

TALLINNA ÜLIKOOL

Haapsalu Kolledž

Rakendusinformaatika õppekava

Ainar Kiison

PIKSLIKUNSTI LOOMINE 2D-VIDEOMÄNGUDELE

Diplomitöö

Juhendaja: MA Laura Hein

Haapsalu 2023

## SISUKORD

SISSEJUHATUS.....	4
1. TEOREETILINE ÜLEVAADE.....	6
1.1 Pikslükunst.....	6
1.2 2D-videomängu vaatenurgad .....	7
1.3 Sprait ja spraidileht.....	8
1.4 Pikslükunsti tööprotsess .....	9
1.4.1 Lõuendi suurus .....	9
1.4.2 Tegelase ja ekraani mõõtude vaheline suhe .....	10
1.4.3 Joon .....	11
1.4.4 Värvipalett.....	14
1.4.5 Virvtoonimine ja viirutamine .....	14
1.4.6 Sakitõrje .....	16
1.4.7 Graafika mängumootoris importimine.....	17
2. PRAKTILINE OSA .....	21
2.1 Graafika tarkvara valik.....	21
2.2 Pikslükunsti tööetappide järgnevus.....	22
2.3 Eeltöö .....	23
2.4 Graafika loomine.....	25
2.4.1 Spraitide joonistamine Piskel rakenduses .....	25
2.4.2 Animatsioon Piskel rakenduses.....	29
2.4.3 Tagatausta loomine Affinity Designer tarkvaras .....	31
2.5 Graafika ekspordimine ja importimine.....	32
KOKKUVÕTE.....	36
SUMMARY .....	37

ALLIKAD .....	38
LISA 1. DIPLOMITÖÖ KÄIGUS 2D-VIDEOMÄNGU JAOKS LOODUD GRAAFIKA ..	41

## SISSEJUHATUS

Tööjõuvajaduse seire- ja prognoosisüsteemi 2019. aasta uuringu kohaselt on mängude arendamine muutunud kiiresti kasvavaks ning oluliseks valdkonnaks. Mängutööstus on digitaliseerumisest ning virtuaalreaalsuse võimaluste kättesaadavamaks muutumisest saanud otsese tõuke ja sellele prognoositakse jätkuvat arengupotentsiaali. Valdkonnnaga koos kasvab vajadus tehnilise personali järele, nagu disainerid, arendajad, audiovisuaalse töö tegijad jne. (OSKA, 2019)

Valdav enamus tänapäevasest videomängude müügist ning kasutamisest toimub läbi digitaalsete müügiplatvormide nagu Steam, Origin, Amazon Prime või GamersGate (Financesonline, 2022). Viimasel kolmel aastal on Steami platvormil mängude müük kasvanud 74% ning kuigi 2022 aasta lõpp tõi kaasa väikese languse, ennustatakse käesolevaks aastaks uut tõusu (Obedkov, 2022).

Mängutööstuses on jätkuvalt populaarseks kategooriaks pikslükunstil põhinevad mängud, millel on stabiilne igapäevane kasutajaskond. Steami uute mängude edetabelis on aprill 2023 seisuga neli pikslükunstil põhinevat mängu kümne kõige kõrgemalt hinnatud mängu hulgas (Steam, 2023). Realistliku välimusega mängudega võrreldes on pikslükunsti visuaalide tootmine tõhusam, mis tähendab väiksemat aja- ja tootmiskulu (Rocket Brush Studio, 2021). Seetõttu on see hea valik algajale mänguarendajale, näiteks tudengile.

Õpingute käigus pikslükunstis mängu luues kogesin, et pikslükunsti kohta pole leida terviklikke materjale, mis keskenduks just mänguarenduse eripäradele. Multimeedia praktika raames kutsekooli õpilastega koostööd tehes põhjustas probleeme eestikeelse materjali puudumine, mis annaks selge ülevaate pikslükunsti loomise tööetappidest, tehnilistest võtetest, mängumootorite valikust ja graafika importimisest.

Eelnevast tulenevalt on antud diplomitöö eesmärk anda ülevaade pikslükunsti tööprotsessist lähtuvalt 2D-videomängutööstuse ning enamlevinud mängumootorite vajadustest. Diplomitöö raames leian vastused järgmistele küsimustele:

Millistest etappidest pikslükunstis graafika loomine 2D-videomängudele koosneb?

Millised tehnilised nõuded ja kvaliteedistandardid kehtivad 2D-videomängude jaoks loodavale pikslükunstile ning mil määral on need seotud mängumootori valikuga?

- Kuidas kõige optimaalsemalt jõuda igas tööetapis kvaliteetse tulemuseni?
- Eesmärgi saavutamiseks püstitasin järgmised ülesanded:
- Pikslükunsti loomist käsitleva kirjanduse läbitöötamine ja parima töövoole leidmine.
- Mängumootorite nõuetega tutvumine pikslükunsti loomise vaatepunktist.
- Pikslükunsti loomise tööetappide läbiproovimine, et testida teooriate toimimist praktikas.

Diplomitöö koosneb kahest osast. Esimeses osas keskendun teoreetilistele alustele – annan ülevaate pikslükunsti mõistest, disainimisest, animeerimisest ja tehnilistest nõuetest 2D-videomängu tööprotsessis.

Teises osas keskendun diplomitöö praktilisele osale. Tuginedes teoreetilisele osale loon 2D-videomängu pikselgraafikat ning impordin seda erinevatesse mängumootoritesse. Töö lõpus esitatakse kasutatud mängumootorite võrdlustabelid pikslükunsti mängu arendamise vajadustest lähtuvalt.

Töös on kasutatud 32 allikat ja sisaldab 30 joonist. Kui ei ole teisiti viidatud, olen töös esinevad joonised loonud käesoleva diplomitöö jaoks.

## 1. TEOREETILINE ÜLEVAADE

Antud peatükis annan ülevaate pikslükunsti mõistest, disainimisest, animeerimisest ja tehnilistest nõuetest 2D-videomängu tööprotsessis.

### 1.1 Pikslükunst

Pikslükunst on digitaalse kunsti vorm, mis on loodud tarkvara abil, kus pilte redigeeritakse pikslite tasemel. Pikslükunsti tunnuseks on ainulaadne visuaalne stiil, kus üksikud pikslid toimivad pilti moodustavate ehitusplokkidena. Pikslükunst on eksisteerinud alates esimestest pilditöötlustarkvara programmidest ja varajastest 2D-videomängudest 1970. aastatel, kuid mõiste võeti esmakordselt kasutusele Xerox PARCis 1982. aastal Robert Flegali ja Adele Goldbergi poolt. (Rouse, 2015.)

Pikslükunst on kasutusel olnud alates varajastest arkaadmängudest, nagu Space Invaders ja Pac-Man, kuni tänapäevaste *indie*-mängudeni, nagu Stardew Valley ja Terraria (Steam, 2023(a)). Pikslükunstil põhinevad mängud on jäänud mängutööstuse põhiosaks ning neil on stabiilne igapäevane kasutajaskond. Steami platvormi kõige rohkemate mängijatega pikslükunstil põhineval mängul Terraria on kõrghetkel 2020. aastal olnud ligi 500 000 mängijat (Steam, 2023 (b)).

Pikslükunsti üks peamisi eeliseid on võime edastada väikeses mahus palju teavet. Realistliku välimusega mängudega võrreldes on pikslükunsti visuaalide tootmine tõhusam, mis tähendab väiksemat aja- ja tootmiskulu (Rocket Brush Studio, 2021). Seetõttu on see hea valik *indie*-mängutootjatele ja alustavatele mänguarendajatele.

Piksligraafika esteetika pärineb 8-bitistest ja 16-bitistest arvutitest ning videomängukonsoolidest (Progetiiger, s.a.). Pikslükunsti mängudes on liikumine ja animatsioon sageli piiratud kaadrite arvuga, mis tekitab nostalgilist tunnet, rõhutades samal ajal ajastuse ja täpsuse tähtsust. Paljud kaasaegsed *indie*-mängud kasutavad pikslükunsti, et avaldada austust klassikalistele videomängudele. Lisaks videomängudes kasutamisele on pikslükunst muutunud omaette populaarseks kunstivormiks (Daniel, 2017).

## 1.2 2D-videomängu vaatenurgad

Mängu graafikaga alustades tuleks esimese asjana paika määrata vaatenurk, et seda elemente luues järgida. Pikslükunstis tehtud 2D-videomängudel on tavaliselt kasutusel vaid üks kindel vaatenurk, mis määrab, kuidas mängija mängumaailma ja selles olevaid tegelasi vaatab. Sellest tulenevalt vaatlen kolme kõige levinumat 2D-videomängu vaatenurka: küljelt kerimine, ülalt alla ning isomeetriline. (Jan, 2017)

Külgkerivas (ingl *side scrolling*) videomängus on mängija vaade piiratud ekraanil nähtavaga ja mängumaailm kerib horisontaalselt, kui mängutegelane liigub vasakult paremale (või paremalt vasakule). Külgkerimise perspektiivi seostatakse kõige sagedamini platvormimängudega, nagu näiteks Super Mario Bros (1983, Nintendo). Külgkerimise perspektiiv sobib eriti hästi 2D-videomängude jaoks, kuna võimaldab selgelt ja arusaadavalt kujutada mängumaailma ja mängija positsiooni selles. Mängijad saavad kiiresti aru mängu mehaanikast ja eesmärkidest ning keritav ekraan loob edenemise ja hoo tunde, mis hoiab mängijad kaasatud ja motiveeritud. (Jan, 2017)

Ülalt-all (ingl *top-down*) vaatenurga puhul on mängija vaade seatud otse mängumaailma kohale, võimaldades näha kogu mänguvälja korraga. Üks ülalt-all vaatenurga eelistest on see, et see pakub mängijatele kontrolli ja ülevaate tunnet, mis pole teiste vaatenurkade puhul võimalik. Mängijad saavad näha kogu mängumaailma ja teha otsuseid paljude tegurite põhjal, mitte lihtsalt reageerida sellele, mis nende ees on. See perspektiiv võimaldab ka keerukamat mängumehaanikat, kuna mängijad saavad hallata mitut ressursi ja teha strateegilisi otsuseid, mis mõjutavad mängu kulgu. (Jan, 2017)

Isomeetriline (ingl *isometric*) vaatenurk loob 2D-mängus illusiooni 3D-keskkonnast, kasutades objektidele sügavuse ja mõõtmete andmiseks nurga all olevaid jooni ja kujundeid. Isomeetrilises mängus on mängija vaade seatud tavaliselt 45-kraadise nurga alla. See vaatenurk annab mängijatele parema ruumitaju, kuna nad näevad korraga rohkem mängumaailma ning oskavad paremini oma liigutusi ja tegevusi planeerida. (Jan, 2017)

### 1.3 Sprait ja spraidileht

Sprait on tavaliselt läbipaistva taustaga pilt, mis asub või liigub taustapildi ees (The Computer Graphics and Virtual Reality Study Lab @ University of Tartu, s.a.). Pikslükunsti kontekstis on sprait kahemõõtmeline graafika, mis kujutab tegelast, objekti või muud elementi videomängus või interaktiivses meedias (Wain, 2023).

Algselt kasutati spraitte varajastes videomängudes riistvaraliste piirangute tõttu, kuna need võimaldasid arendajatel luua piiratud arvu pikslitega üksikasjalikku graafikat. Tänapäeval on spraidid mängude arendamisel populaarne valik tänu nende mitmekülsusele ning nende loomise ja animeerimise lihtsusele. (Azzi, 2011) Spraitte võib olenevalt mängu vajadustest luua erineva suuruse ja eraldusvõimega ning neid saab animeerida, kasutades liikumise tekitamiseks kaadrite seeriat (Lambert, 2013).

Pole haruldane, et mängudel on kümneid kuni sadu spraitte. Kõigi nende üksikute kujutiste mugavamaks haldamiseks ja ekraanile kuvamise protsessi kiirendamiseks kasutavad paljud mängud spraidilehti (ingl *sprite sheet*) (Lambert, 2013).

Spraidileht koondab ühe objekti spraidid üheks pildifailiks, mida saab kasutada objekti animeerimiseks (The Computer Graphics and Virtual Reality Study Lab @ University of Tartu, s.a.). Palju kiirem on tuua mängu üks pilt ja kuvada ainult osa sellest, kui palju pilte tuua ja neid kuvada. Spraidilehel olevaid poose kiiresti üksteise järel kuvades luuakse illusioon sellest, et tegelane või objekt liigub (Lambert, 2013). Spraidilehed koosnevad kahest osast: raamidest (ingl *frame*) ja tsüklitest (ingl *cycle*). Raam on üks pilt ehk sprait spraidilehest. Kui raamid on järjestatud pidevaks liikumiseks, loob see tsükli. (Lambert, 2013) Joonisel 1 on näha spraidilehte, mis koosneb kuueteistkümnest raamist ja neljast tsüklist.





Joonis 1. Spraidileht (Bowen, s.a.)

## 1.4 Pikslikunsti tööprotsess

Videomängu graafika kvaliteedi tagamiseks tuleks pikslikunsti loomisel järgida kindlaid tööpõhimõtteid ja -etappe. Käesolevas peatükis tutvustan pikslikunsti olulisemaid mõisteid ja tehnikaid vastavalt nende loogilisele järgnevusele tööprotsessis. Tööprotsessi etappide järgnevused olen välja töödanud autorite Max Hervieux'i (2015), Daniel Silberi (2016), Michael Azzi (2019) ja Mark J.P. Wolfi (2008) teostele tuginedes.

### 1.4.1 Lõuendi suurus

Pikslikunsti loomisel tuleks alustada lõuendi suuruse valikust. Lõuendi suurus viitab joonistusala kõrguse ja laiuse pikslite arvule. Lõuendi suurus on mängu üldise esteetika jaoks ülioluline. Väiksemad lõuendid annavad retrolikuma ilme, suuremad lõuendid mõjuvad moodsamalt. Väiksemate lõuendi suurusetega on pikslite vähesuse tõttu keerulisem töötada, kuna kujutiste loomisel tuleb rohkem üldistada ning kasutada vähem detaile. (Azzi, 2019)

Mängu jaoks pikslikunsti tehes tuleb arvestada, et lõuendi suurus sobituks mängu kogueraldusvõime ja kuvasuhtega. See tagab, et graafika näeb välja terav ja sobib ideaalselt mängu piiridesse. (Wain, 2023)

Kogueraldusvõime on ekraani pikslimassiivi laiuse ja kõrguse korrutis – näiteks FHD (ingl *full high-definition*) eraldusvõime on 1920 horisontaalset pikslit ja 1080 vertikaalset pikslit. Kuvasuhe (ingl *aspect ratio*) on ekraani laiuse ja kõrguse vaheline suhe – näiteks kuvasuhe 16:9 tähendab, et ekraani laius on 16 ühikut ja kõrgus 9 ühikut. (Holmes, 2023)

### 1.4.2 Tegelase ja ekraani mõõtude vaheline suhe

Peale kogueraldusvõime valimist tuleb otsustada, milline on mängu ekraani ja tegelase spraitide vaheline suhe. Näiteks suhe 1:24 tähendab, et ekraan mahutab 24 tegelaskuju spraiti. Suhe 1:24 muudab teie tegelase ekraanil suureks, samas kui 1:300 on veel nähtav, kuid üsna raskesti eristatav. (Azzi, 2019; Wain, 2023) Joonisel 2 on mängitava tegelase suhe ekraaniga 1:38 ehk moodustab umbes 2,5% ekraanist. See võimaldab mängijal tegelast ümbritsevat keskkonda rohkem näha ning tegelaskujuga paremini manööverdada. (Azzi, 2019)



Joonis 2 Näide tegelaskuju ja ekraani vahelisest suhtest (Azzi, 2019)

Lõuendi suurus vastab spraidi suurusele – seega on oluline arvestada ka soovitud detailsuse tasemega. Kui tahta, et pikslkunst oleks väga üksikasjalik, on töötamiseks vaja suuremat lõuendi suurust. Levinud tegelaskujude või objektide lõuendite suurused on 8 x 8, 16 x 16 või 32 x 32 pikslit, vt joonis 3 (Wain, 2023). Samas kasvatab suurem lõuend töökoormust. Hea viis enda jaoks sobiva lõuendi suuruse leidmiseks on esmalt joonistada vähim detail, mida soovite kujutada (nt tegelaskuju silmad) ning lähtuda ülejäänu osas sellest (Pixel Art Wiki, s.a.).



Joonis 3. Erinevate lõuendi suuruste mõju spraidi detailsusele (Cheishiru, 2021)

### 1.4.3 Joon

Spraidi joonistamisel tuleks alustada piirjoonest (joonis 4) (Wain, 2023). Piirjooned on pikslkunst põhikomponendid, kuna need määravad joonistatava objekti või tegelase kuju. Piirjooned aitavad luua selge ja äratuntava kujundi, millest on lihtne aru saada ning aitavad kontrasti abil spraidil mängukeskkonnast eristuda. (Azzi, 2019)



Joonis 4. Piirjoonega spraidi joonistamise alustamine (Wain, 2023)

Jooni kasutatakse pikslkunstis ka liikumise edastamiseks ning kujutisele tekstuuri ja sügavuse lisamiseks (Hervieux, 2015).

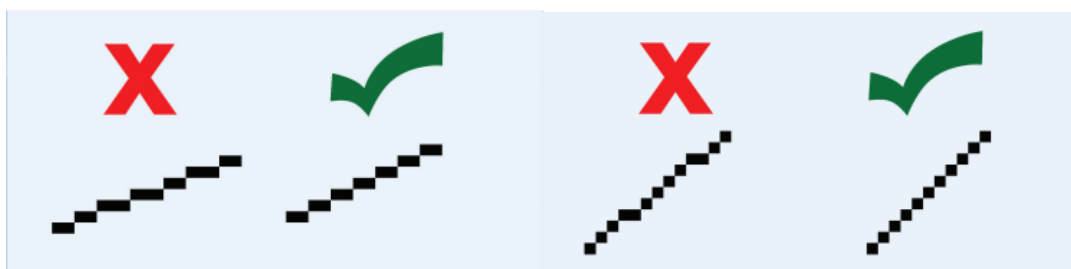
Pikslikunstis on joonte loomisel üks peamisi väljakutseid ühtlase laiuse ja sujuvuse saavutamine – isegi väikseim kõrvalekalle kavandatud joonest võib olla märgatav. Erinevat tüüpi joonte loomiseks on pikslikunstis mitu tehnikat. (Hervieux, 2015)

Põhilise sirgjoone saab luua, asetades pikslid lihtsalt sirgjoonele. Paksemate joonte loomiseks saab paigutada mitu pikslit kõrvuti ja õhemate joonte loomiseks võib üksikud pikslid vahele jätta. Üldiselt on soovitatav määrata joonte paksuseks üks piksel, nagu on näha joonisel 5. (Silber, 2016; Hervieux, 2015)



Joonis 5. Ühe piksli laiune joon paremal (Silber, 2016)

Diagonaalsete joonte loomine on keerulisem – sujuvuse saavutamiseks tuleb kasutada võrdseid segmendi pikkusi ehk ühtlast sammumustrit (joonis 6). Kui diagonaal on nurga all, mida ei ole võimalik võrdsete segmendi pikkustega saavutada, tuleks kasutada vaheldumisi kahte erinevat segmendi pikkust (joonis 7). (Silber, 2016; Hervieux, 2015)

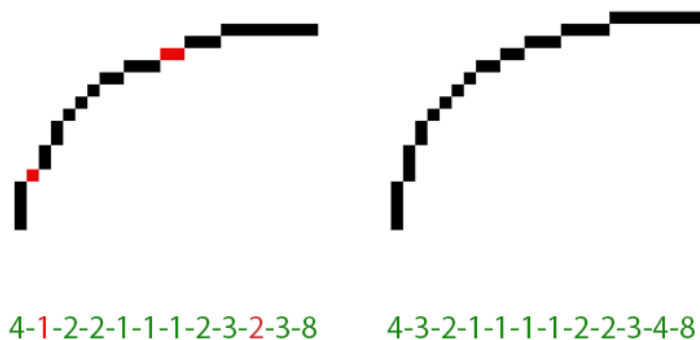


Joonis 6. Ühtlane sammumuster annab diagonaaljoonele parema kvaliteedi (Silber, 2016)



Joonis 7. Vahelduvad segmendi pikkused diagonaali joonistamisel (Silber, 2016)

Kõige keerulisem on sujuvate kõverjoonte loomine, mille juures tuleks järgida põhimõtet, et iga kõverat moodustav segment järjest astmeliselt muutuks (Azzi, 2019). Vastavat põhimõtet väljendab parempoolne näide joonisel 8.



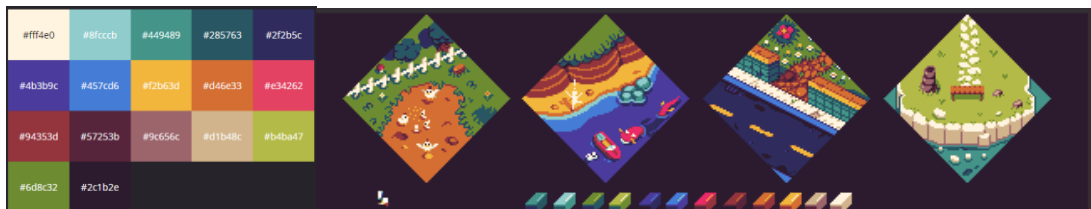
Joonis 8. Kõverjoone segmentide astmeline muutumine tagab joone sujuvuse (Wain, 2023)

#### 1.4.4 Värvipalett

Värvipalett on pikslükunstis oluline aspekt, kuna see määrab värvide vahemiku, mida konkreetsetes kunstiteoses kasutada saab. Pikslükunsti iseloomustab sageli piiratud värvipalett. See tulenes algselt kasutatavate arvutite ja mängukonsoolide piirangutest. (Hervieux, 2015)

Teine põhjus piiratud värvipaleti kasutamiseks on hoida ühtset välimust erinevatel ekraanidel ja seadmetel. Kasutatavate värvide arvu piirates näeb loodud teos suurema tõenäosusega erinevatel ekraanidel ühesugune välja, mis on eriti oluline vanemate ja madalama eraldusvõimega seadmete puhul. (Wolf, 2008)

Retro stiilis pikslükunsti loomisel tuleb värvipaletti piirata, kuid lõppkokkuvõttes on kasutatavate värvide hulk tänapäeval pikslükunsti looja enda otsustada (Wain, 2023). Joonisel 9 on näha värvipaletti ja selle põhjal loodud graafikat.



Joonis 9. (Lospec, 2019)

#### 1.4.5 Virvtoonimine ja viirutamine

Virvtoonimine (ingl *dithering*) on tehnika, mida kasutatakse pikslükunstis, et luua illusioon täiendavatest värvidest ja varjunditest lisaks kunstiteose piiratud värvipaletis olevatele. Virvtoonimise abil saab luua mitmesuguseid efekte, alates kergest varjutamisest kuni keerukate tekstuuride tekitamiseni. Virvtoonimist kasutatakse tavaliselt videomängude puhul, kus piiratud värvipalett ja graafika madal eraldusvõime nõuavad loomingulisi lahendusi sügavuse ja detaili edasiandmiseks. (Silber, 2016)

See toimib erinevat värvi pikslite vaheldumisel kindlas mustris, et luua uute värvide või varjundite välimus. Virvtoonimist kasutades saab luua mulje vahevärvidest või toonidest, mida värvipaletis tegelikult ei eksisteeri. (Hervieux, 2015)

Joonisel 10 on näha, et kui piksliekunstnik soovib luua värvivarjundit, mida paletis pole, saab ta erinevat värvi piksleid malelauda meenutavasse mustrisse asetades luua illusiooni uuest värvitoonist. Erinevat värvi pikslite kindlas mustris vaheldumist tajub silm kahe värvi segunemisel tekkiva uue tonaalsusena. (Hervieux, 2015)



Joonis 10. Virvtoonimise näide (Hervieux, 2015)

Viirutamine (ingl *hatching*) on tehnika, mida tavaliselt kasutatakse traditsioonilises kunstis tekstuuri ja varju loomisel, tõmmates rida paralleelseid jooni. Piksliekunstis saab sarnaselt viirutamist kasutada, paigutades piksliread kindlasse mustrisse, et luua tekstuuri või varjutuse muljet nagu on näha joonisel 11. Piksliekunstis kasutatakse viirutamist tavaliselt tekstuuride, näiteks puidu või roomaja soomuste loomisel. Pikslite klastreid konkreetse mustriga hoolikalt korraldades, saab anda tekstuurile realistliku ja üksikasjaliku välimuse. (Hervieux, 2015)

Viirutamist saab kasutada ka sügavuse või varjutuse tekitamiseks, eriti piirkondades, kus värvid on sarnased või kus soovitakse sujuvat värviüleminekut. Eristatakse ka ristviirutamist, mis erinevalt tavalisest viirutamisest koosneb joonte joonistamist ühes suunas ja seejärel teise paralleelsete joonte komplekti joonistamist risti esimese komplekti peale, luues ruudustikutaolise mustri. Siiski võib piksliekunstis viirutamine olla raskendatud, kuna saadaolevate pikslite piiratud arv võib muuta mustrite või tekstuuride loomise silmale koormavaks. (Hervieux, 2015)



Joonis 11. Erinevad viirutustehnikad (Hervieux, 2015)

#### 1.4.6 Sakitõrje

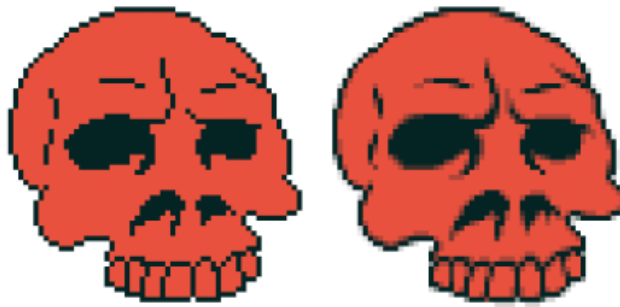
Sakitõrje (ingl *anti-aliasing*) on meetod sakilisuse kõrvaldamiseks pildi joontele ja servadele sileda ilme andmisega kuvapinnal (Standardipõhine tarkvaratehnika sõnastik, 2023). Graafiliste objektide, sh tähemärkide piirjooned paistavad sakilistena, kui väljundseade (kuvar või printer) ei ole sileda joone kuvamiseks või printimiseks piisavalt suure eraldusvõimega. Sakitõrje vähendab sakkide silmatorkavust nõnda, et ümbritseb need hallide (ingl *grayscale* e halltoonides kujutise puhul) või värviliste (värvikujutiste puhul) vahetoonidega. See vähendab küll piirjoonte silmaga tajutavat sakilisust, kuid ühtlasi muutuvad need "ähmasemaks". (Vallaste, 2023)

Pikslükunstis tähendab sakitõrje joone servale pikslite lisamist, et vähendada joone sakilisust. Paljudes kujundustarkvara programmides ja mängudes teeb selle protsessi ära automaatne filter. Sakitõrje kasutamisel luuakse värv, mis on segu joone värvist ja taustavärvist, ning kasutatakse seda värvi joone mõne või kõigi sammude pehmendamiseks. (Azzi, 2019)

Liiga palju sakitõrjet võib rikkuda pikselkunsti esteetikat ja muuta pilt uduseks või moonutatatuks. Tehnika, mis selle vastu võib aidata, on aeg-ajalt pilt välja suumida, et näha, kuidas pilt kaugelt paistab. (Azzi, 2019)

Joonis 12 näitab, kuidas joonele lisatud sakitõrje muudab joone palju sujuvamaks. Vasakpoolne pilt on kahe värviga joonistatud kolju algkujutis. Pärast väheste üleminekuvärvide lisamist musta ja teiste värvide vahel on joon palju loomulikum, aga ka udusem. (Silber, 2016)





Joonis 12. Sakitõrje on lisatud parempoolsele kujutisele (Silber, 2016)

#### 1.4.7 Graafika mängumootorisse importimine

Pikslükunstis mängu loomiseks toob GameMakeri artikkel välja järgmised parimad mängumootorid: Godot, Unity, Unreal Engine ja GameMaker Studio (Bramble, 2022).

GameMakeri mängumootor on tänapäeval üks pikima tööajaga *indie*-mängumootoreid, mis algselt ilmus aastal 1999. Tänu oma pikaajalisusele on GameMakeri kohta leida arvukalt juhendeid ja õpetusi. GameMaker võimaldab visuaalset koodi kirjutamist ning mängumootori enda programmeerimiskeele GameMaker Language kasutamist. GameMaker on pigem mõeldud 2D-mängude jaoks, aga võimaldab ka 3D-mängude tegemist. GameMaker võimaldab eksportida mängu mitmele erinevale platvormile, näiteks Nintendo Switch, Xbox Series X, Xbox Series S, PlayStation 4 ja PlayStation 5. (Bramble, 2022; GameMakerBlog, 2009)

Unity alustas algselt 3D-mängumootorina 2005. aastal, enne kui 2013. aastal lisati ametlikult 2D-tugi. Kuigi 3D-mängud aitasid muuta Unity populaarseks, on see ka 2D-mängude jaoks ideaalne valik. Unity on üks enim toetatud *indie*-mängumootoreid maailmas, millel on tuhandeid kirjalikke ja visuaalseid juhendeid. Unity peamiseks programmeerimiskeeleks on C#. Üks Unity suurimaid väärtusi on suur hulk toetatud platvorme: näiteks Windows, MacOS, Linux, Android, iOS, kõik suuremad kodukonsoolid ning VR-platvormid Oculus ja Steam VR. Unity pakub tasuta mängude loomise tarkvarapaketti, mis on kasutajale tasuta seni, kuni mängu tulu on alla 100 000 dollari aastas. (Bramble, 2022; Peckham, 2019)

Unreal Engine on üks parimaid mänguarenduse tarkvara lahendusi maailmas. See mängumootor ilmus 1998. aastal. Unreal Engine on tööriist mängutööstuse professionaalidele.

Unreal Engine'i peamine programmeerimiskeel on C++. Ehkki mõeldud pigem 3D-mängude jaoks, võimaldab Unreal Engine ka 2D-mänge. Unreal Engine on tasuta ning pakub tuhandeid õpetusi ja juhendeid. Mängumootori kasutamine on tasuta, kuni mäng teenib alla 1 miljoni dollari aastas. (Bramble, 2022; Thomsen 2010)

Godot mängumootor on avatud lähtekoodiga tasuta mängumootor. Kuigi Godot toetab ka 3D-mängude loomist, toimib see paremini 2D-mängude arendamisel, osaliselt tänu pikslipõhisele mõõtmisüsteemile. Godot ilmus 2014. aastal. Godot peamiseks arenduskeeleks on mängumootorisine GDScript. Kuigi Godot ei toeta ametlikult konsooli eksporti, on see võimalik kolmandate osapoolte abil. Godot kohta on leida vähem juhendeid ja materjale kui eelneva kolme mängumootori kohta. (Bramble, 2022; Linietsky 2019)

Mängumootorite olulisemad näitajad olen toonud kategooriate kaupa välja võrdlevas tabelis 1.

Tabel 1. Mängumootorite võrdlus (Godot, s.a.; Unity, 2023; Unreal Engine, 2023; GameMaker, 2023)

	Godot	Unity	Unreal Engine	GameMaker
<b>Viimane versioon 19.05.2023 seisuga</b>	4.0.3	2022.2.9	5.2	2023.4.0.84
<b>Hind</b>	Tasuta	Tasuta kuni mäng teenib alla 100000\$ aastas.	Tasuta kuni mäng teenib 1 miljon USD aastas siis 5% honorari	Tasuta
<b>Programmeerimis- keel</b>	C#  C++	C#  UnityScript  Boo	C++	Gamemaker Language

	GDScript			
<b>Toetatud operatsiooni-süsteemid</b>	Linux MacOS Windows	MacOS Windows	MacOS Windows	MacOS Windows
<b>Toetatud helivormingud</b>	WAV Ogg Vorbis	AIFF WAV MP3 Ogg Vorbis	WAV	WAV MP3 Ogg Vorbis
<b>Toetatud pildivormingud</b>	BMP DDS EXR HDR JPG PNG TGA SVG WEBP	BMP TIF TGA JPG PSD PNG	BMP PCX PNG PSD TGA JPG EXR	PNG SWF

<b>Võimaldab spraidilehe importi</b>	Jah	Jah	Jah	Jah
<b>Võimaldab <i>parallax</i> tagatausta</b>	Jah	Jah	Jah	Jah

Mängumootoreid võrreldes leidsin, et ainus pildivorming, mida kõik ühiselt toetavad on PNG (ingl *Portable Network Graphics*). PNG-pildivorming on rastergraafika failitüüp, mis kasutab kadudeta tihendamist, mis tähendab, et pildi tihendamise või lahtipakkimise ajal ei lähe andmed kaotsi.

Eristatakse PNG-8, PNG-24 ning PNG-32 vormingut. PNG-8 toetab 8-bitist värvisüsteemi, mis teeb toetatud värvide arvuks 256, ja läbipaistvust, võimaldades pikslitel olla kas täiesti läbipaistvad või täiesti läbipaistmatud. PNG-24 toetab 24-bitist värvisüsteemi, mis teeb toetatud värvide arvuks 16 miljonit ja PNG-32 lisab 24-bitisele värvisüsteemile osalise läbipaistvuse ehk alfa-kanali. PNG-8 on seetõttu vähem sobilik piltidele, mis nõuavad sujuvaid värvi üleminekuid või poolläbipaistvaid alasid. Piiratud värvipaletist tulenevalt on PNG-8 failisuurused üldjuhul väiksemad ning sobivad hästi olukordades, kus faili suuruse minimeerimine on prioriteet. (MiniTool, 2023)

Seega, kui on tegemist retro stiilis piiratud värvipaletti kasutava graafikaga, võiks spraidilehed eksportida PNG-8 pildivormingus, mis aitab ka mängumootorisse importimisel andmemahte vähendada. Muul juhul tuleks eelistada PNG-24 või PNG-32 pildivormingut, et säilitada graafika kvaliteeti.

## **2. PRAKTILINE OSA**

Käesolevas peatükis rakendan teoreetilises osas tutvustatud tööpõhimõtted praktikasse, luues pikslükunsti 2D-videomängu jaoks. Mängus kontrollib mängija peategelast, kes on Tallinna Ülikool tudeng ja üritab ajavaenlasi trotsides kooli lõpetada. Tegelane liigub mängumaailmas erinevatel platvormidel joostes ja hüpates ning ajavaenlasi kohvikruusiga visates. Vaenlast puudutades või platvormilt alla kukkudes on mäng läbi. Mängu graafika loomiseks kasutasin Piskel veebirakendust ja Affinity Designer tarkvara.

### **2.1 Graafika tarkvara valik**

Graafika loomiseks oli tähtis leida parimad pikslükunsti joonistamist võimaldavad programmid. Williamsi (2023) artikli järgi on pikslükunsti koostamiseks sobivaimad järgmised tarkvarad: Photoshop, GIMP, Krita, Pixlr, Pyxel Edit, GraphicsGale, Aseprite, Pro Motion NG, Pixie Engine Editor, Piskel, Pixilart. Nimetatud programmid võiks jagada üldistavalt neljaks. Vastavalt otstarbele saab jaotada neid laiaotstarbelisteks ja pikslükunsti spetsiifilisteks ning maksumuse alusel on võimalik eristada tasulisi ja tasuta programme.

Laiatstarbeliste graafiliste programmidega (näiteks Adobe Photoshop) on võimalik pikslükunsti loomiseks mitmeid tööriistu kasutada, aga need on üldjuhul kallid ega pruugi võimaldada pikselkunsti animeerimiseks vajalikku spraidilehe eksportimist. Laiatstarbeliste programmide üheks eeliseks on see, et need toetavad mustrite loomist, mis võimaldab lõputult jätkuvate graafiliste elementide (nt tausta) omavahelist liituvust jälgida.

Pikslükunsti spetsiifilised programmid (näiteks Aseprite) ei ole samuti üldjuhul tasuta, kuid on odavamad kui laiaotstarbelised programmid ning omavad mitmeid pikslükunsti eripärale suunatud tööriistu. Üldjuhul on neil väga head animeerimise võimalused ja toetatud nii kaaderhaaval- kui ka ajateljepõhine animeerimine.

Tasuta programmid on soositumad algajatele. Need omavad elementaarseid pikslükunsti loomiseks vajalikke tööriistu ning neid on lihtne õppida. Parimad tasuta programmid võimaldavad ka animeerimiseks vajalikku spraidilehe eksporti.

Lisaks leidub nii laiaotstarbelisi (näiteks GIMP) kui ka pikslükunsti spetsiifilisi (näiteks Piskel) tasuta tarkvara programme. Laiatstarbeliste tasuta programmide miinuseks võib pidada

ülemäärast keerukust ning pikslükunsti spetsiifiliste tööriistade puudumist. Ehkki tasuta pikslükunsti loomiseks mõeldud programmid on vähemate töövõimalustega, on mängugraafika loomisel nende eeliseks lihtsus ning elementaarsete pikslükunsti tööriistade olemasolu.

Diplomitöö graafika loomiseks valisin tasuta pikslükunsti programmi Piskel, mis katab mu graafilised vajadused mängu tegelaste ja platvormi loomiseks. Tagatausta loomiseks valisin laiaotstarbelise tarkvara Affinity Designer, kuna mul on olemas selle litsents ning varasem kasutuskogemus. Piskel rakendus ei toeta jätkuvaid mustreid, kuid Affinity Designeri Symbol funktsioon võimaldab luua ühest elemendist katkematut mustrit, mis muudab tausta graafika loomise mugavamaks.

## **2.2 Pikslükunsti töötappide järgnevus**

Max Hervieux'i (2015), Daniel Silberi (2016), Michael Azzi (2019) ja Mark J.P. Wolfi (2008) teostele tuginedes võiks pikslükunsti loomise tööprotsess koosneda järgnevatest sammudest:

1. mängu vaatenurga valik;
2. lõuendi suuruse ja eraldusvõime valik;
3. spraitide ja ekraani mõõtude vaheline suhe;
4. piirjoonte loomine;
5. värvipaleti loomine ja värvikasutus;
6. virvtoonimise ja viirutamise kasutamine;
7. sakitõrje kasutamine;
8. spraitide animeerimine ja spraidilehtede koostamine;
9. graafika eksportimine;
10. mängumootoris importimine.

Lähtuvalt töötappide iseloomust võib grupeerida esimesed kolm (vaatenurga valik; lõuendi suurus ja eraldusvõime; ekraani ja spraitide mõõtude vaheline suhe) eeltöö ühisnimetaja alla. Järgmised viis sammu (piirjoonte loomine; värvipaleti loomine ja värvikasutus; virvtoonimise ja viirutamise kasutamine; sakitõrje kasutamine; spraitide animeerimine ja spraidilehtede koostamine) võiks liigitada graafika loomiseks. Viimase grupi moodustavad graafika eksportimine ja importimine.

Järgmistes peatükkides kirjeldan oma mängu graafika loomist eelnevalt loetletud sammude kaupa.

### 2.3 Eeltöö

Siin peatükis tutvustan eeltöökä liigituvaid toiminguid ja otsuseid oma mängu pikslükunsti loomisel.

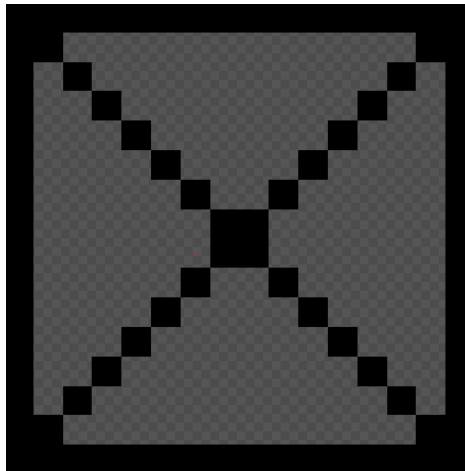
Pikslükunsti loomisel alustan vaatenurga e perspektiivi valikust, kuna see määrab ära pikslükunstis loodavate mänguelementide laadi ja kuju. Kuna külgerimise vaatepunkt on pikslükunstis mängude puhul kõige tüüpilisem ning sobib hästi platvormeri žanris mängudele, teen ka oma mängu vastavalt.

Tulenevalt minu mängu žanrist ja sisust on vaja luua järgnevad elemendid:

- tegelane, kelle liikumist mängija kontrollib;
- vaenlane, keda mängija üritab vältida;
- kohvitass, mida mängija viskab;
- platvorm, mille peal tegelane liigub;
- tagataust, mis loob mängu atmosfääri.

Olen otsustanud mängu teha arvutis mängimiseks ning tegelased animeerida spraidilehe abil. Kuna Steami (Steam(c), 2023) statistika järgi kasutab 64.6% kasutajatest 1920 x 1080 eraldusvõimega ja 16:9 kuvasuhtega kuvarit, siis valin selle oma mängu lõpperaldusvõimeks. Kuna lõuendi suurus peab sobituma mängu kogueraldusvõime ja kuvasuhtega, siis loon selle ka samades mõõtmetes. See tagab, et graafika näeb välja terav ja sobib ideaalselt mängu piiridesse.

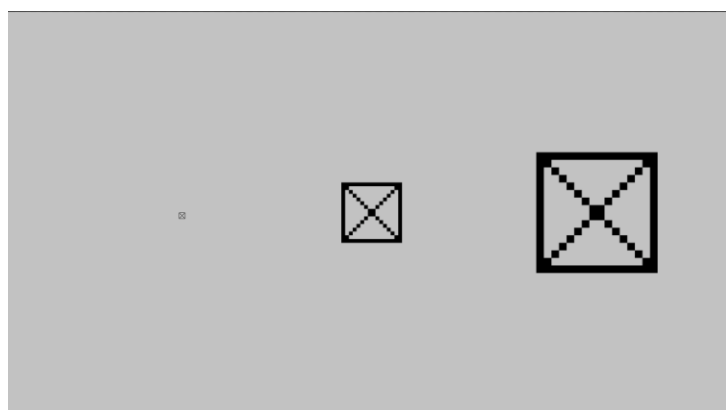
Spraitide lõuendi suuruseks valisin klassikalistele pikslükunsti mängudele omase 16 x 16 pikslit. Selles suuruses tegin nii tegelase, vaenlase kui ka platvormi. Platvorm ei pea iseenesest olema sama lõuendi suurusega kui tegelane, kuid see hoiab mängu visuaalselt ühtlasena. Joonisel 13 on näha 16 x 16 piksli suurust lõuendit, kus musta värviga on markeeritud 1 piksli suurused alad.



Joonis 13. Lõuend suurusega 16 x 16 pikslit

Mängu ekraani ja spraidi mõõtude suhtest sõltub see, kui suurelt me spraiti näeme. Kui 16 x 16 piksline sprait oleks mängu tegelane, ei paistaks see 1920 x 1080 pikslise eraldusvõimega ekraanilt silma. Joonisel 14 tõin välja 16 x 16 piksli suuruse spraidi, sama spraidi kümnekordse suurenduse ehk 160 x 160 piksli suuruse spraidi ja kahekümnekordse suurenduse ehk 320 x 320 piksli suuruse spraidi asetatuna 1920 x 1080 piksli suurusele ekraanile.

Kuigi suurmas mõõdus tegelase sprait oleks mängus hästi nähtav, on kümnekordse suurendusega sprait paremini manööverdatav, millest tulenevalt otsustan mängu spraitide koostamiseks valida 160 x 160 pikslise eraldusvõime. 160 x 160 piksli suurune sprait moodustab ligikaudu 1,23% Full HD-ekraanist. Suurenduse kasutamine imiteerib vanade, madala eraldusvõimega mängude graafilist stiili, mis loob retroliku tunde.



Joonis 14. Erinevate suurustega spraidid Full HD ekraanil

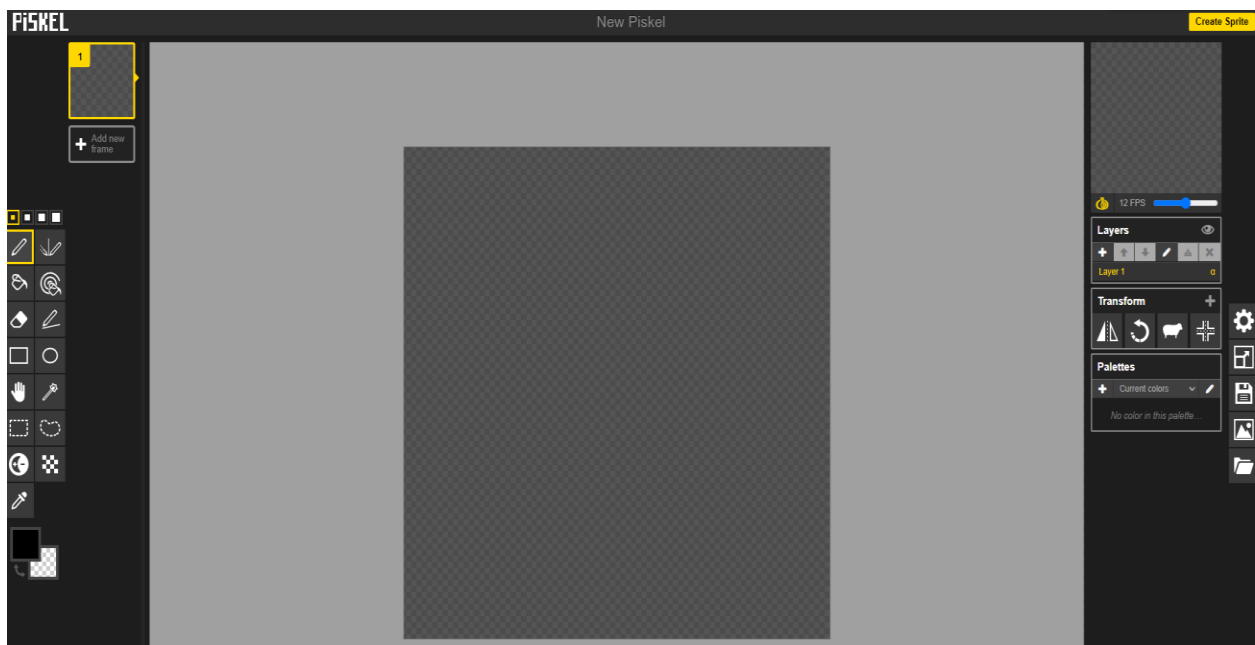


## 2.4 Graafika loomine

Järgmisena kirjeldan tööprotsessi valitud tarkvara programmides.

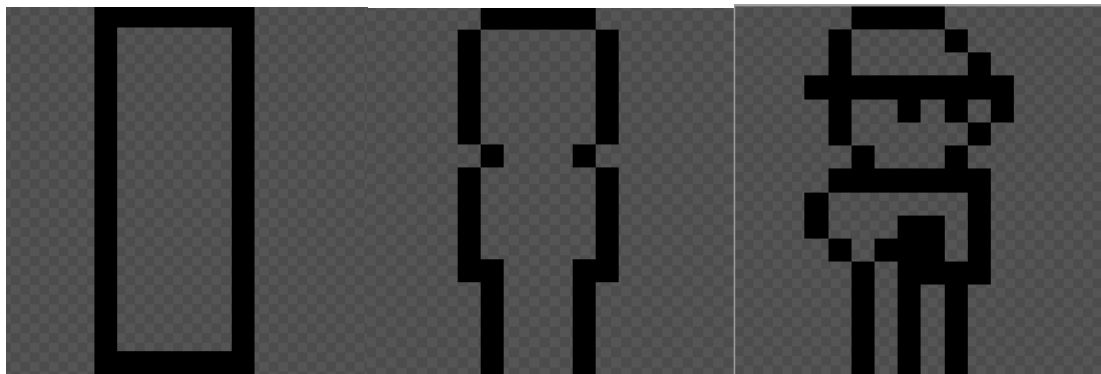
### 2.4.1 Spraitide joonistamine Piskel rakenduses

Kui eeltöö on valmis, saab alustada spraitide joonistamist, milleks kasutan vabavaralist veebipõhist pikslükunstile orienteeritud rakendust Piskel (joonis 15). Alustan tegelase spraidi joonistamisega, seejärel liigun edasi vaenlase, kohvitassi ja platvormi juurde.



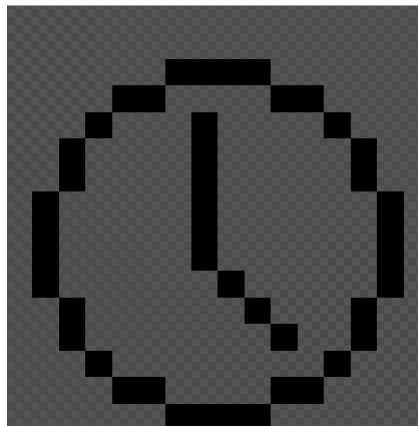
Joonis 15. Piskel rakenduse töökeskkond

Joonte kujundamiseks kasutan Pen tööriista, mis märgib iga piksli valitud värviga. Kuna spraidi kujundamisel tuleks alustada piirjoonest, panen esmalt musta värviga paika tegelase kõrguse ja laiuse. Seejärel kasutan vaheldumisi Pen ja Eraser tööriista, et määrata ära pea ja muude jäsemete paiknemise. Lõpetuseks lisan detailid – panen paika silmade asetuse, mütsi suuruse ning tegelase kehahoiaku. Joonisel 16 on näha tegelase piirjoone arengut.



Joonis 16. Tegelase piirjoone loomise etapid

Tulenevalt mängu temaatikast on vaenlaseks kell. Ümmarguse piirjoone kujundamisel järgisin teoorias väljatoodud põhimõtet, et iga kõverat moodustav segment järjest astmeliselt muutuks. Erinevalt eelnevalt joonistatud tegelasest ei täida kell lõuendit täienisti ära (Joonis 17). Kuna kavatsen vaenlase animeerimise protsessi käigus hüppama panna, siis on selle jaoks vaja lõuendi sees ruumi. Samuti jätan ma lõuendi sees ruumi mõlema tegelase külgedele, et hiljem rakendada neile sakitõrjet.

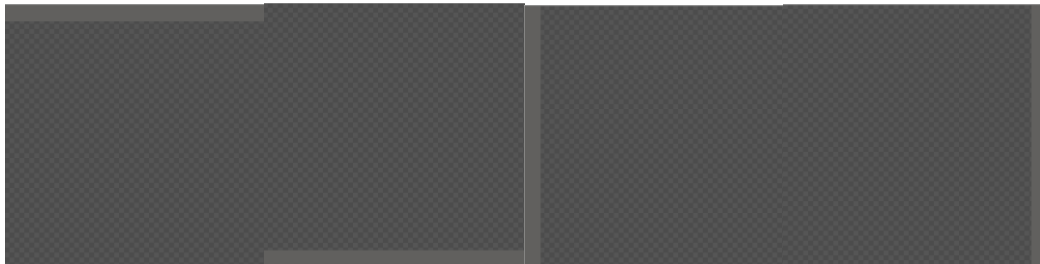


Joonis 17. Lõuend kella piirjoonega

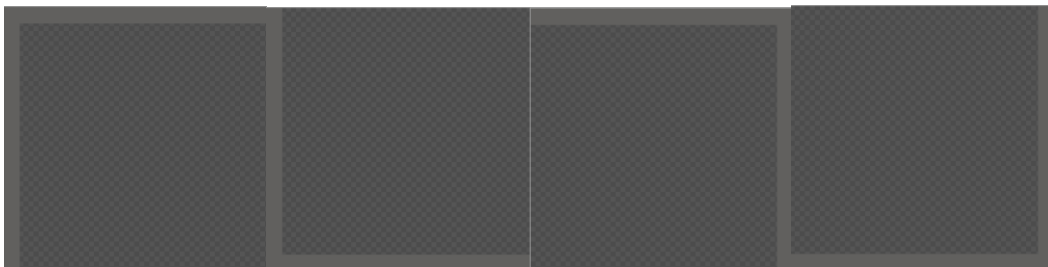
Mängu kolmas oluline element on kohvitass, millega tegelane vaenlast viskab. Selle joonistamiseks ei kasuta ma piirjoont, et see ei oleks visuaalselt liiga suure kaaluga.

Betooni meenutava platvormi joonistamisel ei kasuta ma samuti musta piirjoont, et mitte rikkuda tegelaste ja platvormi vahelist kontrasti. Selle asemel teen platvormi piirjoone tumehalli. Platvormi spraitide joonistamisel võtan arvesse, et see peab piirjoone kaudu lõppema

igast suunast. Joonisel 18 on näha nelja spraiti, mille küljed on kinni ning joonisel 19 on neli spraiti, mille nurgad on kinni.

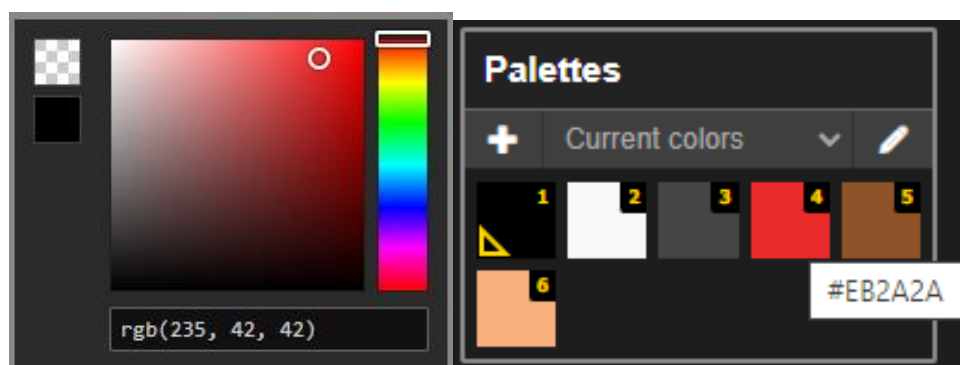


Joonis 18. Ühest küljest suletud piirjoonega platvormi spraidid



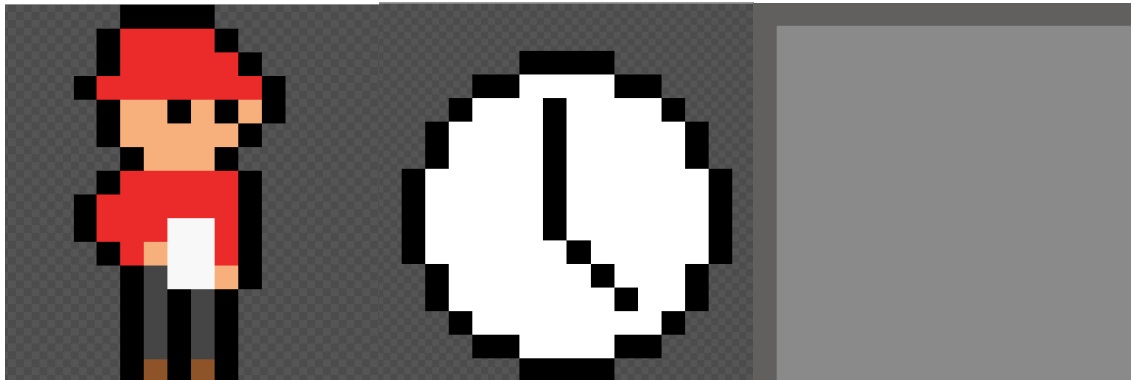
Joonis 19. Nurkadest suletud piirjoonega platvormi spraidid

Spraitide piirjoonte valmimise järel saab alustada nende toonimisega – selleks oli vaja luua värvipalett. Värvipaleti koostamisel võtan arvesse, et traditsiooniliselt on pikslkunstis mängudes värvipalett piiratud ning üritan väheste värvidega hakkama saada. Kindlasti soovin mängus kasutada Tallinna Ülikoolile omast punast värvi, mida kasutan tegelase mütsi ja särgi värvimiseks. Sama värvi korduvaks kasutamiseks on Piskel rakenduses võimalik vaadata iga värvi RGB-koodi ning Color Picker tööriistaga näha ka HEX-värvikoodi (joonis 20).



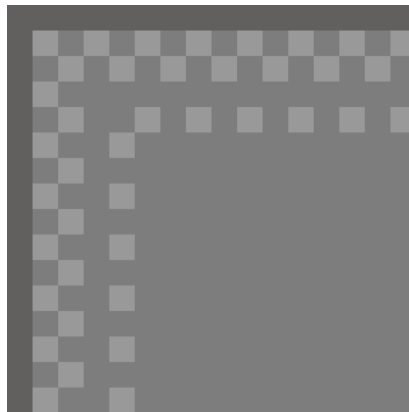
Joonis 20. Color Picker tööriist ja värvipalett

Värvid joonistan eelnevalt kasutatud Pen tööriistaga või Paint Bucket tööriista abil, mis täidab pikslite klasteri või raami sees oleva piirkonna valitud värviga. Joonisel 21 on näha peategelase, vaenlase ja platvormi värvilahendused. Peategelase käes on näha kohvitassi.



Joonis 21. Peategelase, vaenlase ja platvormi värvilahendused

Järgmiseks kasutan oma platvormi spraitidel virvtoonimist, et luua kerge varjutus, mis muudab muidu üksluise platvormi visuaalselt rikkamaks. Selleks joonistan Pen tööriista kasutades heledama halli tooniga piirjoone alla malelaua mustri. Mustrit joonistades teen iga järgneva rea järjest hõredama (joonis 22), mis tekitab sujuva ülemineku.



Joonis 22. Virvtoonimine platvormi spraidil

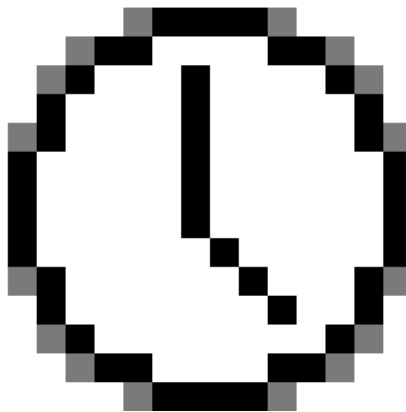
Spraidi peal võib soovi korral kasutada sakitõrjet. Paljudes kujundustarkvara programmides ja mängudes teeb selle protsessi ära automaatne filter, aga Piskel rakenduses tuleb sakitõrje teha nõ käsitsi. Rakendan sakitõrjet oma vaenlase tegelase spraitidele, et ringjoon mõjuks sujuvamana. Esiteks valin värvi, mis oleks poole heledam kui piirjoon. Antud juhul on spraitide

piirjoone värv must, RGB (255, 255, 255) väärtusega. Seega sellest poole heledam värv on kas RGB (123, 123, 123) või RGB (122, 122, 122) väärtusega (joonis 23).



Joonis 23. Sakitõrjeks kasutatud värvitoon

Kasutades Pen tööriista kannan valitud värvi spraidi piirjoone väliserva liitekohtadesse (joonis 24), jälgitades samal ajal, et ei rakenda liiga palju sakitõrjet, sest muidu võib sprait jääda liiga udune.



Joonis 24. Kella spraidi peal rakendatud sakitõrje

#### 2.4.2 Animatsioon Piskel rakenduses

Tegelase ja vaenlase spraitide animeerimise alustamisel koostan tabeli animatsioonidest, mida mängus plaanin kasutada (tabel 2).

Tabel 2. Animatsioonide loetelu

<b>Tegelane</b>	<b>Animatsioon</b>
Tudeng	Seismine
	Jooksmine
	Rünnak
	Hüppamine
	Langemine
	Surm
Ajavaenlane	Liikumine
	Surm

Animeerimiseks kasutan Piskel tarkvaras spraidilehe tüüpi animeerimist, mis tähendab, et kõik kaadrid on eraldi, kuid järjestikku asetatud. Lülitades sisse Onion Skin funktsiooni, saab näha poolläbipaistvalt eelmise raami paiknemist. Selle poolläbipaistva spraidi peale joonistan järgmise kaadri.

Seisva tegelase animatsioon kujutab tegelast kohvi joomas. Iga uue kaadri peale joonistasin kohvitassi piksli laiuse võrra liikumas tegelase suu juurde ja seejärel tagasi (joonis 25). Spraitte animeerides ei pea nii tihedalt kaadreid tegema, aga see aitab luua sujuvama animatsiooni. Tahtsin ka, et tegelane hoiaks oma kohvitassi paar hetke suu juures, seega kordasin seda kaadrit kolm korda.



Joonis 25. Tegelase seisva animatsiooni kaadrid

Sama tehnikat kasutades animeerin ülejäänud tabelis loetletud animatsioonid (vt Lisa 1).

### 2.4.3 Tagatausta loomine Affinity Designer tarkvaras

Tagataust loomiseks kasutan Affinity Designer tarkvara. Soovin tagatausta teha *parallax* stiilis, mis tähendab, et erinevad tagatausta kihid liiguvad erinevate kiirustega tegelasega kaasa ning korduvad horisontaalsel teljel. Värvitooniks valin taas punase, aga teen kihid tegelase punasest värvist kas heledamad või tumedamad, et hoida kontrasti tegelase ja tausta vahel.

Alustan Affinity Designeris lõuendi suuruse valimisega. Tagataust tuleb mängu eraldusvõimega võrdne ehk siis 1920 x 1080 pikslit. Selleks, et taust oleks horisontaalselt katkematu, peaksid taustaelemendid olema nii paremast kui vasakust servast võrdsed. Selle jälgimiseks tuleb Affinity Designeris kasutada Symbols paneeli ning kolm korda laiemat lõuendit (5760 x 1080 pikslit).

Loon lõuendi keskele 1920 x 1080 piksli suuruse elemendi, mille muudan Symbol paneeli abil kopeeritavaks elemendiks. Tõstan ühe koopiat keskele loodud elemendist paremale ja teise vasakule. Avan Layers paneelis esimesena loodud Symbol nime kandva kihi ja joonistan siia alla kõik elemendid, mida soovin oma taustal kasutada. Kõik mida sellele Symbol kihile joonistan, kopeerub automaatselt ka sellest loodud koopiatele paremal ja vasakul. Sel meetodil loon kõik *parallaxi* jaoks vajalikud taustakihid, mida on näha joonisel 26.

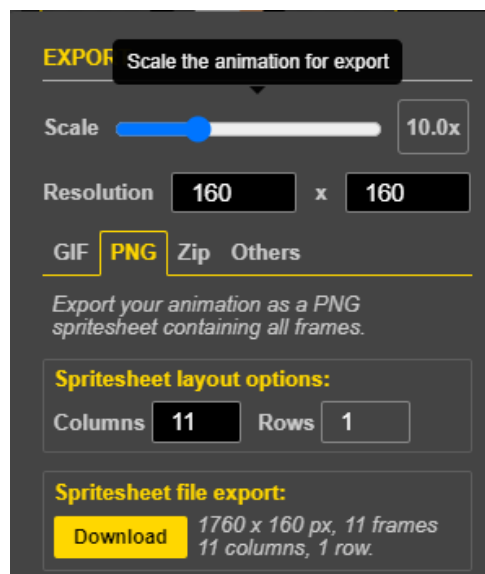


Joonis 26. Taustakihid kolm korda laiemal lõuendil

## 2.5 Graafika eksportimine ja importimine

Kui graafika on valmis, saab alustada selle eksportimist mängumootori jaoks sobivasse pildivormingusse. Kuna kõik olulisemad mängumootorid, mida käsitlesin teoorias, toetavad PNG pildivormingut, siis ekspordin sellesse formaati.

Piskel rakenduses ekspordin lõpetatud animatsiooni tsüklid PNG pildivormingus spraidilehena kahekümnekordse suurendusega, tuginedes otsusele, mille rakendasin eeltöö faasis. Piskel ei võimalda eksportimisel failivormingu seadeid muuta ning ekspordib vaikimisi PNG-32 pildivorminguga, mis toetab osalist läbipaistvust (joonis 27).



Joonis 27. Spraidilehe eksport Piskel rakenduses

Affinity Designer tarkvaras ekspordin iga tausta kihi eraldi PNG pildivormingusse. Siin on seadetes võimalik ka PNG-8 pildivormingu valimine, mis hoiab pildi andmemahu väiksena.

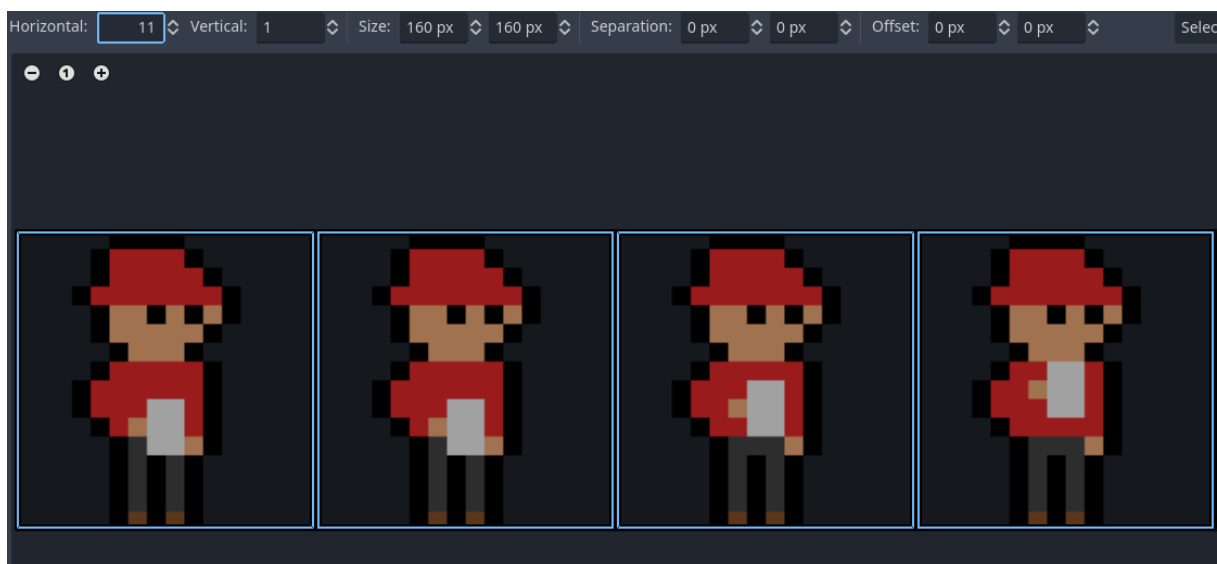
Failide eksportimise järel liigun tööga mängumootorisse. Teoorias tutvustatud mängumootorite seast tegin valiku Godot kasuks, kuna oman sellega varasemat kokkupuudet ning see sobib just eeskätt 2D-videomängude arendamiseks.

Graafika mängumootorisse importimine võimaldab testida elementide omavahelist visuaalset sobivust ning seda, kuidas ekraani ja tegelaskuju vaheline mõõdusuhe toimib tegelaskuju animeerumisel ning liikumisel.



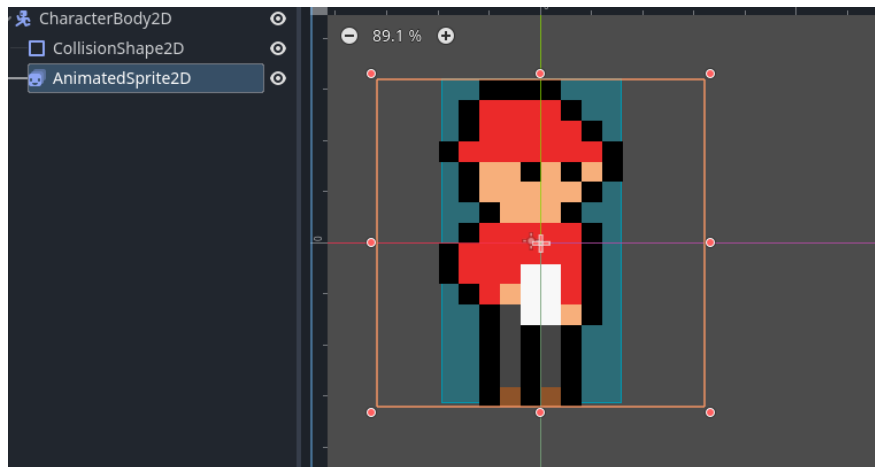
Alguses katsetasin ka kahekümnekordse suurendusega spraidilehtede importimist mängumootoris, aga tegelase suurte mõõtude tõttu läks hüppamisel tegelast järgiva kaamera pilt mängumaailma piiridest välja. Seetõttu tuli töövoos alguses paika määratud mõõtude suhe ümber hinnata. Selline muudatus ei ole suureks probleemiks, sest Piskel rakendus võimaldab kord eksporditud spraidilehte uuesti sisse importida, et seda siis sätete muutmise järel uuesti välja eksportida. Tänu mängumootoris tehtud katsetustele tegin valiku kümnekordse suurendusastme kasuks, mida olen kirjeldanud peatükis 2.3. See olukord toob välja, et töövoog ei ole alati lineaarne ning eeltöö käigus tehtud otsused võivad praktika käigus vajada korrigeerimist.

Loon Godot mängumootoris uue projekti ja määran projekti ekraani eraldusvõimeks 1920 x 1080, mis tugineb eeltöös langetatud otsustele. Tekkinud projektikausta tõstan eelnevalt eksporditud spraidilehed ja tagatausta kihid. Mängumootoris animatsiooni paneelis loon nimetused tegelaste animatsioonidele ja seon need ära vastavate spraidilehtedega. Spraidilehe importimise käigus tuleb see kaadriteks muuta, kasutades spraidilehe vertikaalselt ja horisontaalselt osadeks jaotamist. Joonisel 28 on spraidileht jaotatud vertikaalselt üheks ja horisontaalseks üheteistkümneks osaks. Iga osa vastab ühele kaadrile.



Joonis 28. Kaadriteks jaotatud spraidileht Godot mängumootoris

Godot mängumootoris määrab *CollisionShape2D* tegelase kokkupuute pinna suuruse. Seadistan selle vastavalt tegelase spraidi kõrgusele ja laiuzele. Joonisel 29 näeme sinisega markeeritud *CollisionShape2D* ala vastab tegelase mõõtudele.

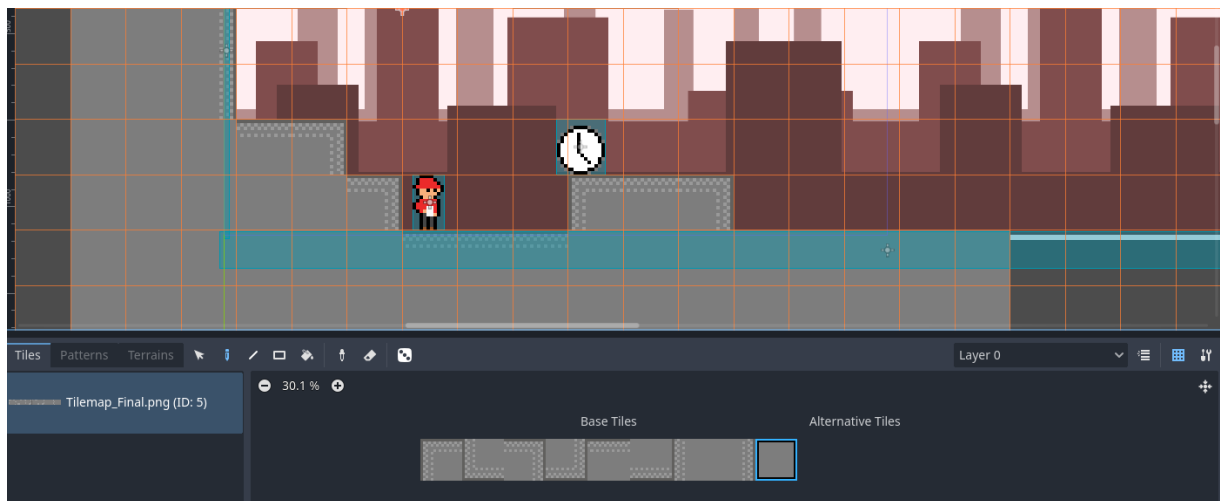


Joonis 29. Tegelase mõõtudele vastav kokkupuute pind.

Kuna Godot võimaldab ka võtmekaadritega (ingl *key frames*) animeerimist, siis on võimalik spraidilehtede kaadreid ümber tõsta, korrata jne ning seeläbi animatsiooni kiirendada või aeglustada. See tähendab, et graafika joonistamise ja animeerimise faasis, mida tegin Piskel rakenduses, ei pea tingimata animatsiooni tempodele mõtlema – oluline on vaid see, et kõik liikumise astmed oleks kaetud.

Tekitan mängumootoris tausta jaoks neli erinevat *parallax* kihti. Igale kihile ühe loodud taustakihi alustades kõige tagumisest. Selleks, et tausta kiht hakkaks horisontaalselt mängumaailmas korduma, tuleb *Mirroring* funktsiooni kasutada, jälgides, et selle laius vastaks tausta kihi laiuzele. Selleks, et erinevad tausta kihid liiguksid erinevate kiirustega ja tekiks *parallax* efekt, peaks iga ettepoole tulev kiht olema eelmisest väiksem.

Platvormi loomiseks tuleb esmalt mängumootorisse platvormi spraidileht *TileMap* funktsiooni kasutades sisse importida ning horisontaalselt ja vertikaalselt osadeks jagada. Platvormi osade abil saab hakata erineva kujuga platvorme genereerima, kasutades *Paint* tööriista. Kursoriga mängumaailma peal klikkides saab määrata platvormi asukohta ja seadetes valida, millise disainiga platvormi osa sinna asetub. Platvormi täpsemaks asetamiseks tuleks kasutada *Grid* süsteemi, mida on näidatud joonisel 30.



Joonis 30. Grid tööriista abil platvormi paigutamine

Sellega on kõik loodud pikselkunsti elemendid mängumaailma sisse viidud ning saab lugeda graafikale suunatud mänguarenduse tööetapid lõppenuks. Olulisemad tähelepanekud praktilisest osast on, et ehkki tööetappidel on loogiline järgnevus, ei pruugi protsess olla tingimata algusest lõpuni lineaarne, kuna kõige olulisemad nüansid ilmnevad mängumootoris. Samuti täheldasin, et animatsiooni kaadrite joonistamise faasis ei pea tingimata tempodega arvestama, kuna seda on võimalik mängumootoris kohandada.

## KOKKUVÕTE

Diplomitöö eesmärk oli anda ülevaade pikslükunsti tööprotsessist lähtuvalt 2D-videomängutööstuse ning enamlevinud mängumootorite vajadustest. Eesmärgi täitmiseks töötasin läbi pikslükunsti alase kirjanduse ja määrasin paika parima töövoogu; tutvusin mängumootorite nõuetega pikslükunsti loomise vaatepunktist ja lõin pikslükunstis graafika 2D-videomängule, et testida teooriate toimimist praktikas.

Diplomitöö esimeses osas andsin ülevaate pikslükunsti mõistest, spraitide disainimise ja animeerimise eripäradest 2D-videomängu tööprotsessis. Hervieux'i (2015), Silberi (2016), Azzi (2019) ja Wolfi (2008) teostele tuginedes töötasin välja optimaalse töövoogu pikslükunsti loomisel: mängu vaatenurga valik; lõuendi suuruse ja eraldusvõime valik; spraitide ja ekraani mõõtude vaheline suhe; piirjoonte loomine; värvipaleti loomine ja värvikasutus; virvtoonimise ja viirutamise kasutamine; sakitõrje kasutamine; spraitide animeerimine ja spraidilehtede koostamine; graafika eksportimine ning mängumootorisse importimine.

Diplomitöö teises osas rakendasin töövoogu praktikasse, luues pikslükunstis 2D-videomängu elemendid. Kasutasin spraitidena animeeritavate elementide loomiseks Piskel veebirakendust, kuid tausta lõin Affinity Designer tarkvaras, mis võimaldas kontrollida elementide omavaheliste liitumiste sujuvust. Valminud spraidilehed importisin PNG-pildivormingus Godot mängumootorisse, kus testisin graafika omavahelist sobivust ja toimivust.

Mängumootoris tehtud katsetused tõid välja probleemi, et eeltöö käigus valitud ekraani ja spraitide vaheline mõõdusuhe polnud praktikas toimiv, mistõttu tuli eksportida graafika teise mõõdusuhtesse kui algselt kavandatud. See olukord tõi välja, et teooria põhjal välja töödatud töövoog pole alati lineaarne ning eeltöö käigus tehtud otsused võivad vajada hilisemat korrigeerimist. Samuti täheldasin, et animatsiooni kaadrite joonistamise faasis ei pea tingimata tempodega arvestama, kuna seda on võimalik mängumootoris kohandada.

Kokkuvõtvalt leian, et tööle seatud eesmärk sai täidetud. Teooriale tuginedes välja töötatud töövoog leidis põhiosas praktikas kinnitust, v.a. esialgne valik ekraani ja tegelase mõõtude suhetes. Järgmisena on plaanis teha diplomitöö käigus kogutud teadmised ja kogemused eestikeelse blogi kujul kõigile pikslükunsti- ja 2D-videomänguhuvilistele avalikult kättesaadavaks.

## SUMMARY

The aim of the diploma thesis was to provide an overview of the pixel art workflow based on the needs of the 2D video game industry and common game engines. To achieve this goal, I studied literature on pixel art creation and established the best workflow based on that. I familiarized myself with the requirements of game engines from the perspective of pixel art creation and created pixel art graphics for a 2D video game to test the theories in practice.

In the first part of the thesis, I provided an overview of pixel art, sprite design and animation in the 2D video game development. Based on the works of Hervieux (2015), Silber (2016), Azzi (2019), and Wolf (2008), I developed an optimal workflow for pixel art creation which is following: selecting the game perspective, choosing canvas size and resolution, establishing the relationship between sprites and screen dimensions, creating outlines, generating a color palette, employing dithering and hatching, applying anti-aliasing, animating sprites, creating sprite sheets, exporting graphics and importing them into the game engine.

In the second part of the thesis, I implemented the workflow step by step into practice by creating pixel art elements for a 2D video game. I used the Piskel web application and Affinity Designer software for pixel art. The completed sprite sheets were imported into the Godot game engine, where I tested the compatibility and functionality of the graphics.

The game engine revealed a problem with the initially chosen ratio between the screen and the sprites, which did not work well in practice. As a result, the graphics had to be exported with a different ratio than originally planned. This situation highlighted that the theory based workflow is not always linear, and decisions made during the pre-production phase may require later adjustments. Additionally, I observed that the animation tempo is not important while drawing animation frames since it can be adjusted in the game engine.

In conclusion, I believe that the goal of the thesis was achieved. The workflow was largely confirmed in practice. The results of the diploma thesis will be made public in the form of a blog.

## ALLIKAD

Azzi, M. (2019). *Pixel Logic : A Guide to Pixel Art*. E-Raamat.

Bowen, J. (2014). *3D Pixel Character*. [2023, mai 21].

<http://jordiebowen.weebly.com/3d-modelling--animation/3d-pixel-character>

Bramble, R. (2022). *What's the Best Game-Making Software for Beginners?* [2023, mai 21].

<https://gamemaker.io/en/blog/best-game-making-software>

Chang, J. (2022). 172 Key Video Games Industry Statistics: 2022 Market Share Analysis & Data. [2023, mai 21].

<https://financesonline.com/video-games-industry-statistics/>

Cheishiru. (2021). *Isometric View Pixel Art Style*. [2023, mai 21].

<https://tips.clip-studio.com/en-us/articles/4969>

GameMaker. (2023). *GameMaker Manual*. [2023, mai 21].

<https://manual.yoyogames.com/#t=Content.htm>

GameMakerBlog. (2009). *Ten Years of Game Maker 1999-2009*. [2023, mai 21].

Godot. (s.a.). *Godot Engine 4.0 documentation in English*. [2023, mai 21].

<https://docs.godotengine.org/en/stable/index.html>

Goodnight, E. (2010). What's the Difference Between Pixels and Vectors? [2023, mai 21].

<https://www.howtogeek.com/32597/whats-the-difference-between-pixels-and-vectors/>

Goodnight, E. Z. (2010) What's the Difference Between Pixels and Vectors? [2023, mai 21].

<https://www.howtogeek.com/32597/whats-the-difference-between-pixels-and-vectors/>

Hervieux, M. (2015). *Learning Pixel Art*. E-Raamat.

Holmes, T. (2023) *What is Aspect Ratio?* [2023, mai 21].

<https://wistia.com/learn/production/what-is-aspect-ratio>

- Jan, M. (2017). *Game Developer's Guide to Graphical Projections*. [2023, mai 21].  
<https://medium.com/retronator-magazine/game-developers-guide-to-graphical-projections-with-video-game-examples-part-1-introduction-aa3d051c137d>
- Lambert, S. (2013). *Introduction to Spritesheet Animation*. [2023, mai 21].  
<https://gamedevelopment.tutsplus.com/tutorials/an-introduction-to-spritesheet-animation--gamedev-13099>
- Linietsky, J. (2019). *A Decade in Retrospective and Future*. [2023, mai 21].  
<https://godotengine.org/article/retrospective-and-future/>
- Lospec. (2019). *Indecision Palette*. [2023, mai 21].  
<https://lospec.com/palette-list/indecision>
- MiniTool. (2023). *PNG 8 VS PNG 24: What is the Distinction and Which Would You Use?* [2023, mai 21].  
<https://moviemaker.minitool.com/moviemaker/png-8-vs-png-24.html>
- Obedkov, E. (2022). *Steam Games Revenue Grew to \$3.1 Billion in First Half of 2022, but Analysts Expect Slight Recession*. [2023, mai 21].  
<https://gameworldobserver.com/2022/07/26/steam-games-revenue-3-1-billion-2022-recession-vg-insights>
- OSKA. (2019). *Ülevaade Eesti tööturu olukorrast, tööjõuvajadusest ning sellest tulenevast koolitusvajadusest*. [2023, mai 21].  
[https://oska.kutsekoda.ee/wpcontent/uploads/2020/05/T%C3%B6%C3%B6j%C3%B5uproгноos-2019\\_2027\\_terviktekst.pdf](https://oska.kutsekoda.ee/wpcontent/uploads/2020/05/T%C3%B6%C3%B6j%C3%B5uproгноos-2019_2027_terviktekst.pdf)
- Peckham E.C. (2019). *How Unity Built the World's Most Popular Game Engine*. [2023, mai 21].  
<https://techcrunch.com/2019/10/17/how-unity-built-the-worlds-most-popular-game-engine>
- Pixel Art Wiki. (s.a.). *Canvas Size*. [2023, mai 21].  
[https://pixelart.fandom.com/wiki/Canvas\\_Size](https://pixelart.fandom.com/wiki/Canvas_Size)

Rouse, M (2015). *Pixel Art*. [2023, mai 21].

<https://www.techopedia.com/definition/8884/pixel-art>

Silber, D. (2016). *Pixel Art for Game Developers*. Taylor & Francis Group.

Standardipõhine tarkvaratehnika sõnastik. (2023). [2023, mai 21].

<https://stats.cyber.ee/term/3983-antialiasing>

Steam(a), (s.a.). *Top Rated Games on Steam*. . [2023, aprill 1].

<https://steamdb.info/stats/gameratings/>

Steam(b), (2023). *Terraria*. [2023, mai 21].

<https://steamdb.info/app/105600/charts/>

Steam(c), (2023). *Steam Hardware & Software Survey: April 2023*.

<https://store.steampowered.com/hwsurvey/Steam-Hardware-Software>

The Computer Graphics and Virtual Reality Study Lab @ University of Tartu. (s.a.). *Anti-Aliasing*. [2023, mai 21].

<https://cgvr.cs.ut.ee/arvutigraafika-terminid/>

Unity Technologies. (s.a.). *Introduction to Tilemaps*. [2023, mai 21].

<https://learn.unity.com/tutorial/introduction-to-tilemaps#>

Unity. (2023). *Unity Documentation*. [2023, mai 21].

<https://docs.unity.com/>

Unreal Engine. (2023). *Unreal Engine 5.2 Documentation*. [2023, mai 21].

<https://docs.unrealengine.com/5.2/en-US/>

Wain, S. (2023). *How to make pixel art for 2D Games*. [2023, mai 21].

<https://gamemaker.io/en/blog/make-pixel-art-2d-games>

Williams, E. (2023). *11 Best Pixel Art Software in 2023*. [2023, mai 21].

<https://fixthephoto.com/best-pixel-art-software.html>

Wolf, M.J.P. (2008). *The Video Game Explosion*. Greenwood Publishing Group.



## LISA 1. DIPLOMITÖÖ KÄIGUS 2D-VIDEOMÄNGU JAOKS LOODUD GRAAFIKA

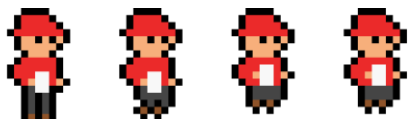
Teglane seisab



Teglane viskab



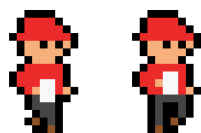
Teglane hüppab



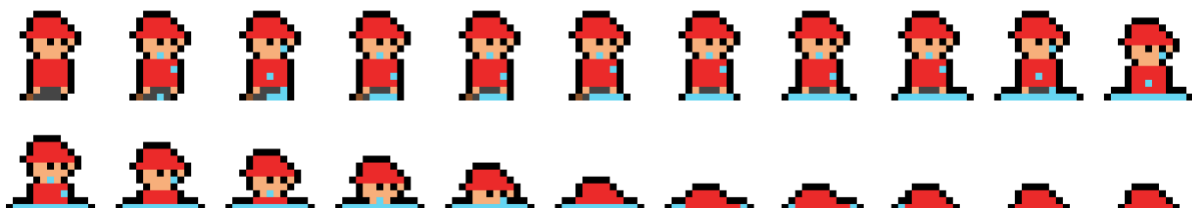
Teglane laskub



Teglane jookseb



Teglane sureb



Vaenlane seisab



Vaenlane sureb



Platvormi spraidileht



Erinevad taustakihid



Lõplik kuvatõmmis mängumootorisse imporditud spraitidest

