

VILNius gediminas technical university

faculty of fundamental sciences

department of information technologies

Dmytro Teplov

**Bachelor thesis title in lithuanian**

**Procedural generation for diverse applications: creating a versatile 2D world map generator**

Final Bachelor Thesis

Information Technologies Study Programme, State code

Information Technolodies specialization

Study Field of Informatics

Vilnius, 2024

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Fundamentinių mokslų fakultetas

Informacinių technologijų katedra

PATVIRTINTA

Katedros vedėjo

doc. dr. Dmitrij Šešok

Dmytro Teplov

**Bachelor thesis title in lithuanian**

**Procedural generation for diverse applications: creating a versatile 2D world map generator**

Baigiamasis bakalauro darbas

Inžinerinės informatikos studijų programa, valstybinis kodas 6121BX033

? specialization

Informatikos studijų kryptis

|  |  |  |
| --- | --- | --- |
| **Vadovas** |  |  |
|  | (Pedag. vardas, vardas, pavardė) |  |
| **Konsultantas** |  |  |
|  | (Pedag. vardas, vardas, pavardė) |  |

Vilnius, 2024

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

*Vilnius Gediminas Technical University*

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

*Faculty of Fundamental Sciences*

INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

*Department of Information Technologies*

PATVIRTINA / *APROVED*

##### Katedros vedėjas / *Head of the department*

doc. dr. Dmitrij Šešok

**BAIGIAMOJO BAKALAURO DARBO UŽDUOTIS**

***ASSIGNMENT OF BACHELOR THESIS***

Vilnius

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Studentei | | Name Surname | | | |
| *For student* | |
| Baigiamojo darbo (projekto) tema: | | | Title in Lithuanian | | |
| *Bachelor thesis topic:* | | | *Title in English* | | |
|  | | | |  | |
| Baigiamojo darbo užbaigimo terminas | | | | Bakalauro baigiamasis darbas 2 atsiskaitymo data | |
| *The deadline for bachelor thesis* | | | | *Bachelor Thesis 2 exam date* | |
| BAIGIAMOJO DARBO (PROJEKTO) UŽDUOTIS: | | | | | |
| Assignment text in Lithuanian. | | | | | |
| *THE ASSIGNMENT OF BACHELOR THESIS:* | | | | | |
| Assignment text in English. | | | | | |
|  | | | | |  |
| Baigiamojo bakalauro darbo (projekto) konsultantai: | | | | |  |
| *The consultant(s) of bachelor thesis:* | | | | |
|  | | | | | (Pedag. vardas, vardas, pavardė / *Academic title, name, surname* ) |
| Vadovė |  | | |  |  |
| *Supervisor* |
|  | (Parašas / *Signature*) | | |  | (Pedag. vardas, vardas, pavardė / *Academic title, name, surname* ) |

There will be an annotation from mano.vilniustech.lt system (in Lithuanian). You can fill the table of add the one which will be generated in the system

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vilniaus Gedimino technikos universitetas  **Fundamentinių mokslų** fakultetas  **Informacinių technologijų** katedra |  | | | |
|  | | | | |
| **Informacinių technologijų** studijų programos baigiamasis bakalauro/magistro darbas  Pavadinimas **Baigiamojo darbo pavadinimas** | | | | |
| Autorius **Vardas Pavardė** | | Vadovas pedag. vardas **Vardas Pavardė** | | |
|  | | | | |
|  |  | | **Kalba** | |
|  |  | |  | lietuvių |
|  |  | |  | anglų |
|  |  | |  |  |
| **Anotacija** | | | | |
| Baigiamajame magistro darbe nagrinėjamos investicinių projektų vertinimo problemos, vertinimo būtinumas ir tikslai, užsienio šalių patirtis šioje srityje. Išnagrinėti pagrindiniai principai, kuriais remiantis nustatomas investicinių projektų efektyvumas. Pateiktas investicinių projektų efektyvumo vertinimo klasifikavimas, išanalizuoti įvairių jo rūšių skirtumai. Projekto efektyvumas apskaičiuojamas analizuojant investicinio projekto pinigų srautus. Baigiamajame darbe nagrinėjama neapibrėžtumo bei infliacijos įtaka investicinio projekto efektyvumui. Vertinant infliacijos įtaką, išnagrinėtas dabartinių ir prognozuojamų kainų tarpusavio ryšys. Išnagrinėjus teorinius ir praktinius investicinių projektų vertinimo aspektus, pateikiamos baigiamojo darbo išvados ir siūlymai.  Darbą sudaro … dalys: įvadas, ……………………, išvados ir siūlymai, literatūros sąrašas.  Darbo apimtis – 96 p. teksto be priedų, 24 iliustr., 16 lent., 34 bibliografiniai šaltiniai.  Atskirai pridedami darbo priedai. | | | | |
|  |  | |  | |
| **Prasminiai žodžiai:** bent 5 žodžiai | | | | |

There will be an annotation from mano.vgtu.lt system (in English).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vilnius Gediminas Technical University  **Faculty of Fundamental Science**  **Information Technology** department |  | | | |
|  | | | | |
| **Information Technology** bachelor study program final thesis  Title **Title of the thesis** | | | | |
| Author **students** **Name Surname** | | Supervisor prof. dr. **Name Surname** | | |
|  | | | | |
|  |  | | **Language** | |
|  |  | |  | Lithuanian |
|  |  | |  | English |
|  |  | |  |  |
| **Annotation** | | | | |
| Annotation text  … | | | | |
|  |  | |  | |
| **Keywords:** … | | | | |

**Table of Contents**

[Introduction 9](#_heading=h.30j0zll)

[1. Analysis and Overview of Relevant Technologies 11](#_heading=h.1fob9te)

[1.1. Review of Similar Systems 11](#_heading=h.3znysh7)

[2. System Requirements Specification 12](#_heading=h.1t3h5sf)

[2.1. Overview 12](#_heading=h.4d34og8)

[2.2. Overall Description 12](#_heading=h.2s8eyo1)

[2.3. Specific Requirements 12](#_heading=h.3rdcrjn)

[3. System Architecture Description 14](#_heading=h.z337ya)

[3.1. Architecture 14](#_heading=h.3j2qqm3)

[3.2. Dynamic Behavior of Architecture 14](#_heading=h.1y810tw)

[4. System Documentation 15](#_heading=h.4i7ojhp)

[5. System Testing 16](#_heading=h.2xcytpi)

[Conclusions 17](#_heading=h.1ci93xb)

[References 18](#_heading=h.3whwml4)

[Annexes 19](#_heading=h.qsh70q)

[Annex A. Title of the Annex 19](#_heading=h.3as4poj)

[Annex B. Priedo pavadinimas 19](#_heading=h.1pxezwc)

**List of Images**

[**Figure 1.** All images must have a title, defining what is shown in the image 13](#_heading=h.2et92p0)

**List of tables**

[**Table 1.** All tables should have titles as well 13](#_heading=h.tyjcwt)

**List of abbreviations and terms**

|  |  |
| --- | --- |
| **RFID**  **NFC**  **QR Code**  **PICC** | Radio Frequency Identification  Near Field Communication  Quick Response code  Proximity Integrated Circuit Card |
| **USB**  **EEPROM**  **RAM**  **SRAM**  **…** | Universal Serial Bus  Electric erasable and programmable read-only memory  Random Access Memory  Static Random Access Memory  … |
|  |  |
|  |  |
|  |  |

**Introduction**

**Relevance of the thesis.**

In the world of in-game cartography, artists have always held the dominating power. Behind all of the world maps from iconic games like The Witcher 3: Wild Hunt, Red Dead Redemption 2 or Control there is always at least one artist, a team more likely. For small teams of game developers who could not afford to commission an artist for such task, there has always been a need for a tool that would help them achieve great results using minimal input. Additionally, artists working on map creation for video games, or even for board games, could be interested in a tool that would expedite their workflow. This tool could be used for layout prototyping or to create a finished map with their own stylized assets.

**Problem of the thesis**

The primary challenge this thesis aims to address is the creation of a user-friendly desktop application that enables individuals, regardless of their artistic abilities, to design stylized 2D world maps for various projects. These projects can range from board games like Dungeons and Dragons to in-game world maps for indie game studios. A significant feature of the proposed tool is the ability to upload custom assets. This allows users to create maps in any style they desire, making it particularly appealing to indie game studios that typically have at least one artist on the team capable of creating unique assets.

However, developing such a tool presents several challenges. These include ensuring ease of use for non-artists, implementing effective procedural generation algorithms for various map features, and allowing for the seamless integration of custom assets. This thesis will delve into these issues and propose solutions to overcome them.

**Objective of the thesis** – enhance the efficiency and quality of 2D map generation, for a variety of applications such as indie video game development, tabletop games, and literature by developing a procedural 2D map generator application.

**Tasks of the thesis:**

1. Analysis of existing tools for map generation and analysis of algorithms for area approximation, texture blending and uniform distribution.
2. Analysis of multiplatform desktop development technologies.
3. Design 2D map generation tool.
4. Develop and test the prototype.
5. **Map generation algorithms and tools**

**Analysis of current state of 2D map creation**. Map creation is a historically labor-intensive endeavor [1]. They are vital in numerous fields, providing a simplified and easily digestible way to represent spatial information and in some cases position of the user. Maps, be it digital or analog, are created by artists and require considerable amount of time. From that point on, the focus of this paper will be on digital 2D maps created for virtual worlds.

Currently, digital 2D map for virtual worlds creation process involves several steps. Creating initial prototype of a map is a first step and it would include rough blocking, where the artist would outline areas like continents, bodies of water, terrain types or, on the smaller scale, cities and towns, roads, forest areas and so on. Most of the time there are several prototypes carried out to figure out the layout of the map [2].

Next step would be to flash out the artistic style of the map and intricacies regarding that. That would typically include the defining the distinct art style and producing various assets in this art style to eventually place and blend them to final canvas [2].

This paper heavily oriented at gaming industry, since it is a primal producer of virtual world that require maps, however the goal of the developed application is to satisfy the needs of any potential end-user requiring digital cartographic solution. A fantasy book writer and a board game developer should also be able to benefit from the developed tool.

**Significance of this project**. Developing a tool that would allow to expedite parts of map creation process is not a new concept. However, this project is primarily geared towards accommodating individuals with varying levels of expertise, ranging from those with no prior experience with painting applications to seasoned, experienced artists.

There are several already existing tools and applications that help to accelerate parts or the whole process, but some are focusing too much on giving control to the user and getting too close to overwhelming them if they want to quickly create a map for their project. Others would restrict the process and make almost everything procedurally and leave only minor changes to the user, like background color change.

This is where this project would come as a solution. It will be appealing to both sides. The core mechanic will be the ability to outline the area types on the canvas by choosing an area in the user interface and paint the areas directly in the application. This 2D map generation tool will allow to convert quick brush strokes into a beautiful stylized map regardless of user skill level. In addition to its user-friendly nature, the application will feature a comprehensive toolset designed to satisfy the needs of professionals as well. This toolset will help professionals to not feel any limitations due to the application's procedural nature, as it enables a level of fine-tuning akin to most popular painting applications.

The average independent game developer would find significant value in integrating this tool into their standard workflow. Doing so will alleviate the pressure of generating distinct maps for each project, especially in due to the time and budget constraints often encountered by smaller studios.

Conversely, larger studios would also derive notable advantages from this tool since it would allow for finer adjustments and the importation of uniquely styled assets, a level of customization made feasible by the presence of dedicated 2D artists within the studio. An important aspect of the project lies in its open-source nature, signifying that individuals with sufficient resources have the capability to adapt the application to better align with their specific requirements.

* 1. **Map generation tools and technologies**
     1. **inkarnate.com**
     2. **watabou.github.io**
     3. **Competitor 3**

**Algorithms** are a core part of this project. And as such there will be extensive algorithm analysis to pick the most suited ones.

This section is divided in four parts to discuss algorithms for uniform point distribution over an area, noise generation and texture blending separately. Each section will comprehensively discuss a selection of algorithms, providing an analysis of the advantages and disadvantages associated with each algorithm.

* + 1. **Uniform point distribution algorithms over an area (modification for towns)**

Uniform point distribution algorithms are a vital part of the asset distribution during the map creation. Determining the placement of each building within the user-defined ‘town’ area in a manner that avoids overlap and maintains logical coherence presents a significant challenge.

In this section the discussion will be about Poisson-disk sampling and why it is better than basic noise distribution.

Poisson-disk sampling (PDS) is a method that ensures each sample is independent with a certain random distribution, while maintaining a minimum distance ‘r’ from each other [4]. The concept of distance in this context is quite abstract, as the definition of samples vary drastically from application to application. PDS is a method, a strategy, and thus requires an efficient algorithm to execute this method of sampling.

Maximal Poisson-disk sampling (MPDS): a set is considered maximal when no additional samples can be added to the sampling domain without breaching the minimum distance requirement, indicating that the domain is fully occupied [7].

First off, the method itself can be broken down into 5 steps, this unoptimized version is also called “dart throwing” technique [7]:

* + - * Define domain. Setting the boundaries of an area within which the samples should be generated is the first step towards generating the uniform distribution.
      * Choose value r that will represent minimum distance between samples.
      * Randomly generate a sample on the domain, and check if it satisfies the minimum distance property. The sample is added to active list, a list containing all valid samples, in case it is located farther than distance r to other and discarded otherwise.
      * In order to satisfy the maximal condition, step 3 will be executed until there is no place to put an additional sample.

Such method, coded naively will perform badly, and in case if maximal condition must be satisfied – extremely bad.

There are two major algorithms for calculating MPDS and each introduce separate optimizations

* + 1. **Bounding box calculation methods**
    2. **Base noise calculation algorithms**
    3. **Texture Blending (Texture Splatting Algorithm)**
  1. **Analysis of multiplatform desktop development technologies**
     1. **Programming Language**
     2. **Graphics Application Programming Interface**

1. **System Requirements Specification** 
   1. **Diagrams**
2. **System Development**

**Conclusions**

1. …
2. …

**References**

1. T. Gao, J. Zhang and Q. Mi, "Procedural Generation of Game Levels and Maps: A Review," 2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Korea, Republic of, 2022, pp. 050-055, doi: 10.1109/ICAIIC54071.2022.9722624.
2. Röhl, Tobias & Herbrik, Regine. (2008). Mapping the Imaginary - Maps in Fantasy Role-Playing Games. First publ. in: Forum Qualitative Sozialforschung 9 (2008), 3, Art. 25. 9.
3. Yao, X., Zhang, Y., Bao, F. et al. The blending interpolation algorithm based on image features. Multimed Tools Appl 77, 1971–1995 (2018). <https://doi.org/10.1007/s11042-017-4379-5>
4. Wang, T. (2021). Poisson-Disk Sampling: Theory and Applications. In: Lee, N. (eds) Encyclopedia of Computer Graphics and Games. Springer, Cham. <https://doi.org/10.1007/978-3-319-08234-9_398-1>
5. Yan, DM., Guo, JW., Wang, B. et al. A Survey of Blue-Noise Sampling and Its Applications. J. Comput. Sci. Technol. 30, 439–452 (2015). <https://doi.org/10.1007/s11390-015-1535-0>
6. R. Bridson. Fast Poisson Disk Sampling in Arbitrary Dimensions, SIGGRAPH 2007
7. David Cline, Kenric B. White, Parris Egbert, "Poisson Disk Point Sets by Hierarchical Dart Throwing", Symposium on Interactive Ray Tracing, pp. 129-132, Sept. 27.
8. Quan, W., Yan, D-M., Guo, J., Meng, W., & Zhang, X. (n.d.). Maximal Poisson-disk Sampling via Sampling Radius Optimization. NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

**Annexes**

* + - 1. **Title of the Annex**

Pagrindinis tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas.

* + - 1. **Priedo pavadinimas**

Pagrindinis tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas tekstas.