



Emil Gusgård

# Tietokoneroolipelin luominen Godot-pelimoottorilla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikan tutkinto-ohjelma

Insinöörityö

2.6.2023

# Tiivistelmä

Tekijä: Emil Gusgård  
Otsikko: Tietokoneroolipelin luominen Godot-pelimoottorilla  
Sivumäärä: 48 sivua  
Aika: 2.6.2023

Tutkinto: Insinööri (AMK)  
Tutkinto-ohjelma: Tieto- ja viestintäteknikka  
Ammatillinen pääaine: Pelisovellukset  
Ohjaaja: Lehtori Miikka Mäki-Uuro

---

Insinöörityön tavoitteena oli luoda tietokoneroolipeli käyttäen ilmaista ja avoimen lähdekoodin Godot-pelimoottoria ja selvittää, soveltuuko pelimoottori ja sen oma ohjelmointikieli GDScript tietokoneroolipelejä varten. Godot-pelimoottorilla on mahdollista luoda 2D- ja 3D-pelejä tietokoneen lisäksi myös iOS- ja Android-puhelimille, tableteille, verkkoselaimelle sekä monille virtuaalitodellisuuden ja lisätyn todellisuuden laitteille.

Godot-pelimoottorin versio 1.0 julkaistiin Githubissa vuonna 2014. Godot-pelimoottorin viimeisin versio 4.0 julkaistiin maaliskuussa 2023. GDScript on syntaksiltaan Python-ohjelmointikielen kaltainen, ja se on nopea, vahvasti editoriin integroitu, eikä sillä ole roskienkeruuta.

Tietokoneroolipelit ovat roolipelejä, joita pelataan tietokoneella. Termiä käytetään yleensä erottamaan tietokoneella pelattavat roolipelit perinteisistä pöytäroolipeleistä ja konsoliroolipeleistä. Tietokoneroolipelit tyypillisesti pyrkivät tarjoamaan pelaajalle jonkinlaisen päätarinan ja sivutarinoita, joihin on mahdollista vaikuttaa esimerkiksi eri dialogivalintojen, tehtävien suoritustapojen tai hahmonkehityksen avulla. Mielenkiintoisen tarinan lisäksi monet tietokoneroolipelit pyrkivät tarjoamaan syvällisen taistelujärjestelmän ja hahmonkehityksen. Tietokoneroolipelejä on tehty 1970-luvulta lähtien tähän päivään saakka, ja ne ovat edelleen suosittuja.

Insinöörityössä toteutettiin ensimmäisen persoonan kolmiulotteinen tietokoneroolipeli. Godot-pelimoottorin omilla raahaa ja pudota -metodeilla on mahdollista toteuttaa tietokoneroolipeleille sopiva esineidenhallintakäyttöliittymä, ja sen Kontrolli-luokalla on mahdollista luoda yksinkertainen käyttöliittymä esineiden, pelaajahahmon ominaisuuksien ja loitsujen näyttämistä ja valitsemista varten.

Intuitiiviseksi todetun GDScriptin avulla luotiin esinehallintajärjestelmä, loitsujärjestelmä ja vuorovaikutusjärjestelmä hyödyntäen pelimoottorin omia metodeja mm. säteenheittoa varten.

Insinöörityö osoitti, että Godot-pelimoottorilla on mahdollista tehdä ainakin pienen mittakaavan kolmiulotteinen ja vähäistä polygonimäärää hyödyntävä tietokoneroolipeli projekti Godot-pelimoottorin versiolla 3.5.1.

Avainsanat: Godot, tietokoneroolipeli, roolipeli, CRPG, RPG

## Abstract

Author: Emil Gusgård  
Title: Creating a Computer Roleplaying Game in Godot Game Engine  
Number of Pages: 48 pages  
Date: 2 June 2023

Degree: Bachelor of Engineering  
Degree Programme: Information and Communications Technology  
Professional Major: Game Applications  
Supervisor: Miikka Mäki-Uuro, Senior Lecturer

---

The goal of the thesis was to create a computer roleplaying game using the free and open-source Godot game engine and analyze whether the Godot game engine and its programming language GDScript is suitable for computer roleplaying games. The Godot game engine can be used to create 2D and 3D games as well games for mobile devices, browsers and various virtual reality and augmented reality devices.

The 1.0 version of the Godot game engine was released in GitHub in 2014. The game engine's latest version 4.0 was released in March 2023. GDScript is similar to the Python programming language in its syntax. It is fast, heavily integrated to its engine and it has no garbage collection.

Computer roleplaying games are roleplaying games that are played on a computer. The term is usually used to differentiate from traditional tabletop roleplaying games and console roleplaying games. Computer roleplaying games typically aim to offer players some kind of main story and side stories that can be affected by different dialog choices, mission completion methods or character development choices for example. Apart from an interesting story many computer roleplaying games aim to offer a deep combat system and character development. Computer roleplaying games have been made since the 1970's and they are still popular.

A small-scale first-person 3D roleplaying game was developed in this study. Using Godot's own drag and drop methods, it was possible to create an item interface and with Godot's Control class it was possible to create a simple user interface to display items, player character attributes and spells.

The GDScript was found intuitive, and with it an inventory system, a spell system and an interaction system were created while using the engine's own methods such as ray-casting methods.

Godot has the capability to implement at least a 3D computer roleplaying game of a smaller scope using a low polygon count with the version 3.5.1 of the engine.

Keywords: Godot, computer roleplaying game, roleplaying game, CRPG, RPG

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Godot-pelimoottori	1
2.1	Historia	2
2.2	Rahoitus, kehitys ja lisenssi	2
2.3	Tulevaisuus	3
2.4	Pelimoottorin ominaisuudet	3
2.4.1	Tuetut ohjelmointikielet ja alustat	3
2.4.2	Editori	4
2.4.3	Solmut ja näkymät	6
2.4.4	Signaalit	7
2.5	GDScript-ohjelmointikieli	9
3	Tietokoneroolipelit	10
3.1	Tietokoneroolipelien määritelmä ja ominaisuudet	10
3.2	Tietokoneroolipelien historia	11
4	Tietokoneroolipelin toteutus	16
4.1	Pelin ominaisuudet	17
4.2	Käyttöliittymä	19
4.3	Pelaajahahmon esineet	20
4.3.1	Esineidenhallinta	20
4.3.2	Esinetietokanta	23
4.3.3	Esine-luokka	25
4.3.4	Saaliinhallinta	25
4.3.5	Vuorovaikutteiset säiliöt	26
4.3.6	Inventaario ja ruudukkosäiliö	27
4.3.7	Varustepaikat	28
4.4	Lisäedut	28
4.5	Loitsujärjestelmä	30
4.5.1	Loitsukirja	30
4.5.2	Loitsu-luokka ja loitsunhallinta	32
4.6	Pelimaailmaan vuorovaikuttaminen	33

4.6.1	Vuorovaikutuksen ohjaus	34
4.6.2	Vuorovaikutteinen-luokka ja vuorovaikutteisten näkymien rakenne	35
4.6.3	Lukot	36
4.6.4	Ovet ja säiliöt	38
4.7	Lopputuloksen arviointi	38
5	Yhteenveto	39
	Lähteet	41

## Lyhenteet

CRPG: *Computer Roleplaying Game*. Tietokoneroolipeli.

RPG: *Roleplaying Game*. Roolipeli.

VR: *Virtual Reality*. Virtuaalitodellisuus.

AR: *Augmented Reality*. Lisätty todellisuus.

3D: *Three Dimensional*. Kolmiulotteinen. Otetaan huomioon vaaka-, pysty- ja syvyysakselit.

2D: *Two Dimensional*. Kaksiulotteinen. Otetaan huomioon vaaka- ja pystyakselit.

WASD: Monissa tietokonepeleissä oletusnäppäimet hahmon tai kameran liikkuttamiseksi.

## 1 Johdanto

Insinööriyön tavoitteena oli tutkia, olisiko Godot-pelimoottorilla mahdollista tehdä tietokoneroolipeli. Työssä selvitettiin, miten toteutetaan roolipeleille ominaiset pelaajahahmon esineet, hahmonkehitys ja loitsut sekä miten vuorovaikutetaan pelimaailmaan hyödyntäen pelihahmon ominaisuuksia. Insinööriyössä perehdyttiin Godot-pelimoottoriin, sen omaan GDScript-ohjelmointikieleen ja sen historiaan. Tämän lisäksi insinööriyöraportissa käydään läpi tietokoneroolipelien määritelmää, niiden ominaisuuksia ja historiaa.

Godot-pelimoottori on ilmainen ja avoimen lähdekoodin pelimoottori, joka mahdollistaa kaksiulotteisten ja kolmiulotteisten pelien luomisen tietokoneelle, iOS- ja Android-puhelimille, tableteille, verkkoselaimelle sekä monille VR- ja AR-laitteille. Projektissa käytettiin Godot-pelimoottorin versiota 3.5.1, joka on yksi viimeisiä pelimoottorin kolmannen version julkaisuja. Viimeinen vakaa versio julkaistiin maaliskuussa 2023. [1.] Insinööriyön toteutusvaiheessa Godot 4:ää ei ollut vielä julkaistu.

Insinööriyön luvussa 2 esitellään Godot-pelimoottori ja pelimoottorin oma GDScript-ohjelmointikieli. Luvussa 3 käydään läpi, mitä tietokoneroolipelit ovat, niiden ominaisuudet ja historia. Godot-pelimoottorilla luotu tietokoneroolipeli esitellään luvussa 4.

## 2 Godot-pelimoottori

Godot-pelimoottori on ilmainen ja avoimen lähdekoodin 2D- ja 3D-pelimoottori. Se on julkaistu MIT-lisenssillä. Pelimoottorin ovat kehittäneet Juan Linietsky ja Ariel Manzur [2].

## 2.1 Historia

Godot-pelimoottorin 1.0 versio julkaistiin Githubissa avoimen lähdekoodin projektina helmikuussa 2014 [3]. Godot-pelimoottorin versio 2.0 julkaistiin vuonna 2016, ja se lisäsi mm. usean näkymän (engl. scene) muokkauksen yhtä aikaa editorissa ja paremman näkymien ilmentymisen (engl. instanciate) ja periytymissen [4]. Tammikuussa 2018 julkaistiin Godot 3.0, joka lisäsi mm. virallisen C#-ohjelmointikielen tuen, uudistetun fysiikkamoottorin ja VR-tuen [5]. Maaliskuussa 2023 julkaistiin Godot 4.0, joka erityisesti muuttaa pelimoottorin grafiikkaa Vulkan-ohjelmointirajapinnan avulla. Tämän lisäksi Godot-pelimoottorin oman GDScript-kielen käyttöä ja suorituskykyä parannettiin ja fysiikka- ja animaatiojärjestelmiä kehitettiin eteenpäin. [6.]

## 2.2 Rahoitus, kehitys ja lisenssi

Godot-pelimoottorin lataaminen ja käyttö on täysin ilmaista, ja sitä voi muokata ja jakaa miten haluaa, kunhan jakaa tekijänoikeustiedotteen ja lisenssiehdot [2].

Godot-pelimoottorin kehitystä tuetaan Patreon-joukkorahoitusalustalla, jonka avulla Godot-pelimoottorin kehittäjät saavat lähes 13 000 euroa kuukaudessa, mikä heidän mukaansa mahdollistaa kolmen täysiaikaisen ohjelmoijan palkkaamisen projektiin [7]. Tämän lisäksi pelimoottoria voi tukea rahallisesti sen omilla sivuilla [8].

Godot on saanut suuria lahjoituksia yksityisiltä yrityksiltä ja organisaatioilta, kuten Firefox-verkkoselaimestaan tunnetulta Mozillalta [9] ja 250 000 dollarin lahjoitus Unreal-pelimoottoristaan tunnetulta Epic Gamesiltä sen Epic MegaGrant-ohjelman kautta [10].

## 2.3 Tulevaisuus

Pelimoottorin pääkehittäjä Juan Lienietskyn vuonna 2023 pitämän GDC-puheen varovaisen arvion mukaan Godot-pelimoottorilla julkaistujen pelien määrä kasvaa noin 50 % vuosittain. Tämän arvion perusteella Godot-pelimoottorilla vuosittain julkaistujen pelien määrä vastaisi Unityn määrää vuonna 2028 (yli 4 000 peliä julkaistu vuonna 2028). [11.]

## 2.4 Pelimoottorin ominaisuudet

### 2.4.1 Tuetut ohjelmointikieliset ja alustat

Godot-pelimoottori tukee virallisesti neljää eri ohjelmointikieltä: GDScript, C#, C ja C++ [12]. Yhteisö on luonut ohjelmointikielitettyjä myös monille muille kielille, kuten esimerkiksi Lisp-ohjelmointikielelle [13] ja Lua-ohjelmointikielelle [14].

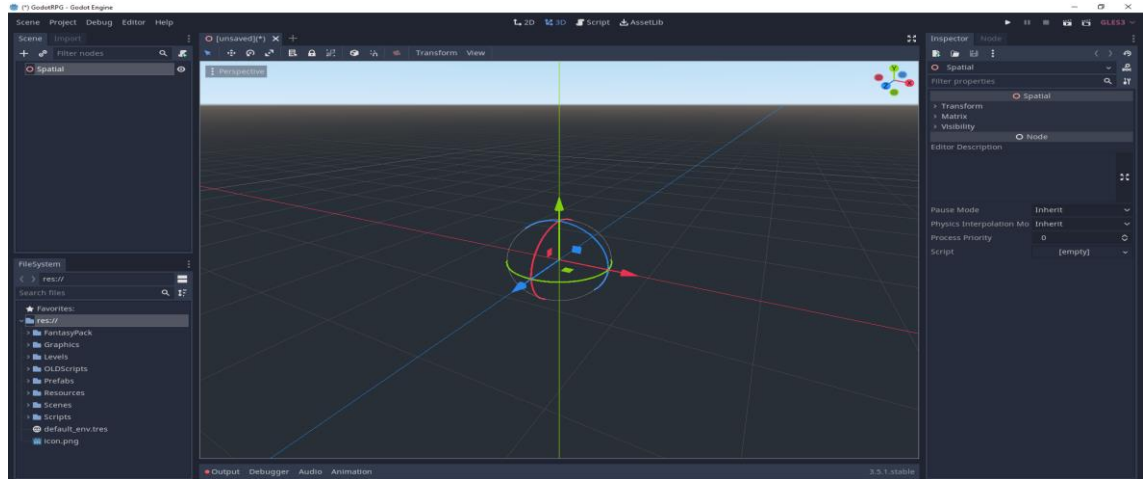
Pelimoottori tuki myös Godot-pelimoottorin kolmannessa versiossa sen omaa visuaalista skriptauskieltä, VisualScriptiä, mutta sen virallista tukea ei jatkettu Godot 4.0:n julkaisun jälkeen, sillä sen käyttö oli hyvin marginaalista. Ennen lopetuspäätöstä julkaistun kyselyn mukaan vain 0,5 % käyttäjistä käytti VisualScriptiä pääasiallisena pelimoottorin kielenä. [15.]

Godot-pelimoottorilla voi tehdä pelejä Android- ja iOS-mobiilialustoille, Windows-, Mac- ja Linux-tietokoneille, verkkoselaimelle HTML5:n avulla sekä monille eri VR- ja AR-alustoille [16].

Godot-pelimoottorilla ei ole virallista konsolitukea, sillä Juan Linietskyn mukaan pelimoottorin vapaan ja avoimen luonteen se ei ole mahdollista. Kuitenkin on olemassa yrityksiä, kuten Lone Wolf Technology ja Pineapple Works, jotka ovat erikoistuneet konsoliversioiden luontiin. [17.]

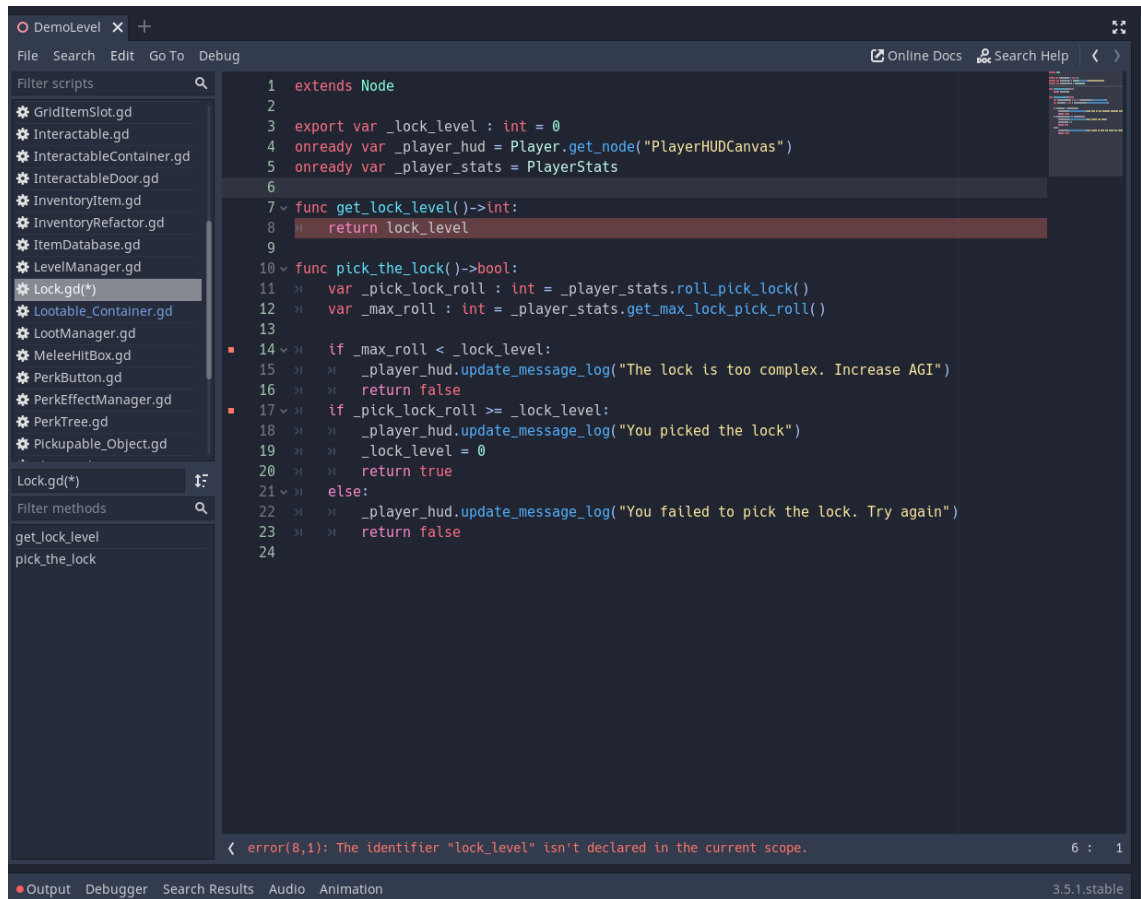
## 2.4.2 Editori

Kuten monet muut modernit pelimoottorit Godot-pelimoottori tarjoaa 2D- ja 3D-katsauksen tasoihin tai näkymiin (engl. scene) (kuva 1).



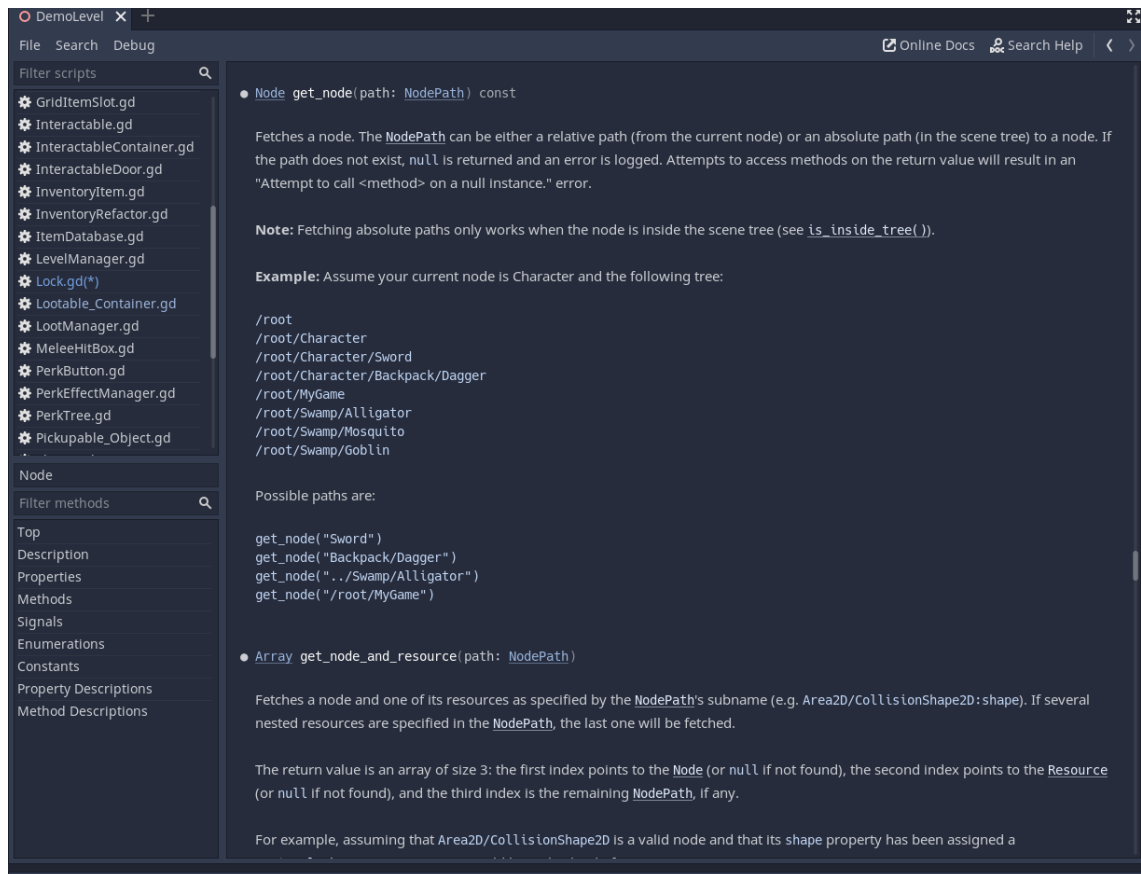
Kuva 1. Editorin 3D-näkymä.

Godot-pelimoottorissa on oma sisäänrakennettu tekstieditori ja kehitysympäristö. Se voi näyttää koodin syntaksivirheet ja mahdollistaa koodin testaamisen virheidenjäljittäjällä (engl. debugger) (kuva 2).



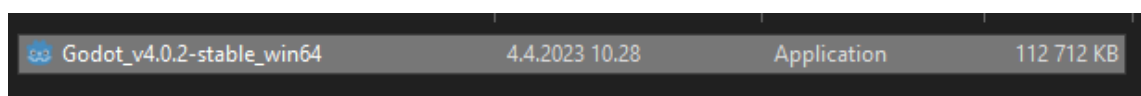
Kuva 2. Editorin sisäänrakennettu kehitysympäristö. Punaiset pisteet koodin vasemmassa marginaalissa kuvastavat virheenjäljityskohtia. Alhaalla ilmoitetaan syntaksivirheestä ja virherivi (rivi 8) on maalattu punaisella.

Kehitysympäristö tarjoaa myös pääsyn Godot-pelimoottorin dokumentaatioon. Painamalla CTRL-näppäintä pohjassa samalla kun painetaan vasemmalla hiirenappulalla jotain Godot-pelimoottorin omaa metodia, voidaan kätevästi päästä käsiksi kyseisen metodin dokumentaatioon (kuva 3).



Kuva 3. Godotin oman `get_node`-metodin dokumentaatio.

On mainitsemisen arvoista, että Godot-pelimoottori on huomattavan pienikokoinen. Sen uusin versio (4.0.2) Windows-käyttöjärjestelmälle vie vain 112 megatavua tilaa (kuva 4).



Kuva 4. Godot-pelimoottorin sovellus Windows-käyttöjärjestelmässä.

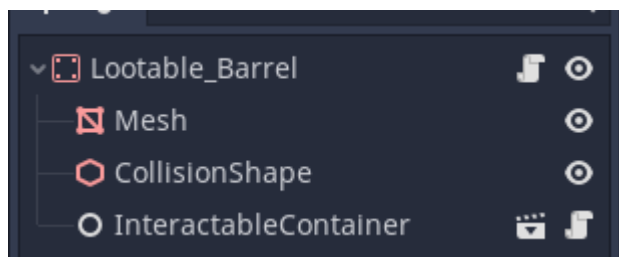
### 2.4.3 Solmut ja näkymät

Godot-pelimoottorin perusluokka on solmu. Jokainen skripti oletuksena perii solmuluokan.

Solmut ovat Godot-pelimoottorin tärkein luokka. Solmuilla voi olla lapsisolmuja, joista muodostuu puurakenne. Solmulla voi olla niin monta lapsisolmua kuin halutaan. Useasta solmusta muodostuvaa puurakennetta kutsutaan näkymäksi (engl. scene), ja niistä voidaan luoda ilmentymiä (engl. instantiate) muihin näkymiin. [18.]

Godot-pelimoottorin näkymä voi olla esimerkiksi käyttöliittymä, pelimaailman hahmo tai kokonainen kenttä. Ne ovat joustavia, ja on myös mahdollista asettaa näkymiä näkymien sisälle. [19.]

Esimerkiksi projektin Tynnyri-näkymä sisältää kaksi lapsisolmua, Mesh ja CollisionShape, sekä InteractableContainer-näkymän lapsinäkymänä (kuva 5).



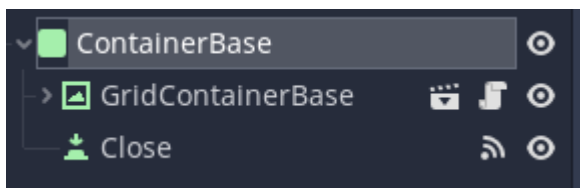
Kuva 5. Esimerkki näkymästä. Kääro-kuvake kuvastaa, että solmuun on asetettu skripti, ja klaffi-kuvake kuvastaa näkymää.

Toisin kuin esimerkiksi Unity-pelimoottorissa, Godot-pelimoottorissa ei ole tehty jakoa peliobjektien puurakenteiden [20] ja pelimaailman kenttien välillä, jossa peliobjektit sijaitsevat [21]. Godot-pelimoottorin näkymä voi toteuttaa molempia toimintoja.

#### 2.4.4 Signaalit

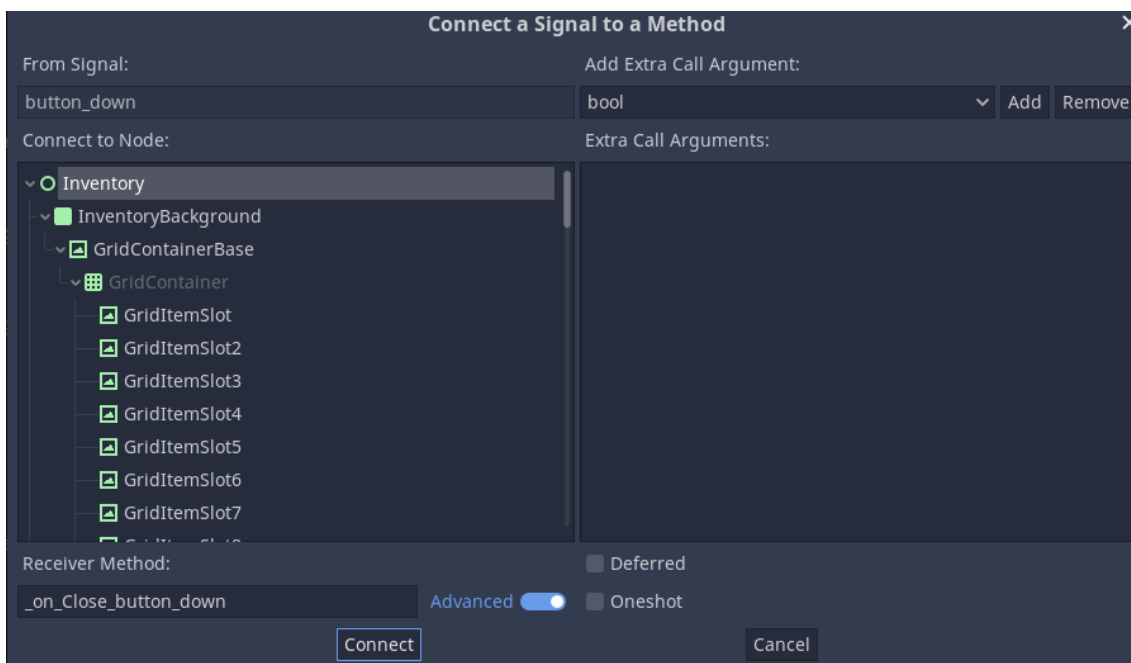
Godot-pelimoottorin signaalit ovat keino toteuttaa tarkkailijamalli (engl. observer pattern). Tarkkailijamallin avulla voidaan ilmoittaa muille olioille, että tiettyjä olioita on muokattu [22]. Signaaleilla peliobjekti voi reagoida toisen peliobjektin muutokseen ilman viittauksia [23].

Signaalit voidaan asettaa editorissa mihin tahansa solmuun. Signaaleja sisältävät solmut kuvataan omalla signaali-kuvakkeella (kuva 6.)



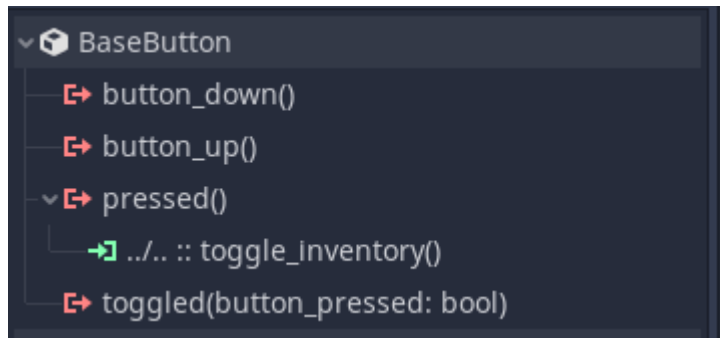
Kuva 6. Esimerkkinä Close-solmuun asetettu signaali editorissa.

Oletuksena signaalia luodessa editorissa se pyrkii luomaan valitun solmun skriptiin vastaanottaja- tai kuuntelijametodin (kuva 7).



Kuva 7. Vastaanottaja-metodin luonti inventaario-solmun skriptiin.

Signaali voidaan myös asettaa kutsumaan tiettyjä funktioita suoraan. Esimerkiksi kuvassa 8 (kuva 8.) BaseButton-luokasta periytyvän solmun metodi napin alas painamiselle (pressed-metodi) on asettanut näkymän Inventaario-luokan ilmentymän kuuntelemaan napin painallusta. Kun nappia painetaan, asetetun luokan metodi (toggle\_inventory) kutsutaan.



Kuva 8. Pressed-metodia kuunnellaan asetetun solmun metodissa.

Luotuja signaaleja voidaan nähdä pelimoottorin oman kehitysympäristön skriptieditorissa, mikä helpottaa hahmottamaa, missä signaaleja on käytetty (kuva 9).

```

62 #using signals for Close Inventory Button
-> 63 func toggle_inventory():
64     > GameManager.toggle_inventory()
65
  
```

Kuva 9. Vihreä kuvake vasemmalla kuvastaa, että kyseistä metodia kutsutaan jossain signaalissa.

## 2.5 GDScript-ohjelmointikieli

GDScript on Godot-pelimoottorin oma ohjelmointikieli, joka tarjoaa mm. nopean lataamisen ja kääntämisen, yksinkertaisen syntaksin, vahvan editori-integraation sekä muuttujien vahvan ja dynaamisen tyyppityksen. Tämän lisäksi sillä ei ole ollenkaan roskienkeruuta. GDScript-koodi näyttää Python-ohjelmointikielen kaltaiselta. [24.]

Kuvassa 10 on ote projektissa käytetystä GDScript-koodilla luodusta metodista `roll_a_crit`. Metodeihin voidaan määrittää nuolioperaatiolla paluuarvo (bool), ja metodissa esitellyt muuttujat (`item_modifiers` ja `roll`) ovat vahvasti tyyditettyjä kaksoispisteellä ja halutulla tyyppillä. GDScriptissä ei tarvitse käyttää sulkeita esim. ehtolauseisiin tai puolipisteitä määrittääkseen lauseen lopun. Sisennyksillä on

GDScriptissä väliä, ja niiden väärinkäyttö voi aiheuttaa ohjelmointikoodin käännös-  
virheitä. Niillä ilmaistaan mm. ehtolauseiden toteutus.

```
func roll_a_crit(item_modifiers : float)->bool:  
  rng.randomize()  
  var roll : int = rng.randi_range(0,100)  
  if roll <= item_modifiers + der_crit_chance_mod:  
    player_hud.update_message_log("CRITICAL HIT!")  
    return true  
  else:  
    return false
```

Kuva 10. Ote GDScript-koodista käyttäen vahvasti tyyplitettyjä muuttujia ja pa-  
luuarvoja.

### 3 Tietokoneroolipelit

Tietokoneroolipelit on laaja aihe, joten rajauksena pääasiassa verkossa pelatta-  
vat tietokoneroolipelit ja MUD-pelit (Multi-user Dungeon) jätetään työn ulkopuo-  
lelle ja keskitytään lähinnä länsimaisiin tietokoneroolipeleihin.

#### 3.1 Tietokoneroolipelien määritelmä ja ominaisuudet

Tietokoneroolipelit ovat roolipelejä, joita pelataan tietokoneella. Termiä käyte-  
tään yleensä erottamaan tietokoneella pelattavat roolipelit perinteisistä pöy-  
täroolipeleistä ja konsoliroolipeleistä.

Tietokoneroolipelit tyypillisesti pyrkivät tarjoamaan pelaajalle jonkinlaisen pääta-  
rinan päätehtävineen sekä sivutarinoita tai sivutehtäviä. Ne eroavat muista pe-  
leistä siten, että usein niissä on mahdollista vaikuttaa tarinankulkuun esimer-  
kiksi eri dialogivalintojen, tehtävien suoritusstapojen tai hahmonkehityksen  
avulla.

Mielenkiintoisen tarinan lisäksi monet tietokoneroolipelit pyrkivät tarjoamaan sy-  
vällisen taistelujärjestelmän ja hahmonkehityksen. Suorittamalla tehtäviä tai päi-

hittämällä vihollisia pelihahmo tai pelihahmot saavat usein kokemuspisteitä kehittämään hahmoa eteenpäin, jolloin suurempien haasteiden voittaminen on mahdollista

Tietokoneroolipelien alalajeja on monia, mm. toimintaroolipelit [25], jotka keskittyvät reaaliaikaiseen taisteluun toimintapelien tyyllillä, kuten esimerkiksi The Elder Scrolls V: Skyrim [26], sekä luolastoroolipelit [27], jotka keskittyvät luolastojen tutkimiseen ja usein myös niissä ratkaistaviin pulmiin. Hyviä esimerkkejä alalajista ovat mm. Wizardry-sarjan pelit [28] ja suomalainen Legend of Grimrock -sarja [29]. Tämän lisäksi tietyt pelit, kuten vuoden 1980 Rogue [30] ja vuoden 1997 Diablo [31, s. 240] loivat omiin nimiinsä perustuvat alalajit, Rogue-like [30] ja Diablo-like tai Diablo-kloonit. Roguelike-alalajin keskeisenä ominaisuutena on se, että pelaajahahmon tai hahmojen kuoleman tai häviön jälkeen peli on aloitettava alusta. Diablo-pelien kaltaiset tietokoneroolipelit puolestaan keskittyvät usein satunnaisesti luotujen luolastojen tutkimiseen ja koukuttavaan hahmojenkehitykseen.

## 3.2 Tietokoneroolipelien historia

### 1970-luku

Ensimmäiset tietokoneroolipelit kehitettiin vuonna 1974 (samana vuonna kuin Dungeons & Dragons -roolipeli julkaistiin ensi kerran) yliopistojen suurtietokoneilla. Yksi ensimmäisistä oli PLATO-suurtietokoneella kehitetty pedit5, joka oli Rusty Redfordin kehittämä. Monet roolipelien perusominaisuudet, kuten hirviöiden päihitys, aarteiden löytäminen, luolaston tai tyrmän tutkiminen ja loitsujen taikominen, oli mahdollista. [30.]

Vuonna 1978 julkaistiin yksi ensimmäisistä tietokoneroolipeleistä kotitietokoneille, Beneath the Apple Manor, jonka loi Don Worth Apple II -tietokoneelle [32, s. 40].

## 1980-luku

Matt Bartonin [30] mukaan yksi tunnetuimpia roolipelejä oli vuonna 1980 kehitetty UNIX-peli Rogue, jonka loivat Michael Toy, Glenn Wichman ja Ken Arnold. Rogue on tunnettu sen satunnaisista luolastoista, ASCII-grafiikoista sekä monimutkaisesta pelattavuudesta. [30.] Roguelike-pelit ovat varsin suosittuja tänäkin päivänä.

Wizardry-sarjan ensimmäisen pelin Wizardry: Proving Grounds of the Mad Overlord julkaisi vuonna 1981 Sir-Tech-pelyhtiö. Wizardry oli ensimmäinen peli, jossa pelaaja ohjaa kokonaista seuruetta eikä vain yhtä hahmoa, tutkimaan vaarallista tyrmiä. Wizardry oli yksi myydyimpiä 1980-luvun tietokonepelejä. [33, s. 54.]

Richard Garriot kehitti ensimmäisen Ultima-tietokoneroolipelin vuonna 1981. Pelaajan tehtävänä on päihittää paha, kuolematon velho nimeltä Mondain keräämällä neljä taikajalokiveä, jotta voidaan mennä ajassa taaksepäin tappaamaan velho, ennen kuin se saa voimia. [34, s. 56.]

Muita tunnettuja roolipelisarjoja, jotka saivat alkunsa 1980-luvulla, olivat mm. Interplayn vuonna 1985 julkaisema Bard's Tale [35, s. 78] ja New World Computing julkaisema Might & Magic – Book I vuonna 1986, joka oli Wizardryn kaltainen mutta antoi pelaajien tutkia avointa ulkomaailmaa [36, s. 88].

Tunnetun Fallout-pelisarjan henkisen edeltäjän Wastelandin julkaisi vuonna 1988 Interplay. Wasteland sijoittuu ydinsodan runtelemaan maailmaan. Muistirajoitusten vuoksi suurin osa pelin tekstistä löytyy pelin painetusta ohjekirjasta. Peli ohjeistaa pelaajan lukemaan tietyn kappaleen tarpeen mukaan. [37, s. 106–107.]

## 1990-luku

1990-luvulla tietokoneroolipelit olivat pääosin kaksiulotteisia, usein ylhäältä kuvattuina tai ensimmäisestä persoonasta, mutta myös ensimmäiset kolmiulotteiset roolipelit julkaistiin teknologisen kehityksen ansiosta.

Vuonna 1992 Blue Sky Productions kehitti ensimmäisen ”aidon” kaupallisen kolmiulotteisen tietokoneroolipelin Ultima Underworldin. Peli tarjosi mm. vapaan 3D-liikkumisen, vertikaalisuuden rinteillä ja kallioilla, täysin teksturoidut ympäristöt sekä fysiikkajärjestelmän, joka vaikutti kaikkiin liikutettaviin esineisiin. [38, s. 166.]

The Elder Scrolls -pelisarja sai alkunsa vuonna 1994 [39], mutta sen jatko-osa mullisti tietokoneroolipelejä. Vuonna 1996 Bethesda Softworks kehitti The Elder Scrolls II: Daggerfallin, jossa on proseduraalisesti luotu maailma, joka sisältää yli 15 000 kylää ja tyrmää, yli 750 000 hahmoa sekä lukuisia kiltoja ja muita järjestöjä. Siinä on lisäksi epälineaarinen tarina, jolla on kuusi eri lopetusta. [40, s. 236.]

Blizzard Northin kehittämä Diablo vuodelta 1997 on yksinkertainen, mutta koukuttava roolipeli, jonka hirviöiden tappamiseen, esineiden löytämiseen ja hahmonkehittämiseen perustuva pelattavuussilmukka (engl. gameplay loop) loi alalajin pelejä, joita kutsutaan Diablo-klooneiksi [31, s. 240].

Interplayn henkinen jatko-osa vuoden 1988 Wastelandille, Fallout vuodelta 1997, herätti henkiin kuolevan genren. Se tarjosi maailmanlopunjälkeisen pelimaailman, joka perustui Yhdysvaltojen 1950-luvun retrofuturismiin synkällä ja ironisen optimistisellä otteella. Falloutin innovaatio oli sen tarjoamat useat ja laadukkaat roolipelivalinnat. Kehittäjien tavoitteena oli, että jokainen tehtävä voidaan suorittaa vähintään kolmella eri tavalla: taistelulla, puhumalla tai hiipimällä. [41, s. 242.]

Biowaren Infinity Enginellä pyörivä Baldur's Gate vuodelta 1998 on tietokone-roolipeli, joka pohjautuu suosittuihin Advanced Dungeons & Dragons -sääntöihin. Pelin taistelu on reaaliaikaista ja muistuttaa reaaliaikaisia strategiapelejä, mutta sen voi myös pysäyttää. Pelimaailma koostuu yhdistetyistä kartoista, joita pelaaja voi kuuden hengen joukkueella tutkia vapaasti. [42, s. 254.] Vuonna 2000 peli sai jatko-osan Baldur's Gate II: Shadows of Amn [43, s. 282]. Baldur's Gaten Infinity Engine -moottoria käyttivät myös Planescape: Torment (1999) [44] ja Icewind Dale 1 (2000) [45] ja sen jatko-osa Icewind Dale 2 (2002) [46].

## 2000-luvun alku

2000-luvun alkaessa kolmiulotteiset roolipelit alkoivat yleistymään entisestään, ja vuosikymmenen lopussa ne olivat genren valtavirtaa. Monet vuosikymmenen peleistä julkaistiin tietokoneen lisäksi myös konsoleille.

Ion Stormin Deus Ex vuodelta 2000 tarjosi pelaajanvapautta ja reaktiivisuutta, mikä saa pelimaailman vaikuttamaan aidolta. Pelaaja voi mm. räjäyttää oven aseiden kanssa, tiirikoida lukon ja hiipiä, pinota laatikoita tai hypätä päästäkseen katolle tai hakkeroida suorittaakseen tehtäviä pelissä. Pelin juoni käsittelee mm. salaliittoja, terrorismia ja poliittista sortamista. Pelin muita aiheita ovat poliittinen filosofia, uskonto, historia ja transhumanismi. [47, s. 280–281.]

Biowaren Neverwinter Nights julkaistiin vuonna 2002, ja Baldur's Gaten tapaan se perustui Dungeons & Dragons -roolipelin sääntöihin ja oli taistelujärjestelmältään yhä reaaliaikainen, mutta pysäytettävissä. Kuitenkin peli tarjosi voimakkaan Aurora-työkalusarjan pelaajille omien seikkailujen luomiseksi sekä jopa enimmillään 96 pelaajan moninpelin. [48, s. 318.]

Bioware alkoi vuodesta 2003 eteenpäin keskittymään tarina- ja hahmovetoisiin roolipeleihin ja julkaisemaan pelinsä myös konsolialustoille. Star Wars: Knights of The Old Republic on ensimmäinen Star Warsiin perustuva tietokoneroolipeli ja kehitetty konsolialustaystävällisemmäksi kuin Biowaren aiemmat pelit [49, s.

330]. Tämän lisäksi yritys kehitti kiinalaisesta wuxia-genrestä inspiroituneen roolipelin Jade Empiren vuonna 2005 [50, s. 362].

Bethesda Softworksin The Elder Scrolls -sarja sai kaksi jatko-osaa: The Elder Scroll III: Morrowind vuonna 2002 [51] ja The Elder Scrolls IV: Oblivion vuonna 2006 [52]. Molemmat osat olivat teknologisia harppauksia avointen 3D-maailmojen luomisessa. Vuonna 2004 Bethesda Softworks sai Interplayltä oikeuden julkaista Fallout 3:n [53] ja lopulta julkaisi sen vuonna 2008 [54].

2000-luvun alkupuolella Bioware aloitti kaksi eri roolipelitrilogiaa, Mass Effect ja Dragon Age. Vuonna 2007 alkoi Biowaren sci-fi-roolipelitrilogia Mass Effect, joka jatkui vuoteen 2012 asti ja oli miljoonien fanien suosiossa. Peli loi ”dialogi-rattaaan”, joka helpotti dialogin valintaa, ja peli pyrki olemaan mahdollisimman ”elokuvamainen”. [55, s. 386.] Puolestaan vuonna 2009 julkaistiin Dragon Age: Origins, joka oli fantasiamaailmaan sijoittuva henkinen jatkaja Baldur’s Gate -sarjalle. Pelissä oli kunnianhimoinen johdanto-osuus, joka riippui vahvasti pelaajan luoman hahmon rodun, hahmoluokan ja sosiaalisen statuksen valinnoista [56, s. 416.]

## 2010-luku

2010-luvulla pelejä alettiin myymään digitaalisesti ja useita pelejä rahoitettiin joukkorahoituksen avulla mm. Kickstarter-, Indiegogo- ja Patreon-alustoilla. Pelimoottorit ja konsolikehitys alkoivat halventua, ja osa moottoreista tarjottiin ilmaiseksi. Syntyi niin sanottu indiekupla. [57, s. 430.]

Perinteisten suuren budjetin tietokoneroolipelien osalta The Elder Scrolls V: Skyrim ja The Witcher 3 tekivät myyntiennätyksiä. Skyrim oli myynyt vuoden 2016 loppuun mennessä yli 30 miljoonaa kopiota maailmanlaajuisesti [58], ja The Witcher 3, joka julkaistiin vuonna 2015, myi 2019 mennessä yli 28 miljoonaa kopiota [59].

Tietokoneroolipelien osalta vuotta 2014 pidetään niin sanotun tietokoneroolipelirenessanssin ensimmäisenä vuotena, ja silloin julkaistiin mm. Divinity Original

Sin, Wasteland 2 ja Legend of Grimrock II [60]. Legend of Grimrock -pelisarja on suomalaisen Almost Human Gamesin luoma, ja studio perustettiin vuoden 2011 alussa [61].

Muita hyvin arvostettuja joukkorahoitettuja tietokoneroolipelejä olivat mm. Pillars of Eternity (2015) [62], sen jatko-osa Pillars of Eternity II: Deadfire (2018) [63], Divinity Original Sin 2 (2017) [64] ja Kingdom Come: Deliverance (2018) [65]. Vuosikymmenen lopussa, vuonna 2019, julkaistiin neljä The Game Awards -palkintoa saanut Disco Elysium [66]. Se on virolaisen ZA/UM Studion kehittämä et-siväroolipeli [67].

## 2020-luvun alku ja tulevaisuus

2020-luvun alun tunnetuimpia suuren budjetin tietokoneroolipelejä ovat mm. The Witcher 3 -pelin kehittäjien Cyberpunk 2077 (2020) [68], FromSoftware'n Elden Ring (2022) [69] ja helmikuussa 2023 julkaistu Hogwarts Legacy [70]. Perinteisempien, pienen tai keskikokoisen budjetin, tietokoneroolipelien joukossa julkaistiin mm. Pathfinder: Wrath of the Righteous (2021) [71] ja Encased (2021) [72].

Ottaen huomioon, kuinka hyvin esimerkiksi Elden Ring (16,6 miljoonaa kappaletta vuoden 2022 kesäkuuhun mennessä) [73] ja Cyberpunk 2077 (yli 20 miljoonaa kappaletta vuoden 2022 syyskuuhun mennessä) [74] myivät, voidaan olettaa, että tietokoneroolipelit ovat tulleet jäädäkseen.

## 4 Tietokoneroolipelin toteutus

Insinööriyön tarkoituksena oli selvittää, onko Godot-pelimoottori sopiva tietokoneroolipelin toteuttamiseen, ja tätä varten toteutettiin pienen mittakaavan kolmiulotteinen tietokoneroolipeli. Tässä luvussa käydään läpi, miten Godot-pelimoottorin editorilla ja sen omilla metodeilla voidaan toteuttaa tietokoneroolipeleille tyypillisiä ominaisuuksia.

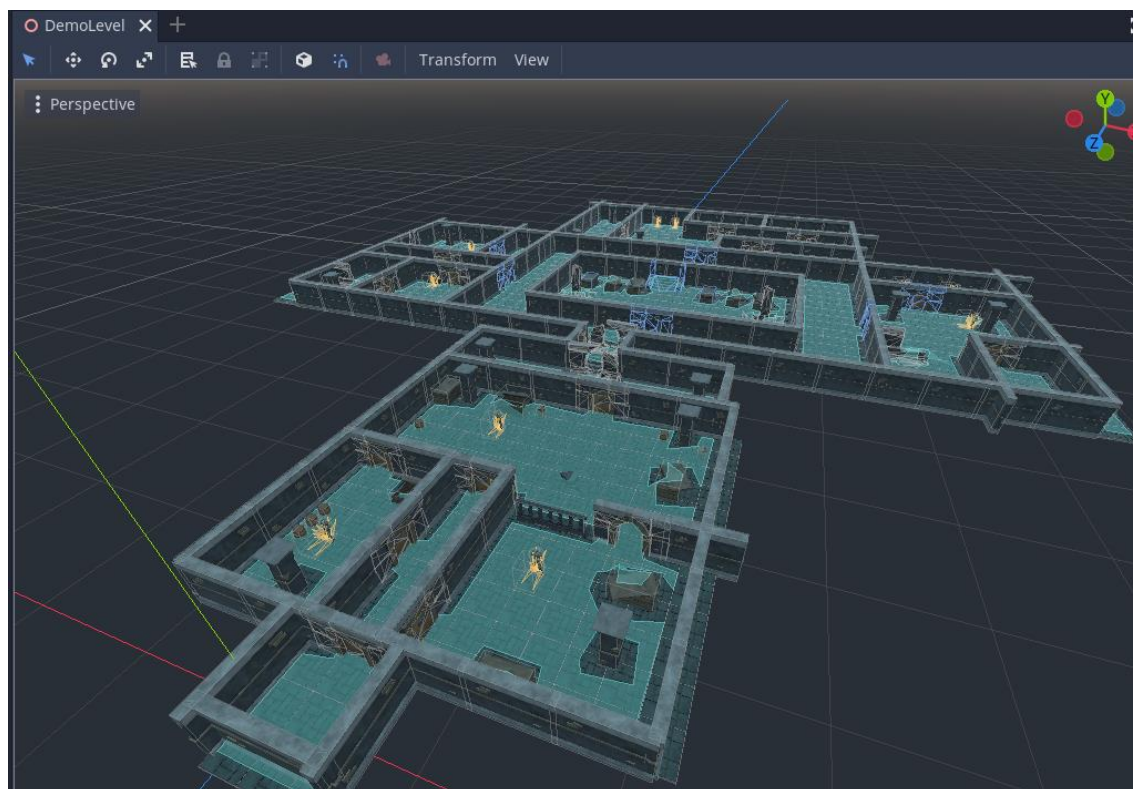
## 4.1 Pelin ominaisuudet

Tietokoneroolipeliprojektia varten toteutettiin kolmiulotteinen ensimmäisen persoonan näkymästä pelattava roolipeli, jossa pelaaja voi reaaliajassa lyödä aseilla, ampua jousella tai heittää loitsuja päihittääkseen viholliset. Hahmon eri ominaisuudet sekä aseiden ja haarniskojen laatu vaikuttavat siihen, kuinka hyvin hahmo pärjää taistelussa, pelaajan refleksien ja sorminäppäryyden lisäksi.

Peliin otettiin vaikutteita etenkin The Elder Scrolls -pelisarjan peleistä ja Dungeons & Dragons -roolipelistä.

Aiheen rajaamiseksi insinööriyössä ei syvennytty itse ensimmäisen persoonan näkymän, liikkuvuuden tai taistelun toteutukseen, sillä roolipelejä varten voi olla monenlaisia taistelujärjestelmiä ja perspektiivejä. Projekti toteutettiin Godot-pelimoottorin versiolla 3.5.1, joka julkaistiin syyskuussa 2022 [75]. Projektin toteutusvaiheessa Godot 4:ää ei ollut vielä julkaistu.

Projektia varten luotiin testikenttä (kuva 11) käyttäen Godot Market Place [76] -sivustolta ostettuja 3D-malleja. Testikentässä on erilaisia arkkuja, tynnyreitä ja hauta-arkkuja, joita pelaaja voi avata, ja ottaa sieltä esineitä itselleen. Kentässä on myös ovia, jotka voidaan avata. Jotkut ovista tai arkuista ovat lukossa, mikä vaatii pelaajahahmolta tiirikointitaitoa.



Kuva 11. Godot-editorin näkymä insinööriyön pelin testikentästä.

Pelaaja liikkuu WASD-näppäimillä ja liikuttaa kameraa hiirellä. Pelaaja voi lyödä, ampuu tai lohtia painamalla hiiren vasenta painiketta, mikäli pelaajahahmolla on jokin ase tai loitsu käytössä (kuva 12). E-painikkeella ja kohdistamalla kamera johonkin tiettyyn esineeseen tai asiaan pelaaja voi olla vuorovaikutuksessa ympäristönsä kanssa, mikäli vuorovaikutus on mahdollista.

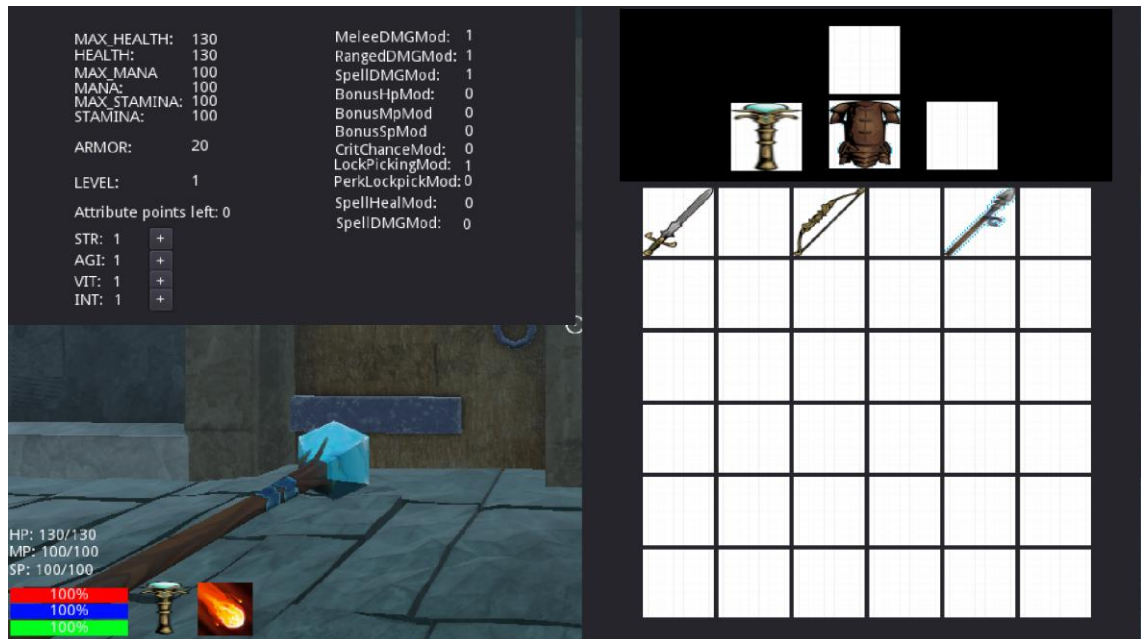


Kuva 12. Taistelu hirviön kanssa miekalla ensimmäisen persoonan näkökulmasta.

Päihittämällä vihollisia hahmo saa kokemuspisteitä, joiden avulla pelaajahahmo saavuttaa uusia tasoja ja saa täten ominaisuuspisteitä, ja lisäetupisteitä, joilla voi kehittää hahmoaan eteenpäin.

## 4.2 Käyttöliittymä

Pelin käyttöliittymä on karkea, mutta tekee tehtävänsä. Tavoitteena oli tehdä toimiva käyttöliittymä, ulkoasu oli toissijaista. Painamalla R-nappia pelaaja saa esille kaikki pelaajahahmon ominaisuudet. Pelaajahahmon pääominaisuudet ovat voimakkuus (STR), ketteryys (AGI), kestävyys (VIT) ja älykkyys (INT), ja ne säätelevät monia eri ns. johdettuja ominaisuuksia. Esimerkiksi ketteryys (AGI) nostaa pelaajahahmon jousivahinkoa (RangedDMGMod), kykyä tehdä kriittistä vahinkoa (CritChanceMod) ja tiirikointikykyä (LockPickingMod). (Kuva 13.)

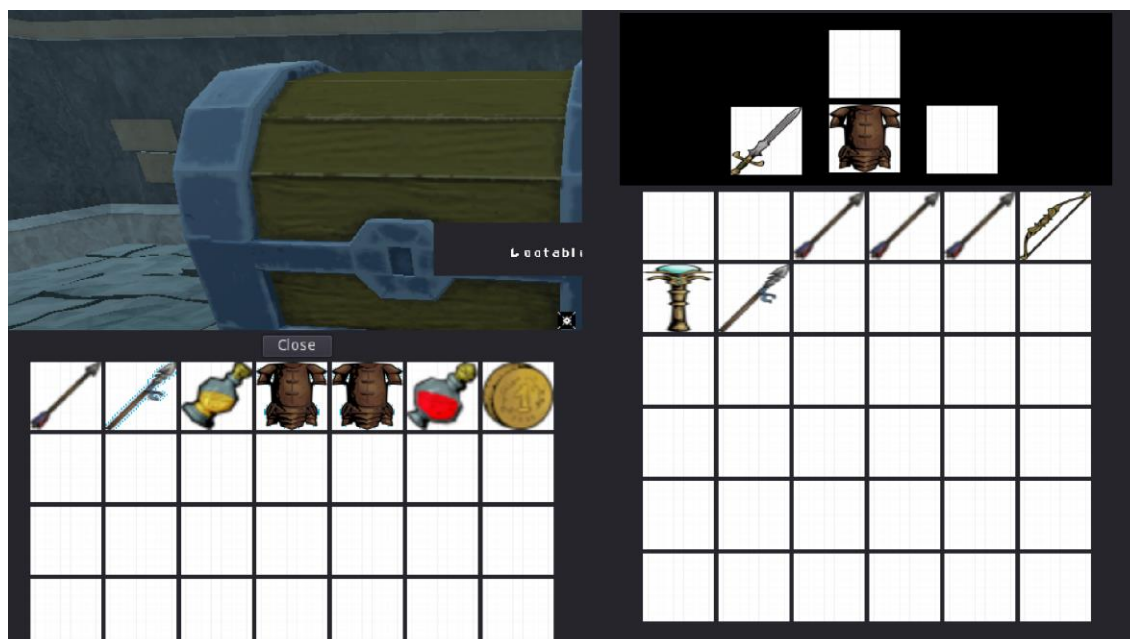


Kuva 13. Pelaajahahmon ominaisuudet, puettut ja kannetut esineet sekä valittu loitsu.

## 4.3 Pelaajahahmon esineet

### 4.3.1 Esineidenhallinta

Peliin toteutettiin raahaa ja pudota -esineidenhallintajärjestelmä. Pelaajan on mahdollista liikuttaa esineitä Godot-pelimoottorin omilla raahaa ja pudota -metodeilla arkkujen, pelaajahahmon repun sekä pelaajahahmon puettujen varusteiden välillä. Jokainen valkoinen ruutu on paikka, johon pelaaja voi sijoittaa esineen. Oikealla ylhäällä on pelaajan puettut esineet tai aseet. (Kuva 14.)



Kuva 14. Esimerkki esineiden hallinnasta arkkua avatessa.

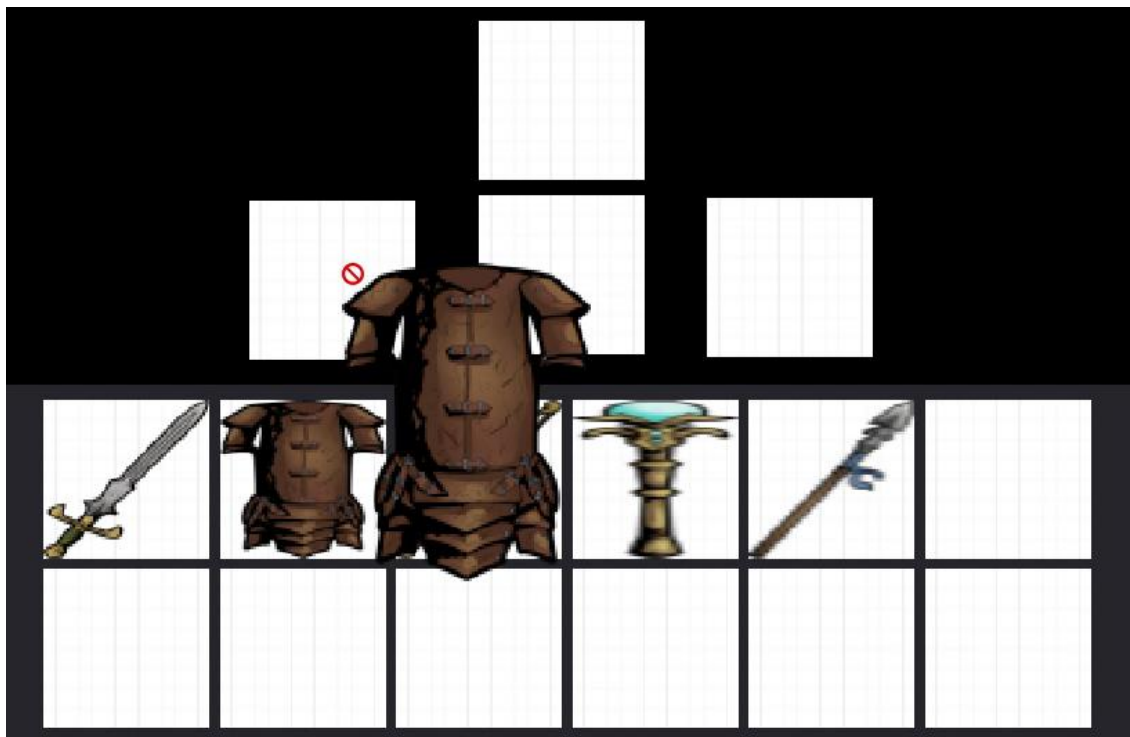
Jokainen esine on solmu, joka sisältää esineskriptin, joka toteuttaa Godot-pelimoottorin oman `get_drag_data`-metodin mahdollistaakseen raahausominaisuuden. Tämän lisäksi jokainen ruutu sisältää esinepaikkaskriptin, joka toteuttaa `can_drop_data`- ja `drop_data`-metodit jotka mahdollistavat esineiden pudottamisen paikkaan sekä tarkistavat, voiko esineen sijoittaa kyseiseen paikkaan.

Raahaa ja pudota -käyttöliittymän toteutus Godot-pelimoottorissa on tuettu mm. kontrollisolmun `can_drop_data`-, `get_drag_data`- ja `drop_data`-metodeilla.

Kaikki käyttöliittymäsolmut periytyvät Kontrolli-luokasta. Kontrolli-luokka mahdollistaa mm. käyttöliittymäsolmujen sijainnin, koon ja marginaalin asettamisen, sekä ankkuroinnin. [77.]

`Get_drag_data`-metodi palauttaa raahattavan solmun dataa kontrollisolmuihin, jotka odottavat pudotettavaa dataa [78]. Tämä metodi sopii pelin esineiden (aseiden, haarniskojen jne.) datan palauttamiseen esinepaikkoihin.

Mikäli pelaajalle halutaan näyttää kursorissa raahattava esine, voi käyttää `set_drag_preview`-metodia [79], mikä helpottaa hahmottamaan esineen siirtämistä (kuva 15).



Kuva 15: Nahkahaarniskan raahaaminen asepaikkaan, mutta `can_drop_data`-metodin stop-merkkigrafiikka indikoi, ettei tämä ole mahdollista.

`Can_drop_data`-metodilla voidaan tarkistaa solmua raahatessa hiirellä, onko kontrollisolmusta periytyvän solmun data sopiva ja voidaanko se pudottaa tiettyyn sijaintiin [80].

Esineidenhallinnan toteuttamisessa `can_drop_data`-metodi sopii hyvin mm. tarkistamaan, onko kyseinen esine sopiva pudotettavaksi tiettyyn haarniska- tai asepaikkaan. Mikäli esineen data ei vastaa varustepaikkaa tai varustepaikka sisältää jo esineen, palautetaan 'epätosi', muuten palautetaan 'tosi', jolloin esineen sijoittaminen onnistuu.

`Drop_data`-metodin avulla Kontrolli-luokasta periytyvään solmuun voidaan vastaanottaa dataa, joka palautetaan `get_drag_data`-metodista. Ennen pudotusta

pelimoottori kutsuu ensin `can_drop_data`-metodin (mikäli solmun skripti toteuttaa tämän metodin) tarkistaakseen, onko pudotus sallittua. [81.]

Kun esinepaikkaan asetetaan esine, poistetaan esinepaikan lapsisolmu (johon esineet sijoitetaan), mikäli sellainen on olemassa, ja esinepaikkasolmuun lisätään uusi solmu, johon sijoitetaan esineen data sekä grafiikka kuvastamaan, että esinepaikassa on tietynlainen esine.

### 4.3.2 Esinetietokanta

Pelaajan käytettävien esineiden ominaisuuksien määrittämiseen ja luontiin tehtiin esinetietokanta-luokka. Esinetietokanta on mahdollista luoda myös käyttämällä JSON- tai SQL-tietokantoja, mutta suhteellisen yksinkertaista roolipeliprojektia varten koettiin kätevämmäksi käyttää tavallista GDScript-luokkaa ja sanakirjatietorakenteita. Projektin esineet määritellään pääosin käyttämällä tekstiä string-muuttujilla, mutta myös kokonaislukuja ja enum-tyyppejä käytetään hyödyksi.

Esimerkiksi kuvassa 16 määritetään pelaajalle puettava haarniska, joten sen esinetyyppi on `ITEM_TYPE.EQUIPPABLE`. Se voidaan sijoittaa pelaajahahmon haarniskapaikkaan (`SLOT_ARMOR`). Sen puolustusarvo (AC) on 20, ja se lisää käyttäjän kokonaiselämäpisteitä kymmenellä (HP). Sen oletusmahdollisuus olla taikaesine on 10 % (`MAGIC_CHANCE`).

```

enum ITEM_TYPE {EQUIPPABLE, CONSUMABLE, GOLD, OTHER}
enum CONSUMABLE_TYPE {INSTA_HEAL, INSTA_MANA, INSTA_STAMINA}

var _rng = RandomNumberGenerator.new()

var _inventory_items = [
  ARMOR : {
    ID : ARMOR,
    NAME : ARMOR,
    TYPE : ITEM_TYPE.EQUIPPABLE,
    ICON : ICON_FILE_PATH + ARMOR + ICON_FILETYPE,
    SLOT : SLOT_ARMOR,
    AC : 20,
    HP : 10,
    MAGIC_CHANCE : 10
  },

```

Kuva 16. Esinetietokannassa haarniskaesineen määrittäminen.

Esinetietokanta-luokassa on metodit esineiden luomiselle ja esinetietojen palauttaminen tietokannasta muille luokille. Luoduilla esineillä on oletusarvot, mutta on mahdollista asettaa vaihtoehtoisilla parametreilla eri taikaesineomaisuusarvo tai yksilöllinen tunniste. Taikaesineomaisuuksia varten on myös oma sanakirjarakenne, jossa on määritetty ominaisuuden nimi ja tyyppi sekä tyyppin arvo. Projektissa on toistaiseksi vain kaksi taikaesineomaisuutta, mutta niitä on mahdollista lisätä. (Kuva 17.)

```

var random_magic_properties = {
  0: {
    NAME : "vigor",
    HP : 20
  },
  1: {
    NAME : "defense",
    AC : 30
  }
}

```

Kuva 17. Taikaesineiden määrittäminen.

Esimerkiksi haarniska, jolla on vigor-ominaisuus, on pelissä nimeltä Armor of Vigor, ja se antaa sen puolustusarvon lisäksi myös 20 pistettä pelaajahahmon kokonaiselämään.

Esineiden taikaominaisuudet ja ominaisuuksien satunnaisuus käydään tarkemmin läpi aliluvussa 4.3.4 Saaliinhallinta.

### 4.3.3 Esine-luokka

Jokaiseen esinenäkymään on asetettu esineskripti. Esineominaisuudet, jotka esiteltiin luvussa 4.3.2, asetetaan siellä. Esineeseen asetetaan kaikki olennaiset ominaisuudet, kuten nimi, tyyppi, onko se taikaesine ja millainen taikaesine sekä mihin varustepaikkaan sen voi pukea, jos se on mahdollista.

Esineiden raahaamisen toteutus käyttöliittymässä on esitelty luvussa 4.3.1 Esineidenhallinta.

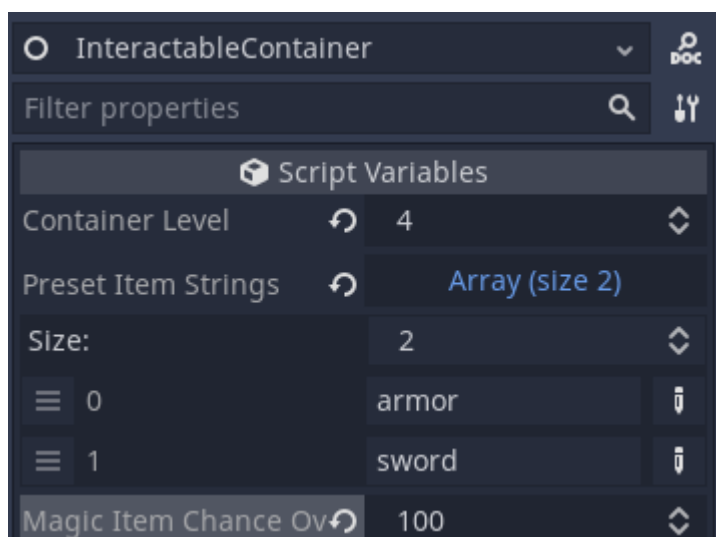
### 4.3.4 Saaliinhallinta

Saaliinhallinta-luokalla määritetään satunnaiset esineet ja arvotaan, ovatko esineet maagisia ja mikä taikaominaisuus esineeseen lisätään. Kuten aliluvussa esiteltiin 4.3.2, eri esineet koostuvat sanakirjarakenteista, joiden avaimet ovat string-tyyppisiä ja esineiden ominaisuudet voidaan palauttaa avaimen avulla. Koko tietokannan kaikki esineet on myös mahdollista palauttaa. Saaliinhallinta-luokassa voidaan arpoa satunnainen esine ottamalla satunnainen esine tietokannasta ja arpoa onko, esine taikaesine. Mikäli esine on taikaesine, voidaan myös palauttaa satunnainen taikaominaisuus tietokannan taikaesineominaisuuksista.

### 4.3.5 Vuorovaikutteiset säiliöt

Pelimaailman arkkujen ja muiden säiliöiden sisältö voidaan asettaa tai muuttaa koodissa editorissa niihin liitetyn VuorovaikutteinenSäiliö-luokan (engl. InteractableContainer) skriptin avulla.

Luokassa on Godot-editorille export-avainsanalla paljastettu muuttujia, joiden avulla on mahdollista määrittää suoraan editorissa, mitä esineitä pelimaailman säiliöllä on varmasti käyttämällä niiden tunnuksia, montako satunnaista esinettä löytyy, ja taikaesinemahdollisuus, mikäli halutaan poiketa oletusarvoista (kuva 18).



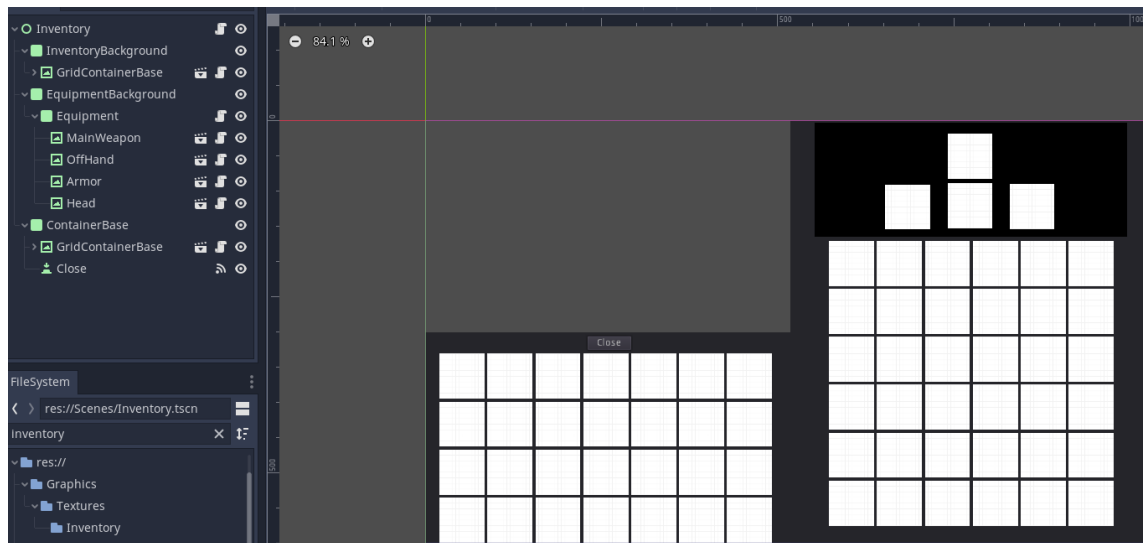
Kuva 18. Editorissa vuorovaikutteisen säiliön ominaisuuksien asettaminen.

Esineet, jotka ovat säiliössä, tallennetaan, kun pelaaja lopettaa säiliön tarkastelun. Esineet yksinkertaisesti sijoitetaan taulukkorakenteeseen ja taulukkorakenteen sisältö palautetaan, kun pelaaja seuraavan kerran avaa kyseisen säiliön.

Itse esineet luodaan Inventaario-luokassa (4.3.6 Inventaario ja ruudukkosäiliö). Tässä luokassa määritetään säiliöiden alustavien esineiden tunnuksia ja tallennetaan esineet, jotka on luotu.

### 4.3.6 Inventaario ja ruudukkosäiliö

Inventaario-luokassa käsitellään esineiden luonti ja hallinnoidaan pelaajahahmon reppua eli omaa esinesäiliötä sekä näytetään maailmassa avattavan säiliön sisältö ja luodaan siihen esineet. Inventaario-näkymä sisältää vasemmalla avattavan säiliön sisällön, joka on näkyvillä, kun pelaaja on avannut jonkun säiliön, ja pelaajahahmon repun, sekä puetut esineet oikealla (kuva 19).



Kuva 19. Inventaario-näkymän rakenne editorissa.

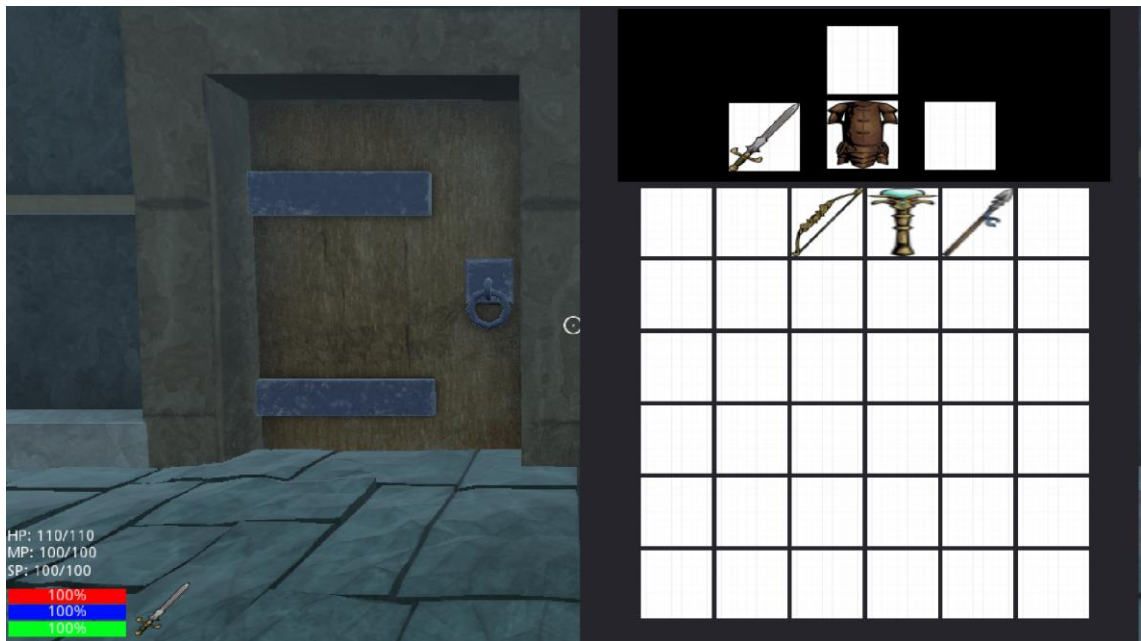
Luokassa luodaan esineet säiliöissä määriteltyjen esinetunnusten avulla. Luo- tuja esineitä varten luodaan ilmentymä InventoryItem-näkymästä (4.3.3 Esine- luokka) ja sille asetetaan tekstuuri sekä muut vaaditut parametrit.

Pelaajan aloitusesineet luodaan ja lisätään suoraan pelaajahahmon reppuun, ja säiliöiden esineet luodaan ja lisätään sillä hetkellä, kun pelaaja avaa säiliön. Esineet asetetaan reppuun tai avattuun säiliöön paikka kerrallaan vasemmalta oikealle, ylhäältä alas ja lisätään ensimmäiseen paikkaan, joka on tyhjä.

Pelaajahahmon reppu ja avattava säiliö sisältävät Ruudukkosäiliö-luokan skrip- tin. Se huolehtii esineiden poistamisesta säiliöstä sekä säiliön esineiden palau- tuksesta niiden tallentamista varten.

### 4.3.7 Varustepaikat

Puetut esineet ja niiden vaikutus pelaajahahmon ominaisuuksiin käsitellään Varustepaikat-luokassa. Kun puettava esine, kuten haarniska tai ase, puetaan, lisätään sen antamat mahdolliset elämäpiste- ja/tai puolustuskykybonukset pelaajahahmolle ja tämän jälkeen päivitetään käyttöliittymä kuvaamaan muutoksia. Mikäli pelaaja pukee tai poistaa aseensa, sen kuvake päivitetään myös käyttöliittymässä (kuva 20).

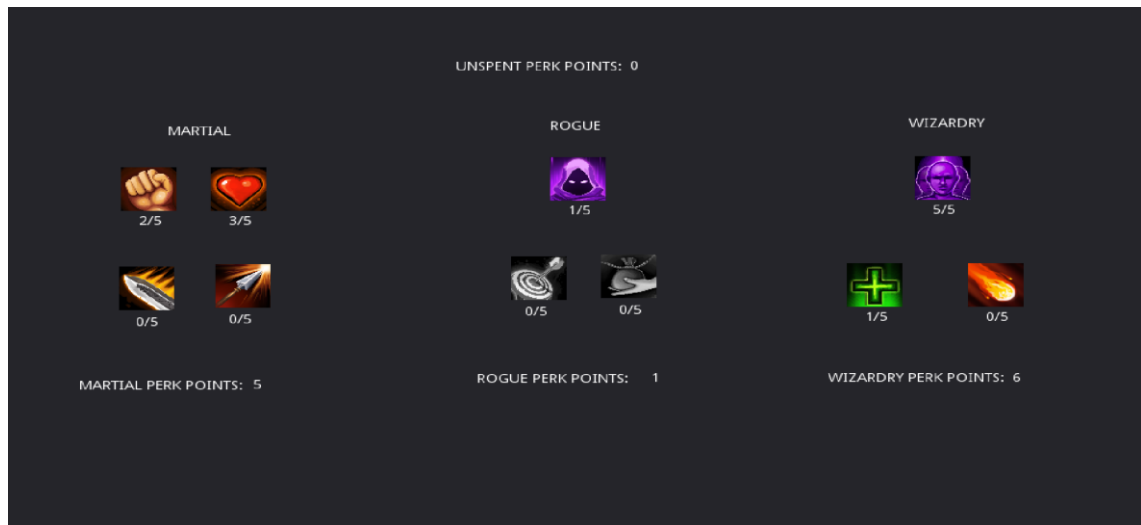


Kuva 20. Puetun aseensa kuvake näytetään käyttöliittymässä vasemmassa alakulmassa.

## 4.4 Lisäedut

Kun hahmo saavuttaa uuden tason, hän saa myös yhden lisäetupisteen (perk point) käytettäväksi. Projektissa on 10 eri lisäetua jaettuna kolmeen eri kategoriaan, taistelijatyypin (engl. Martial), roistotyypin (engl. Rogue) ja velhotyyppiin (engl. Wizardry). Jokaisessa lisäetukategoriassa on kaksi riviä ja 1–2 lisäetua riviä kohti. Saavuttaakseen alemman rivin lisäedut hahmon on ensin käytettävä 5 pistettä ylemmän rivin lisäetuihin. Mikäli ylemmän riviin ei ole käytetty

tarpeeksi lisäetupisteitä, rivin kuvakkeet näytetään harmaana, mikä indikoi, ettei niihin voi laittaa pisteitä (kuva 21).



Kuva 21. Lisäetuikkuna pelitilassa. Eri lisäetukategorioihin ja lisäetuihin on li-sätty eri määriä pisteitä osoittamaan toiminnallisuus.

Jokainen lisäetu kehittää hahmoa eteenpäin, kuten antaa lisää ominaisuuspis-teitä, parantaa tietyn asetyypin tai loitsun tehoa tai antaa lisäetua tiirikoinnissa. Editorissa jokaiseen lisäetusolmuun asetettiin LisäEtuPainike-skripti, joka to-teuttaa lisäetupisteen lisäykseen tarvittavan logiikan sekä grafiikan päivityksen.

Uusien lisäetujen lisääminen pyrittiin tekemään helpoksi. Paljastamalla LisäEtu-Painike-skriptissä lisäedun yksilöivä tunnus ('perk\_id'), sen vaatimukset (requi-red\_perk\_points) sekä sallitu maksimiarvo (max\_points) export-avainsanaa käyttäen voidaan kätevästi asettaa suoraan Godot-editorissa eri lisäetujen yksi-lölliset vaatimukset ja maksimiarvot (kuva 22).

```
export var max_points : int
export var required_perk_points = 0
export var perk_id : int
```

Kuva 22. Godot-editoriin paljastetut muuttujat PerkButton-skriptissä.

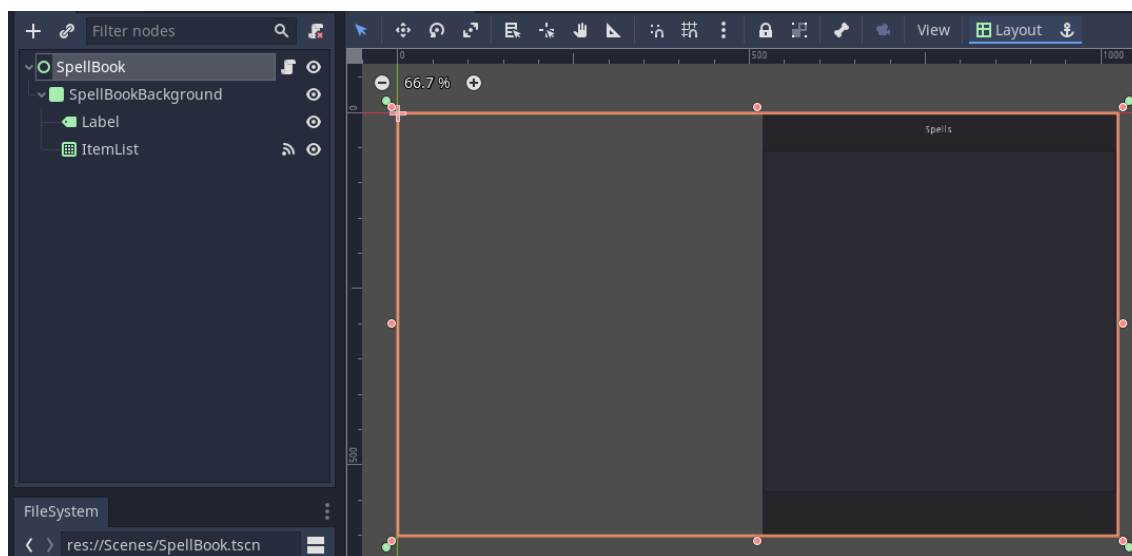
## 4.5 Loitsujärjestelmä

Pelissä on yksinkertainen loitsujärjestelmä, jolla voidaan luoda vahingoittavia tai parantavia loitsuja erilaisin parametrein. Pelaajahahmo voi heittää loitsun, mikäli hänellä on sauva puettuna ja tarpeeksi manapisteitä.

### 4.5.1 Loitsukirja

Pelaajan on mahdollista avata loitsukirja, jossa näkyy lista loitsuista, joita pelaajahahmo osaa. Tätä toteutusta varten pelaajan loitsut ovat ennalta määrättyjä, mutta potentiaalisesti pelaajahahmo voisi oppia ne esim. loitsukirjoista, joita voisi ostaa tai löytää pelimaailmassa.

Lista loitsuista luotiin käyttämällä Godot-pelimoottorin omaa ItemList-luokkaa. ItemList on luokka, joka periytyy kontrollisolmusta, ja sen avulla voidaan luoda valittavia esineitä [85]. Loitsukirja-näkymä koostuu yksinkertaisista grafiikoista, loitsukirjaskriptistä ja Godot-pelimoottorin ItemList-luokan solmusta (kuva 23).



Kuva 23. Loitsukirja-näkymän (SpellBook) rakenne ja näkymä editorissa.

Loitsukirjaskriptissä on logiikka loitsulistan luomiseen, avaamiseen ja piilottamiseen käyttäjältä. Kun loitsulista avataan, palautetaan pelaajahahmon osaamat

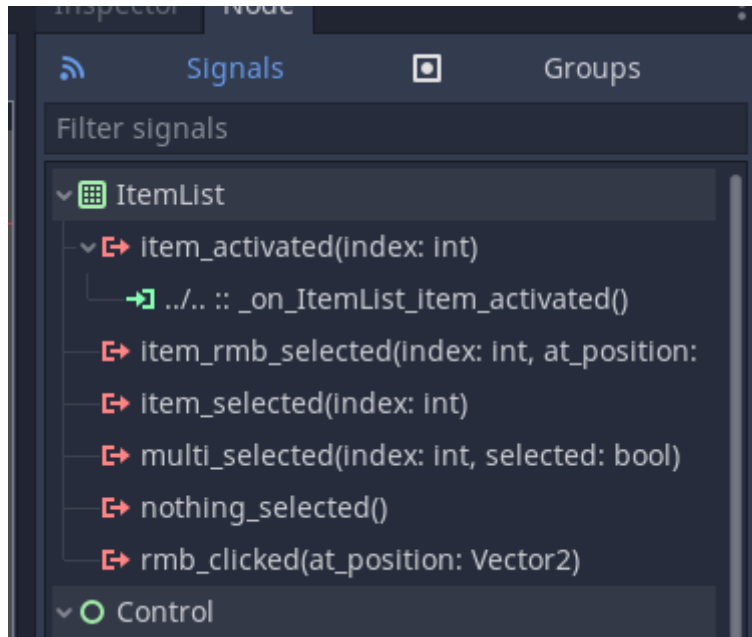
loitsut Loitsunhallinta-luokan ilmentymän avulla (4.5.2 Loitsu-luokka ja loitsunhallinta).

Loitsulista luodaan käymällä kaikki pelaajahahmon osaamat loitsut läpi ja lisäämällä loitsun nimen perusteella kuvake ja listaesineen nimi. Lisäksi loitsun resurssin hinta, mana, lisätään listaesineen nimeen auttamaan loitsun valintaa (kuva 24).



Kuva 24. Loitsukirja avattuna.

Kun loitsukirja suljetaan, poistetaan loitsulistan esineet ja piilotetaan loitsukirja käyttäjältä. Loitsun valintaan käytetään signaalia `on_ItemList_item_activated()`, joka näkyy Godot-pelimoottorin editorissa (kuva 25). Godot-pelimoottori määrittää `ItemList`-luokan esineen aktivoituneeksi, mikäli käyttäjä kaksoisklikkaa esinettä tai painaa Enter-näppäintä [85].



Kuva 25. ItemList-solmun item\_activated-signaali loitsukirjan esineen aktivointiin.

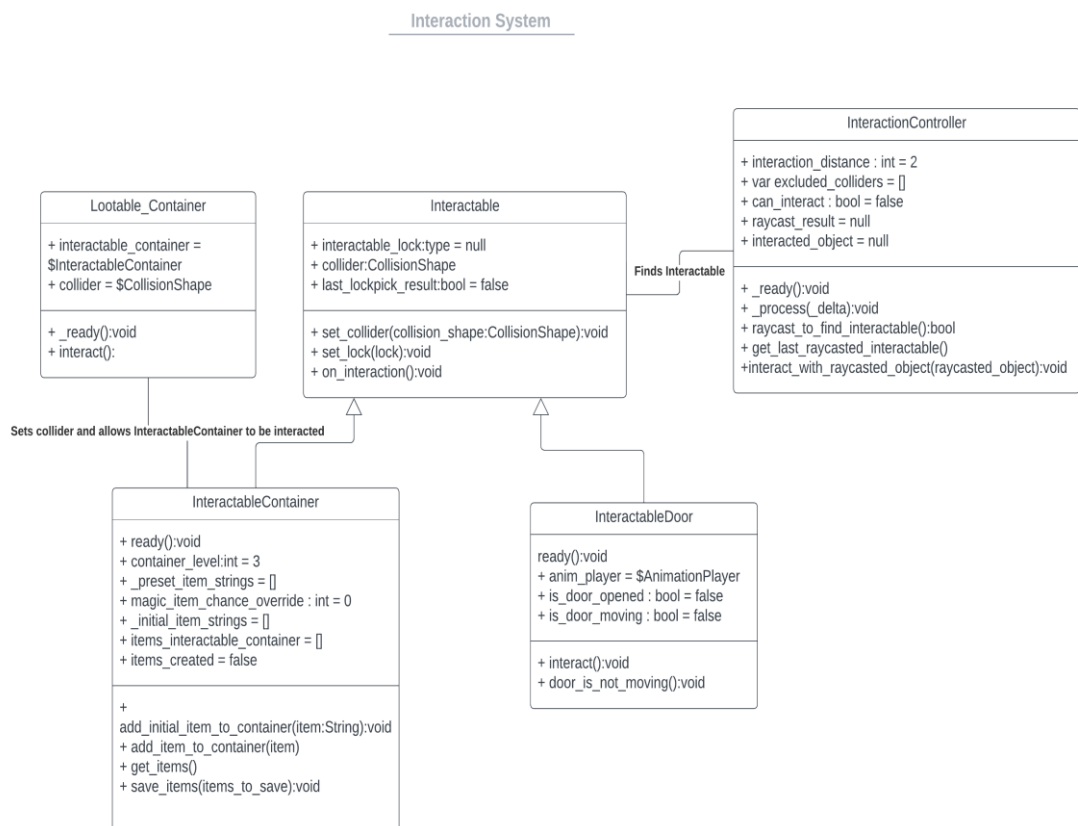
#### 4.5.2 Loitsu-luokka ja loitsunhallinta

Loitsu-luokassa määritetään loitsun parametrit: sen nimi, voimakkuus, manaresurssin hinta, säde, jos kyseessä on alueeseen kohdistuva loitsu jne. Tämän lisäksi loitsun eri kantamat tai etäisyydet sekä vaikuttavat kohteet ovat määriteltävissä samoin, loitsun tyyppi ('parantava' tai 'vahingoittava').

Loitsunhallinnassa määritetään pelaajan tuntemat loitsut, toteutetaan loitsunheittologiikka ja luodaan ne pelimaailmaan. Loitsunhallinnassa toteutetaan tarvittava logiikka erilaisiin loitsuihin riippuen loitsun etäisyydestä ja tyypistä. Esimerkiksi mikäli loitsu on vahingoittava alueloitsu, sen etäisyystyyppi on määriteltävä alueeksi. Mikäli loitsu on vahingoittava, ladataan sille sopiva loitsun näkymä (Räjähdyksenäkymä). Tämän jälkeen tästä näkymästä luodaan instanssi ja instanssi lisätään pelimaailmaan. Pelaajahahmon älykkyyden tuoma lisäloitsu-vahinko otetaan myös huomioon. Lisäksi mikäli loitsu on asetettu olemaan välittämättä pelaajasta, sen törmäytintä (engl. collider) ei oteta huomioon.

## 4.6 Pelimaailmaan vuorovaikuttaminen

Projektin kolmiulotteiseen pelimaailmaan voidaan vaikuttaa luodun vuorovaikutusjärjestelmän avulla. Vuorovaikutuksenohjaus-luokka (engl. InteractionController) mahdollistaa vuorovaikutteisten esineiden löytämisen pelimaailmasta. Vuorovaikutteinen-luokka (engl. Interactable) mahdollistaa vuorovaikutteisten esineiden perustoiminnallisuuden ja siitä perityt VuorovaikutteinenSäiliö- (engl. InteractableContainer) ja VuorovaikutteinenOvi-luokat (engl. InteractableDoor). Avattavat säiliöt -luokka (engl. Lootable Container) puolestaan mahdollistaa vuorovaikutteisen säiliön vuorovaikutuksen. (Kuva 26.)

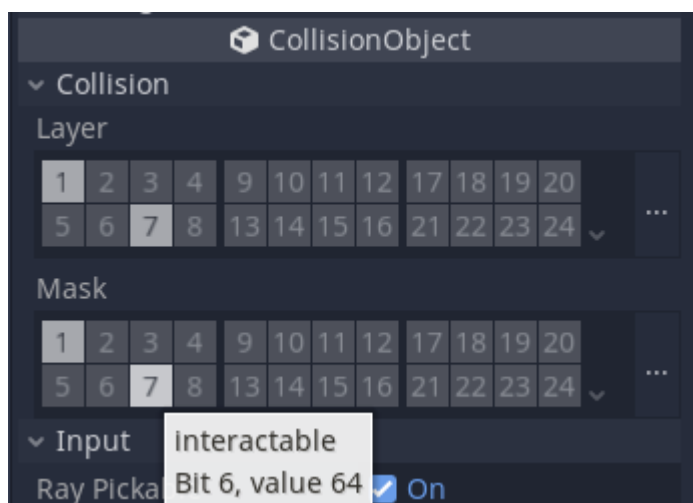


Kuva 26. Vuorovaikutusjärjestelmän rakenne.

### 4.6.1 Vuorovaikutuksen ohjaus

Peliin luotu Vuorovaikutuksenohjaus-luokka mahdollistaa vuorovaikutteisten esineiden löytämisen pelimaailmasta ja niiden käyttömetodien kutsumisen. Koska peliä pelataan ensimmäisestä persoonasta, joka ruudunpäivityksessä käytetään Godot-pelimoottorin omia `get_world-`, `get_direct_space_state-` ja `intersect_ray-` metodeja [82] heittämään säteen pelaajan ruudun keskeltä. Näiden metodien avulla saadaan määritellyn vuorovaikutusetäisyyden päästä lähin törmäytin, jolla on Vuorovaikutteinen-luokan vuorovaikutus-metodi.

Pelissä asetettiin bitti 6 eli arvo 64 vuorovaikutteisia törmäyttimiä varten. Täten ainoastaan pelimaailman esineet, jotka on asetettu Vuorovaikutteinen-tasolle otetaan huomioon säteenheitossa (kuva 27).



Kuva 27. Vuorovaikutteinen pelimaailman törmäytin, joka halutaan määrittää vuorovaikutteiseksi.

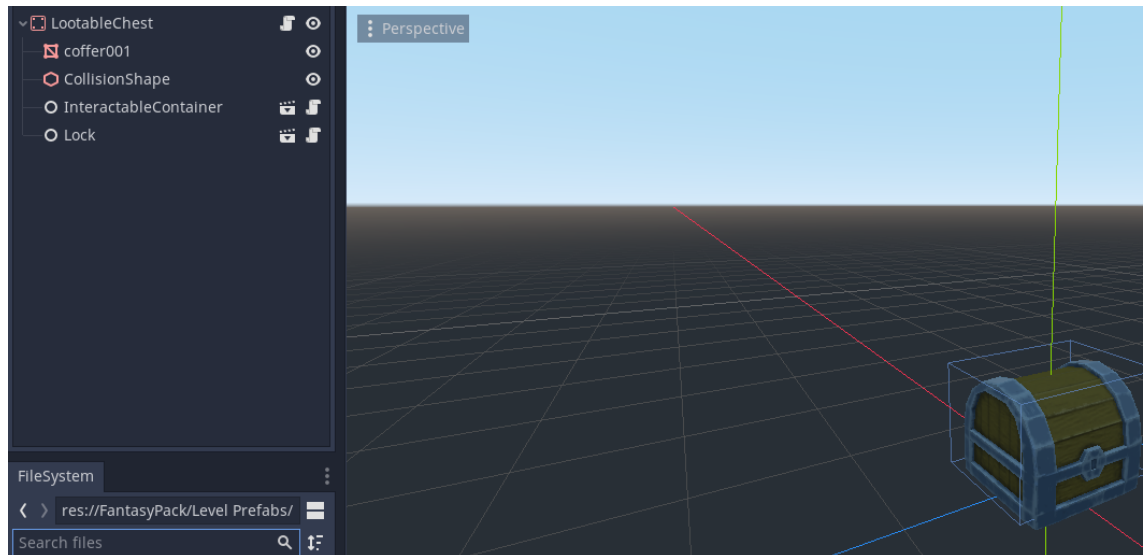
Mikäli pelaaja on vuorovaikutusetäisyydellä ja pelaajan kamera on kohdistettu vuorovaikutteiseen esineeseen pelimaailmassa, vuorovaikutus on mahdollista (kuva 28).



Kuva 28. Pelaajahahmo on avattavan tynnyrin vuorovaikutusetäisyydellä.

#### 4.6.2 Vuorovaikutteinen-luokka ja vuorovaikutteisten näkymien rakenne

Projektin jokainen kolmiulotteinen esine tai hahmo, jolle halutaan jokin tietty interaktio, joka ei ole taistelu (käyttäminen, avaaminen ja tiirikointi), sisältää solmun, johon on liitetty jokin Vuorovaikutteinen-luokasta perivä skripti. Esimerkiksi projektissa avattava arkku, `LootableChest`, sisältää solmun `InteractableContainer`, johon on liitetty `VuorovaikutteinenArkku`-skripti, joka periytyy `Vuorovaikutteinen-luokasta` (kuva 29).



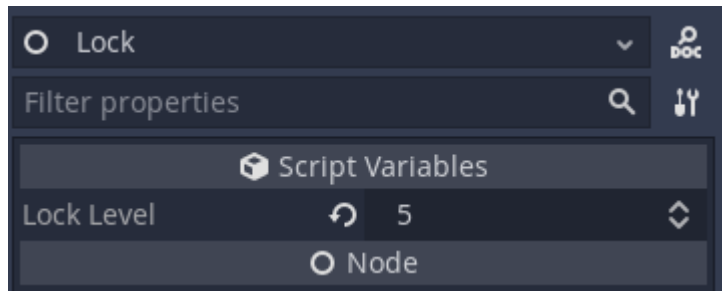
Kuva 29. Esimerkki vuorovaikutuksen toteutuksesta LootableChest-näkymässä.

Vuorovaikutteinen-skriptiä itsessään ei ole liitetty mihinkään projektin näkymään, mutta siitä periytyvät skriptit VuorovaikutteinenSäiliö (ks. luku Arkut/InteractableContainer) ja VuorovaikutteinenOvi (ks. luku Ovet/InteractableDoor) on liitetty useisiin esineisiin pelimaailmassa.

Vuorovaikutteinen-luokka sisältää tiedon siihen asetetusta CollisionShape-luokan solmusta. CollisionShape-luokan solmu mahdollistaa kolmiulotteisen muodon törmäyksen esittämisen. [83.]

### 4.6.3 Lukot

Pelin jokaisella arkulla tai ovella on lukko. Oletuksena lukon taso on 0, mutta export-avainsanalla skriptin lukon taso -muuttujan (engl. Lock Level) arvoa voidaan muokata (kuva 30).



Kuva 30. Skriptin muuttujien paljastaminen editorille.

Pelaajahahmon kyky onnistua tiirikoinnista riippuu sen Agility (AGI) -ominaisuudesta ja ostetuista lisäeduista, jotka parantavat tiirikointikykyä. Tiirikoinnin yhteydessä toteutetaan niin sanottu virtuaalinen nopanheitto, jossa tiirikoinnin tulos on välillä 0 ja pelaajahahmon Agility-ominaisuus, johon lisätään lisätujen antama bonus. Mitä suurempi luku nopanheitosta, sen parempi mahdollisuus onnistua tiirikoinnissa. Mikäli pelaajahahmon on mahdotonta onnistua tiirikoinnissa, vaikka saisi tiirikoinnissa täydellisen heiton, nopan heittoa ei tehdä ja täten pelaajahahmon ei ole mahdollista tiirikoida ovea tällä hetkellä (kuva 31).



Kuva 31. Esimerkki mahdottomasta oven tiirikointiyrityksestä.

Mikäli tiirikointi on mahdollista, mutta nopanheitosta tuli luku, joka on alempi kuin lukon taso, käyttäjälle ilmoitetaan, että yritys epäonnistui mutta on mahdollista yrittää uudestaan. Jos nopanheitto ylittää lukon tason tai vastaa sitä, yritys onnistui ja käyttäjälle ilmoitetaan onnistuminen.

#### 4.6.4 Ovet ja säiliöt

Ovi-luokan skripti `VuorovaikutteinenOvi` periytyy myös `Vuorovaikutteinen-luokasta`. Se sisältää `Vuorovaikutteinen-luokan` perustoiminnallisuuden lisäksi myös logiikan ovien avaamiseen. Ovien liikkeen animointiin käytetään Godot-pelimoottorin `AnimationPlayer`-solmua.

Vuorovaikutusten osalta arkuilla, tynnyreillä ja vihollisten ruumilla (kollektiivisesti säiliöt) on sama perustoiminnallisuus kuin ovilla: katsotaan, onko säiliö lukossa, yritetään tiirikoida se, mikäli mahdollista, jos tiirikointi onnistui tai ei ole lukossa, säiliö voidaan avata. Säiliöiden muu toiminnallisuus on käyty tarkemmin läpi luvussa 4.3.5 `Vuorovaikutteiset säiliöt`.

### 4.7 Lopputuloksen arviointi

Insinööriyössä luodun pelin avulla selvisi, että Godot-pelimoottorissa on mahdollista luoda ainakin pienen mittakaavan tietokoneroolipeli. Yksinkertaisen käyttöliittymän lisäksi myös raahaa ja pudota -käyttöliittymän toteutus koettiin varsin helpoksi käyttämällä Godot-pelimoottorin editoria ja pelimoottorin omia metodeja.

Vaikka insinööriyön tavoitteena ei ollut tutkia 3D-pelin luomista pelimoottorilla, ainakin alhaisen polygonimäärän 3D-projekti pyöri suhteellisen hyvin käyttämättä optimointitekniikoita, kuten esimerkiksi okklusion harventamista [84].

Pelimoottorin oman ohjelmointikielen, GDScriptin käyttö todettiin varsin intuitiiviseksi, eikä sen siirtymiseen Unityssä käytetystä C#-ohjelmointikielestä ollut

suuri harppaus. Pelimoottorin hyvä dokumentaatio helpotti GDScript-kielen oppimista. Kuitenkin projektin laajetessa vähemmän dynaaminen kieli olisi voinut helpottaa projektin toteutusta. GDScriptissä kaikki muuttujat ovat julkisia, eivätkä muuttujat ole vahvasti tyyditettyjä, ellei sitä erikseen määritä. Godot kuitenkin tukee C#-ohjelmointikieltä virallisesti ja sen suorituskyky on mitattu paremmaksi kuin GDScriptin [85], mikä tekee siitä sopivamman kielen suurempiin peliprojekteihin. C++-ohjelmointikielen käyttö on myös mahdollista. Lisäksi pelimoottorin oma kehitysympäristö koettiin pätevänä, mutta koodin kirjoitukseen kolmannen osapuolen tekstieditorit epävirallisilla GDScript-lisäosilla havaittiin paremmaksi vaihtoehdoksi.

Godot-pelimoottorin neljännen version suorituskykyparannukset ja Vulkan-grafiikkarajapinnan käyttöönotto voivat parantaa suurempien tietokoneroolipelien luomista.

## 5 Yhteenveto

Tietokoneroolipelit ovat tietokoneella pelattavia roolipelejä, jotka tyypillisesti keskittyvät syvälliseen hahmonkehitykseen, taisteluun ja tarinaan. Niillä on myös monia alalajeja, ja niillä on rikas historia 1970-luvulta lähtien. Tietokoneroolipelit ovat edelleen suosittuja.

Insinööriyössä toteutettiin pieni tietokoneroolipeli, jotta voitiin selvittää, soveltuisiko Godot-pelimoottori sellaiseen tekemiseen. Godot-pelimoottori on avoimen lähdekoodin pelimoottori, joka on varteenotettava vaihtoehto muiden tunnetuimpien pelimoottorien kuten Unityn ja Unrealin rinnalla. Godot ja sen oma GDScript-ohjelmointikieli koettiin helposti opittaviksi.

Insinööriyön peliprojektin avulla todettiin, että Godot-pelimoottorissa on mahdollista tehdä ainakin pienen mittakaavan tietokoneroolipeli-projekti, joka sisältää tietokoneroolipeleille tyypillisiä ominaisuuksia, kuten inventaario, hahmonkehitys, loitsujen heitto sekä pelimaailmaan vuorovaikuttaminen pelaajahahmon

ominaisuuksia hyödyntäen. Pelimoottorin omat luokat ja metodit havaittiin hyödyllisiksi, ja etenkin pelimoottorin oma raahaa ja pudota -käyttöliittymän toteutus koettiin helppokäyttöiseksi. GDScript-ohjelmointikieli havaittiin projektin laajentuessa epäsopivaksi, mutta pelimoottori tukee myös tietokoneroolipeihin sopivampia ohjelmointikieliä, kuten C# ja C++.

Huolimatta siitä, että Godot-pelimoottori on täysin ilmainen ja kehitetty pääosin joukkorahoituksen ja vapaaehtoisten voimin, se on pätevä vaihtoehto isompien pelimoottorien rinnalla. Pelimoottorin kasvu on ollut viime vuosina huomattava, ja sen on ennustettu ohittavan Unity-pelimoottori suosiossa vuoteen 2028 mennessä, joten pelimoottorin tulevaisuus näyttää valoisalta.

## Lähteet

- 1 Verschelde, Rémi. 2023. Maintenance release: Godot 3.5.2. Verkkoaineisto. Godot. <<https://godotengine.org/article/maintenance-release-godot-3-5-2>>. Päivitetty 8.3.2023. Luettu 6.5.2023.
- 2 Licence. Verkkoaineisto. Godot. <<https://godotengine.org/license>>. Luettu 21.5.2023.
- 3 Dawe, Liam. 2014. Godot Game Engine Is Now Open Source. Verkkoaineisto. Gaming on Linux. <<https://www.gamingonlinux.com/articles/godot-game-engine-is-now-open-source.3096/>>. Päivitetty 14.2.2014. Luettu 6.5.2023.
- 4 Linietsky, Juan. 2016. Godot Engine reaches 2.0 stable. Verkkoaineisto. Godot. <<https://godotengine.org/article/godot-engine-reaches-2-0-stable/>>. Päivitetty 23.2.2016. Luettu 21.5.2023.
- 5 Linietsky, Juan. 2018. Godot 3.0 is out and ready for the big leagues. Verkkoaineisto. Godot. <<https://godotengine.org/article/godot-3-0-released>>. Päivitetty 29.1.2018. Luettu 21.5.2023.
- 6 Godot 4.0 sets sail: All aboard for new horizons. 2023. Verkkoaineisto. Godot. <<https://godotengine.org/article/godot-4-0-sets-sail/>>. Päivitetty 1.3.2023. Luettu 21.5.2023.
- 7 Godot Contributors. Verkkoaineisto. Patreon. <<https://www.patreon.com/godotengine>>. Luettu 6.5.2023.
- 8 Make a recurrent donation. Verkkoaineisto. Godot. <<https://godotengine.org/donate/>>. Luettu 6.5.2023.
- 9 Verschelde, Rémi. 2019. Godot Engine awarded \$50,000 by Mozilla Open Source Support program. Verkkoaineisto. Godot. <<https://godotengine.org/article/godot-engine-awarded-50000-mozilla-open-source-support-program/>>. Päivitetty 11.4.2019. Luettu 6.5.2023.
- 10 Linietsky, Juan. 2020. Godot Engine awarded an Epic MegaGrant. Verkkoaineisto. Godot. <<https://godotengine.org/article/godot-engine-was-awarded-epic-megagrant/>>. Päivitetty 3.2.2020. Luettu 6.5.2023.

- 11 Godot as an Open Ecosystem – Juan Linietsky (GDC 2023). 2023. Verkkoaineisto. Youtube. <<https://youtu.be/XduuHNOGGqI?t=479>>. Päivitetty 31.3.2023. Katsottu 6.5.2023.
- 12 Scripting languages. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/stable/getting\\_started/step\\_by\\_step/scripting\\_languages.html](https://docs.godotengine.org/en/stable/getting_started/step_by_step/scripting_languages.html)>. Luettu 6.5.2023
- 13 GDList. Verkkoaineisto. Github. <<https://github.com/Mercerenies/gdlisp>>. Luettu 6.5.2023
- 14 Godot Lua PluginScript. 2023. Verkkoaineisto. Github. <<https://gilzoide.github.io/godot-lua-pluginscript/topics/README.md.html>>. Päivitetty 4.1.2023. Luettu 6.5.2023.
- 15 Linietsky, Juan. 2022. Godot 4.0 will discontinue VisualScript. Verkkoaineisto. <<https://godotengine.org/article/godot-4-will-discontinue-visual-scripting/>>. Päivitetty 23.8.2022. Luettu 6.5.2023.
- 16 List of Features. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/stable/about/list\\_of\\_features.html](https://docs.godotengine.org/en/stable/about/list_of_features.html)>. Luettu 21.5.2023.
- 17 Linietsky, Juan. 2022. Godot and consoles, all you need to know. Verkkoaineisto. Godot. <<https://godotengine.org/article/godot-consoles-all-you-need-know/>>. Päivitetty 15.7.2022. Luettu 6.5.2023.
- 18 Node. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/3.5/classes/class\\_node.html](https://docs.godotengine.org/en/3.5/classes/class_node.html)>. Luettu 6.5.2023.
- 19 Overview of Godot's key concepts. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/3.5/getting\\_started/introduction/key\\_concepts\\_overview.html#doc-key-concepts-overview](https://docs.godotengine.org/en/3.5/getting_started/introduction/key_concepts_overview.html#doc-key-concepts-overview)>. Luettu 6.5.2023.
- 20 Prefabs. 2018. Verkkoaineisto. Unity Documentation. <<https://docs.unity3d.com/Manual/Prefabs.html>>. Päivitetty 31.7.2018. Luettu 6.5.2023.
- 21 Scenes. Verkkoaineisto. Unity Documentation. <<https://docs.unity3d.com/Manual/CreatingScenes.html>>. Luettu 6.5.2023.
- 22 Design Patterns – Observer Pattern. Verkkoaineisto. Tutorialspoint <[https://www.tutorialspoint.com/design\\_pattern/observer\\_pattern.htm](https://www.tutorialspoint.com/design_pattern/observer_pattern.htm)>. Luettu 6.5.2023.

- 23 Using signals. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/3.5/getting\\_started/step\\_by\\_step/signals.html](https://docs.godotengine.org/en/3.5/getting_started/step_by_step/signals.html)>. Luettu 6.5.2023.
- 24 Scripting languages. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/3.5/getting\\_started/step\\_by\\_step/scripting\\_languages.html](https://docs.godotengine.org/en/3.5/getting_started/step_by_step/scripting_languages.html)>. Luettu 6.5.2023.
- 25 Dennis, Patrick. 2023. 50 Best Action RPGs To Play In 2023. Gameranx. <<https://gameranx.com/features/id/219707/article/40-best-action-rpgs-to-play-in-2021/>>. Päivitetty 17.5.2023. Luettu 21.5.2023.
- 26 The Elder Scrolls V: Skyrim. Verkkoaineisto. The Elder Scrolls. <<https://elderscrolls.bethesda.net/en/skyrim/>>. Luettu 6.5.2023.
- 27 Brewer, Nathan. 2016. Going Rogue. A Brief History of the Computerized Dungeon Crawl. Verkkoaineisto. InSight. <<https://insight.ieeeusa.org/articles/going-rogue-a-brief-history-of-the-computerized-dungeon-crawl/>>. Päivitetty 7.7.2016. Luettu 21.5.2023.
- 28 Derboo, Sam. 2014. Wizardry (Series Introduction). Verkkoaineisto. HG101. <<http://www.hardcoregaming101.net/wizardry-series-introduction/>>. Päivitetty 13.10.2014. Luettu 21.5.2023.
- 29 Legend of Grimrock. Verkkoaineisto. Grimrock. <<http://www.grimrock.net/games/>>. Luettu 21.5.2023.
- 30 Barton, Matt. 2007. The History of Computer Role-Playing Games Part 1: The Early Years (1980-1983). Verkkoaineisto. Game Developer. <<https://www.gamedeveloper.com/design/the-history-of-computer-role-playing-games-part-1-the-early-years-1980-1983->>. Päivitetty 23.2.2007. Luettu 6.5.2023.
- 31 Diablo. 2022. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 32 Beneath Apple Manor. 2022. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 33 Ballestrino, David. 2022. Wizardry: Proving Grounds of the Mad Overlord. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.

- 34 Mitchell, Richard. 2022. Ultima. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 35 Bailey, Robert. 2022. The Bard's Tale. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 36 Clerc, Charles. 2022. Might and Magic: Book I – Secret of the Inner Sanctum. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 37 Avellone, Chris. 2022. Wasteland. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 38 Barnson, Jay. 2022. Ultima Underworld: The Stygian Abyss. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 39 Parker, Wilhelm. 2019. Celebrate 25 Years of The Elder Scrolls. Verkkoaineisto. Bethesda. <<https://bethesda.net/en/article/zTLJdDWzVy0Ak7SMatqNG/celebrate-25-years-of-the-elder-scrolls>>. Päivitetty 25.3.2019. Luettu 6.5.2023.
- 40 Weller, Vince D. 2022. The Elder Scrolls II: Daggerfall. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 41 Pepe, Felipe. 2022. Fallout. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 42 Magalhães, Luis. 2022. Baldur's Gate. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 43 Ng, Suzie. 2022. Baldur's Gate II: Shadows of Amn. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 44 Planescape: Torment. Verkkoaineisto. IGDB. <<https://www.igdb.com/games/planescape-torment>>. Luettu 21.5.2023.

- 45 Icewind Dale. Verkkoaineisto. IGDB <<https://www.igdb.com/games/icewind-dale>>. Luettu 21.5.2023.
- 46 Icewind Dale. Verkkoaineisto. IGDB. <<https://www.igdb.com/games/icewind-dale-ii>>. Luettu 21.5.2023.
- 47 Inderwildi, Andreas. 2022. Deus Ex. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 48 Neverwinter Nights. 2022. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 49 Star Wars: Knights of the Old Republic 2022. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 50 Magalhães, Luis. 2022. Jade Empire. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 51 Morrowind. 2023. Verkkoaineisto. The Unofficial Elder Scrolls Pages. <<https://en.uesp.net/wiki/Morrowind:Morrowind>>. Päivitetty 21.3.2023. Luettu 6.5.2023.
- 52 Oblivion. 2023. Verkkoaineisto. The Unofficial Elder Scrolls Pages. <<https://en.uesp.net/wiki/Oblivion:Oblivion>>. Päivitetty 18.3.2023. Luettu 6.5.2023.
- 53 Bethesda Softworks to Develop and Publish Fallout 3. 2004. Verkkoaineisto. Bethesda Softworks. <[http://web.archive.org/web/20040713104039/http://www.bethsoft.com/news/pressrelease\\_071004.htm](http://web.archive.org/web/20040713104039/http://www.bethsoft.com/news/pressrelease_071004.htm)>. Päivitetty 12.7.2004. Luettu 6.5.2023.
- 54 Fallout 3. 2023. Verkkoaineisto. Fallout Wiki. <[https://fallout.fandom.com/wiki/Fallout\\_3](https://fallout.fandom.com/wiki/Fallout_3)>. Päivitetty 20.4.2023. Luettu 6.5.2023.
- 55 Pepe, Felipe. 2022. Mass Effect. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 56 Magalhães, Luis. 2022. Dragon Age: Origins. Teoksessa Pepe, Felipe (ed.). The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.

- 57 Pepe, Felipe. 2022. The CRPG Book - A Guide to Computer Role-Playing Games - Expanded Edition v3.6b. E-kirja. The CRPG Book Project.
- 58 Gaca, Christian. 2016. Skyrim has sold 30 million copies worldwide. Verkkoaineisto. Gamereactor. <<https://www.gamereactor.eu/skyrim-has-sold-30-million-copies-worldwide/>>. Päivitetty 22.3.2016. Luettu 6.5.2023.
- 59 Makuch, Eddie. 2020. Witcher 3's Sales By System--How The Game Has Sold On PC, PS4, Xbox One, And Switch. Verkkoaineisto. Gamespot. <<https://www.gamespot.com/articles/witcher-3s-sales-by-system-how-the-game-has-sold-o/1100-6475832/>>. Päivitetty 17.4.2020. Luettu 6.5.2023.
- 60 Thoman, Peter "Durante". 2014. 2014: the first year of the CRPG renaissance. Verkkoaineisto. <<https://www.pcgamer.com/2014-the-first-year-of-the-crpg-renaissance/>>. Päivitetty 31.12.2014. Luettu 6.5.2023.
- 61 The Almost Human Team. 2011. Verkkoaineisto. Legend of Grimrock. <<http://www.grimrock.net/2011/07/14/the-almost-human-team/>>. Päivitetty 14.7.2011. Luettu 6.5.2023.
- 62 Pillars of Eternity. Verkkoaineisto. IGDB. <<https://www.igdb.com/games/pillars-of-eternity>>. Luettu 21.5.2023.
- 63 Pillars of Eternity II: Deadfire. Verkkoaineisto. IGDB. <<https://www.igdb.com/games/pillars-of-eternity-ii-deadfire>>. Luettu 21.5.2023.
- 64 Divinity: Original Sin II. Verkkoaineisto. IGDB. <<https://www.igdb.com/games/divinity-original-sin-ii>>. Luettu 21.5.2023.
- 65 Kingdom Come: Deliverance. Verkkoaineisto. IGDB. <<https://www.igdb.com/games/kingdom-come-deliverance>>. Luettu 21.5.2023.
- 66 Chalk, Andy. 2019. Disco Elysium leads The Game Awards with four wins. Verkkoaineisto. <<https://www.pcgamer.com/disco-elysium-leads-the-game-awards-with-four-wins/>>. PC Gamer. Päivitetty 13.12.2019. Luettu 6.5.2023.
- 67 Taylor, Ivy. 2018. Chasing oblivion with Disco Elysium and alcohol addiction. Verkkoaineisto. <<https://www.gamesindustry.biz/disco-elysium>>. Games Industry.biz. Päivitetty 31.10.2018. Luettu 6.5.2023.
- 68 Cyberpunk 2077. Verkkoaineisto. IGDB. <<https://www.igdb.com/games/cyberpunk-2077>>. Luettu 21.5.2023.

- 69 Elden Ring. Verkkoaineisto. IGDB. <<https://www.igdb.com/games/elden-ring>>. Luettu 21.5.2023.
- 70 Hogwarts Legacy. Verkkoaineisto. IGDB. <<https://www.igdb.com/games/hogwarts-legacy>>. Luettu 21.5.2023.
- 71 Pathfinder: Wrath of the Righteous. Verkkoaineisto. IGDB. <<https://www.igdb.com/games/pathfinder-wrath-of-the-righteous>>. Luettu 21.5.2023.
- 72 Encased. Verkkoaineisto. IGDB. <<https://www.igdb.com/games/encased>>. Luettu 21.5.2023.
- 73 Bankhurst, Adam. 2022. Elden Ring Passes 16 Million Sales [Update]. Verkkoaineisto. <<https://www.ign.com/articles/elden-ring-sells-16-million-copies>>. IGN. Päivitetty 18.8.2022. Luettu 6.5.2023.
- 74 Plant, Logan. 2022. Cyberpunk 2077 Passes 20 Million Copies Sold. Verkkoaineisto. <<https://www.ign.com/articles/cyberpunk-2077-passes-20-million-copies-sold>>. IGN. Päivitetty 28.9.2022. Luettu 6.5.2023.
- 75 Verschelde, Rémi. 2022. Maintenance release: Godot 3.5.1. Verkkoaineisto. <<https://godotengine.org/article/maintenance-release-godot-3-5-1/>>. Godot. Päivitetty 28.9.2022. Luettu 6.5.2023.
- 76 Marketplace. Verkkoaineisto. Godot Marketplace. <<https://godotmarketplace.com/>>. Godot Marketplace. Luettu 6.5.2023.
- 77 Control. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/3.5/classes/class\\_control.html](https://docs.godotengine.org/en/3.5/classes/class_control.html)>. Luettu 6.5.2023.
- 78 Get-drag-data. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/3.5/classes/class\\_control.html#class-control-method-get-drag-data](https://docs.godotengine.org/en/3.5/classes/class_control.html#class-control-method-get-drag-data)>. Luettu 6.5.2023.
- 79 Set-drag-preview. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/3.5/classes/class\\_control.html#class-control-method-set-drag-preview](https://docs.godotengine.org/en/3.5/classes/class_control.html#class-control-method-set-drag-preview)>. Luettu 6.5.2023.
- 80 Can-drop-data. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/3.5/classes/class\\_control.html#class-control-method-can-drop-data](https://docs.godotengine.org/en/3.5/classes/class_control.html#class-control-method-can-drop-data)>. Luettu 6.5.2023.
- 81 ItemList. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/3.5/classes/class\\_itemlist.html](https://docs.godotengine.org/en/3.5/classes/class_itemlist.html)>. Luettu 6.5.2023.

- 82 Ray-casting. Verkkoaineisto. Godot Docs. <<https://docs.godotengine.org/en/3.5/tutorials/physics/ray-casting.html>>. Luettu 6.5.2023.
- 83 CollisionShape. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/3.5/classes/class\\_collisionshape.html](https://docs.godotengine.org/en/3.5/classes/class_collisionshape.html)>. Luettu 6.5.2023.
- 84 Occlusion culling. Verkkoaineisto. Godot Docs. <[https://docs.godotengine.org/en/stable/tutorials/3d/occlusion\\_culling.html](https://docs.godotengine.org/en/stable/tutorials/3d/occlusion_culling.html)>. Luettu 6.5.2023.
- 85 GDScript vs C# performance. Verkkoaineisto. Github. <<https://github.com/godotengine/godot/issues/36060>>. Luettu 6.5.2023.

