadrid tegelase hüppamise animeerimiseks (Joonis 5). Kasutajaliidese poole eest vastutas Andero. Lisaks tegi ta

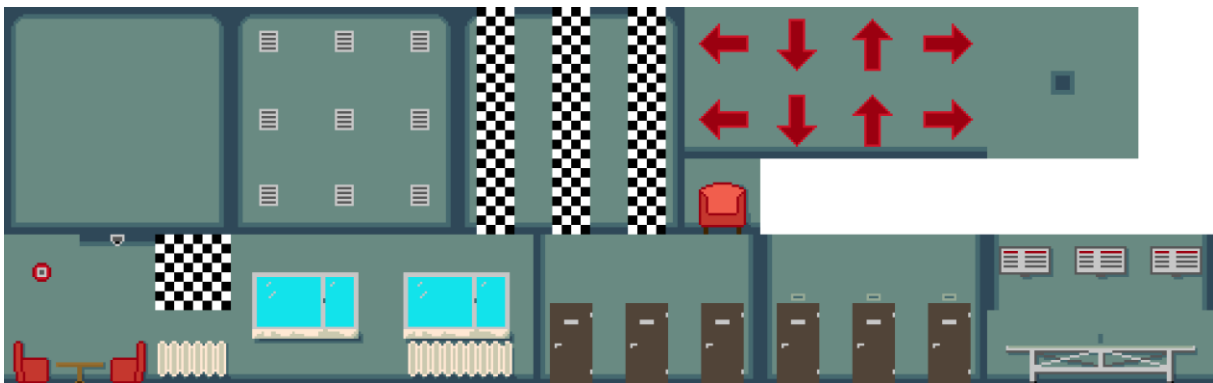
batuudi/*bumper*’i disaini, samuti kasutasime tema loodud raamatu kujundust üleskorjatava esemena (Joonis 6). Peale selle koostas ta nii *dashimise* animatsiooni kaadrid (Joonis 7) kui ka tegelase jalutamise kaadrid (Joonis 8). Kaadrid pani kokku animatsiooniks mängumootoris Taavi Rannu.



Joonis 2. Suurendatud tegelase ees- ja külgvaade mänguseina taustal



Joonis 3. Esiplaani plaadikomplekt



Joonis 4. Tagaplaani plaadikomplekt



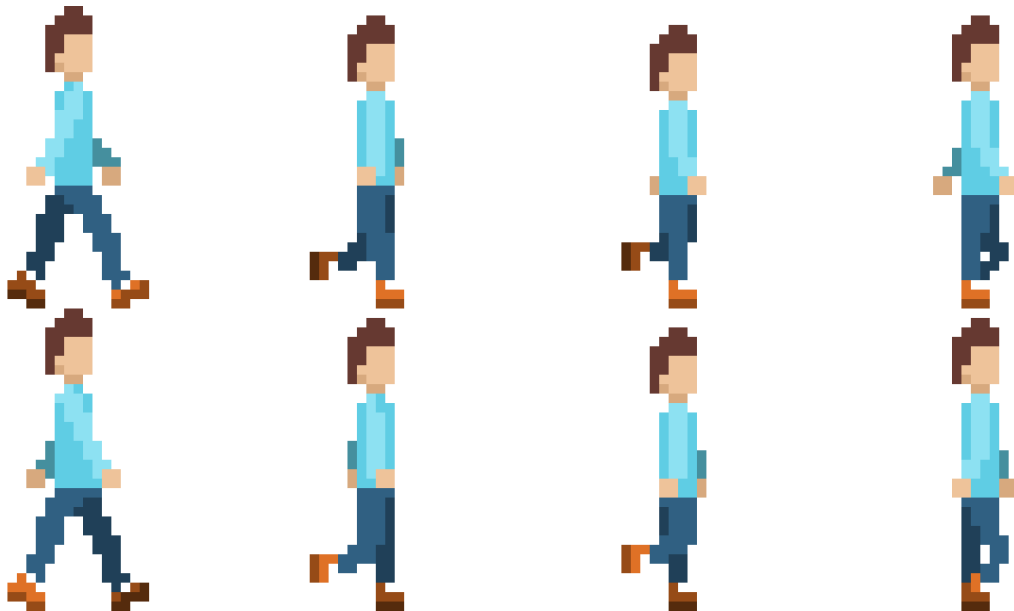
Joonis 5. Hüppamise animatsiooni kaadrid (tavaolek, tõus, õhusolek, langemine)



Joonis 6. Üleskorjatav raamat



Joonis 7. Tegelase *dashimise* animatsiooni kaadrid



Joonis 8. Tegelase jalutamise animatsiooni kaadrid

4.3. Tööde järjekord

Alustasime endale sobiva tarkvara otsimisega, mille jaoks katsetasime läbi erinevad rakendused ja koostasime endale meeldivamatest võimalikest lahendustest analüüsi. Viiest analüüsitud rakendusest otsustasime, et peamine kujundusprogramm saab olema *Pixelorama*. Otsustasime *Pixelorama* kasuks muuhulgas seetõttu, et see töötab mitmel operatsioonisüsteemil, see on vabavaraline ning sellel on sisseehitatud animatsiooniliides.

Esialgsed graafilised tööd olid pigem katsetused, tutvudes pikselkunsti loomisega. Paralleelselt praktiliste katsetustega tutvusid disainerid pikselgraafika teoreetilise poolega. Peamiseks

allikaks oli Michael Azzi raamat *Pixel Logic: a Guide to Pixel Art*, mida soovitas kasutada juhendaja Laura Hein.

Esimestest praktilistest katsetustest jõudis mängu lõppversiooni üleskorjatav raamat. Seejärel algas tegelase loomine, keda oli kokku kolm erinevat varianti. Lõplikku varianti aitasid valida juhendajad praktikakohtumise käigus. Juhendajate poolt soovitud tegelase disaini arendasime edasi, kuna esialgne versioon tundus liiga kõrreline.

Seejärel alustas Taavi Ansper taseme taustaks oleva plaadikomplekti loomisega. Sellest arenes kuus erinevat versiooni, iga järgmine versioon edasiareng eelmisest. Vahepeal koostas Andero batuudi disaini, mille Taavi Ansper hiljem lisas eraldi esiplaani plaadikomplekti.

Järgmises etapis loodi kasutajaliidese elemendid ning viimane suurem etapp oli tegelase animatsioonikaadrite loomine. Ka animatsioonide puhul võtsime kuulda juhendajate tagasisidet, kes tajusid, et näiteks kõndimise esialgsele animatsioonile tuli lisada veel kaadreid, et tagada animatsiooni sujuvus.

4.4. Hinnang töö käigule

Pikselkunsti loomine oli huvitav, juba algusest peale oli tegu põneva stiiliga, milles mängugraafikat teha. Usume, et saime ülesandega päris hästi hakkama ning tulevikus on kindlasti ka ruumi olemasolevaid kujundusi lihvida ja kujundusi juurde lisada.

5. KASUTAJALIIDES

Kasutajaliidese kujundas Andero ja Taavi Rannu teostas kujunduse funktsionaalse lahenduse Unitys. Kuna Tudengiralli mängu süžee on seotud ülikooli läbimisega, põhineb kasutajaliides osaliselt Tallinna Ülikooli õppeinfosüsteemi (TLÜ ÕIS) kasutajaliidesel. See tähendab näiteks, et Tudengiralli peamenüü sarnaneb TLÜ ÕIS-i peamenüuga ülesehituse ja värvilahenduse poolest: pealkirjal on punane taust ja menüüvalikud on valgel taustal kastide sees.



Pilt side-by-side tudengiralli menüü ja ÕIS-i menüü

Tudengiralli peamenüüst on võimalik navigeerida mängu endani kui ka seadete valikusse. Seadetest saab muuta taustamuusika helitugevust ja heliefektide helitugevust. Helitugevuste reguleerimiseks kujundati lihtne horisontaalne *slider* koos ikoonidega, mis viitavad vaigistatud helile ja kõige suuremale võimalikule helitugevusele.

Liikudes kasutajaliideses mängimiseni, saab kasutaja valida, kas ta soovib mängida 1. taset või 2. taset. Mängides näeb kasutaja ekraani vasakul üleval nurgas taimerit. Mängu pausile pannes saab kasutaja muuta taustamuusika helitugevust ja heliefektide helitugevust. Pärast taseme läbimist avaneb kasutajale aken, kuhu ta saab sisestada oma nime, et ta nimi ja tasemeks kulunud aeg ilmuks kõige kiiremate aegade edetabelisse.

Kasutajaliides kujundati Affinity Designeris. Kuna see on olemuselt lihtne ja sarnaneb TLÜ ÕIS-ile, siis tiim võttis kasutajaliidese kujunduse hästi vastu. Juhendajatelt saadud tagasiside põhjal asendati esialgne sisuteksti kirjatüüp parema loetavuse eesmärgil.

6. HELI

Käesolev peatükk annab ülevaate heliefektide loomisest ning taustamuusika valimisest.

6.1. Helide loomine

Helide loomine kuulus Toivo Pärnpuu tööülesannete hulka. Samuti integreeris ta helid mängu koodi. Mängu jaoks sai esimeses versioonis planeeritud järgnevad heliefektid:

1. **Hüppamine.** Hüppamise heli salvestati kasutades telefoni ja hüpates pehme vaiba peal. Kuna mängu peategelane maandus hüppamise heli mängimise ajal, pidime heli oluliselt lühendama ja kiirendama.
2. **Maandumine** maandumise heli sai algselt planeeritud ning salvestatud koos hüppamisega, kuid kuna mängus selle heli maha mängimine tekitas liigse helide paljususe, siis jäi see heliefekt mängu seest välja.
3. **Jooksmine ja kõndimine.** Jooksmise ja kõndimise helide jaoks salvestas Toivo erinevaid versioone. Üks versioon salvestati loksutades veepudelit, teine aga liikudes mööda treppe. Ka need salvestatud efektid kasutust ei leidnud, kuna koos taustamuusikaga tekkis helide paljusus.
4. **Eseme korjamise heli.** Selle efekti loomiseks kasutas Toivo Audacity tarkvaras generaatorfunktsioone, et luua sobilik efekt, mis sarnaneb helipildilt Super Mario videomängu korjatavate esemete helidega.

6.2. Muusikaline taust

Mängu visuaalse stiili ja teema kajastamiseks otsustasime kasutada 8- või 16-bitist muusikalist tausta. Pärast põhjalikke otsinguid leidsime sobiva loo, mis aitab mängu atmosfääri luua ja täiendab mängu pikslikunsti elemente. Väljavalituks osutus lugu nimega Tension, mille autoriks on Onys. Väljavalitud lugu on litsentseeritud CC0 1.0 litsentsiga.

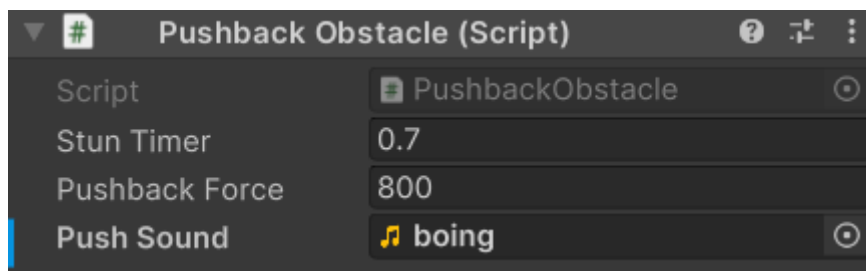
6.3. Helide maha mängimise lahendus

Heli maha mängimise lahenduse teostus jäi samuti Toivo teha. Helide lisamise esimene versioon algas AudioSource elemendi lisamisega vajalike mänguobjektide külge. Peagi selgus, et sellisel viisil ei ole võimalik kogu helitausta lahendada.

Pärast mõningast uurimist leidsime Unity õppematerjalidest helide mahamängimiseks lahenduse, mis põhineb Unity õppeprojektil "Architecture and Polish" (osa 2). Mängu on lisatud tühi GameObject, millele on lisatud kaks AudioSource komponenti, mis on ühendatud SoundManager.cs skriptiga.

Erinevate objektide puhul helide maha mängimiseks oli vaja lihtsalt defineerida muutuja skriptis (public AudioClip pushSound). Näiteks kui mängija puutus kokku takistusega,

mängis süsteem vastavat heli (`SoundManager.instance.PlaySingle(pushSound)`). Helifaili saab muuta Unity inspektori abil.



Joonis 9. Heli muutmine Unity inspektori abil

KOKKUVÕTE

Praktika tulemuse saab lugeda edukaks, kuna praktikaks seadud põhieesmärk, ehk kahe tasemega mängu loomine, on täidetud. Põhieesmärgini jõudmisel näitasid tiimiliikmed üles pidevat arengut, seda nii iseseisvalt oma vastutusalade lõikes mänguks vajalikke elemente luues kui ka laiemalt, võttes kuulda juhendajate tagasisidet, mille põhjal muutis tiim oma lähenemist. Juhendajate tagasiside põhjal tegi tiim praktika käigus muutusi nii mänguga seotud elementides kui ka rühmatöö korraldusliku poole pealt. Mängu tugevusteks on kindlasti kõrgel tasemel Unity mängumootori võimaluste kasutamine, kuna lõpptulemus on see, mida tiim algselt planeeris, ehk 2D-platvormimäng. Lisaks võib mängu tugevuseks lugeda tasemete jaoks loodud graafikat, mis oma modulaarsuses lubab luua hulgaliselt eriilmelisi tasemeid. Võimaliku puudusena võib välja tuua vähese testimise. Praeguse seisuga on mängu testinud tiimiliikmed ja osa juhendajatest. Selleks et veenduda, et mäng vastab võimalike kasutajate soovidele ning on täpselt õige raskusastmega, peaks laskma mängu testida kolmandatest osapooltest kasutajatel.