**Project Plan**

DV1478 Kandidatarbete i datavetenskap

2017-04-30

|  |  |  |
| --- | --- | --- |
| Thesis | Tentative title | Procedural city generation viable in games |
| Classification | Theory of computation, Randomness, geometry and discrete structures, Computational geometry |
| Student 1 | Name | Elias Frank |
| Email | elias.frank@gmail.com |
| Social security number | 19940327–4179 |
| Student 2 | Name | Niclas Olsson |
| Email | ohlsson.niclas@gmail.com |
| Social security number | 19930421–3797 |
| Supervisor | Name | Hans Tap |
| Email | hans.tap@bth.se |

**1 Introduction**

Exploring a huge open world environment is a desirable feature in games. But creating a big open city such as in the *Grand Theft Auto*[1] series and *Batman: Arkham City*[2] involves years of work for a lot of people. Making big open cities in games is simply not feasible for most game companies. These games all have massive success with their big open worlds making a feasible generated city an attractive technique for smaller companies to be able to compete with big open worlds of their own.

To create massive amounts of content without a big workforce algorithmically based solutions exist, *Procedural Content Generation* (PCG). PCG was in the past used to minimize the disk space required for games. *.kkrieger*[3] is an excellent example of this. It has since evolved into a method to minimize workforce required for content. *No man’s sky*[4] is an excellent example of a game using PCG to minimize workforce while maximizing content.

**2 Aim and objectives**

This work aims to explore the possibility of using Perlin noise to procedurally generate a city that can be used in games. By conducting a user study, we will investigate whether the resulting city is considered believable in a game setting.

We aim to achieve the following objectives:

* Implement a method for procedurally generating cities using Perlin noise.
* Implement a user interface that allows the user to change the way the city is generated.
* Test whether the generated cities are believable.
* Use the test results to answer the research question.

**3 Research question**  
Can Perlin noise be used in a hierarchical manner to procedurally generate a city viable in games?

**4 Method**

## 4.1 Constraints

## 4.2 Implementation plan

## 4.3 Experiment design

Testing implementation 5? Times, answer questionnaire.

## 4.4 Viability

District, Roads, Houses, City

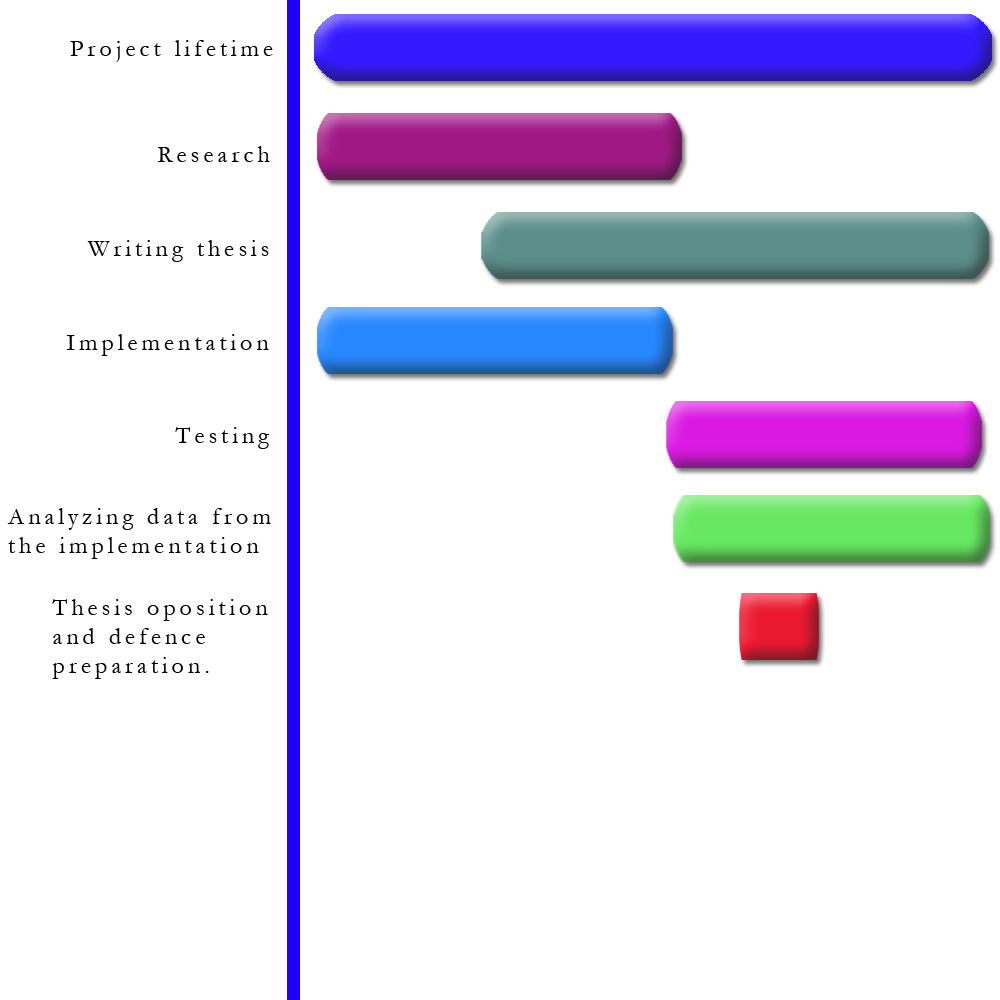
## 4.5 Data Collection

Experiment Questionnaire, loading times, ram/vram?

**5 Expected outcomes**

With Perlin noise a city will be generated. With the user study, we will gather data through a questionnaire that will show that the users find the city viable to use in games.

**6 Time and activity plan**

  
*Figure 3: Visual representation of activity plan*

The implementation and research will be the first things to be done and they will be done simultaneously. Some parts of the implementation require research before we have the knowledge to implement it, but all the parts of the implementation, such as the rendering pipeline, that do not require any research, will be done in conjunction with the research.

The writing of the thesis will begin shortly after we have some basic research and base implementation done. The thesis will be worked on every week until completion. When the implementation has enough features, data gathering and analysis of this data will begin. This data is crucial to answer the research question and conclude the thesis.   
MINDRE TEXT HÄR?

* Project lifetime 2017-03-27 – 2017-06-11
* Research 2017-03-27 – 2017-05-14
* Writing thesis 2017-05-01 – 2017-06-11
* Implementation 2017-03-27 – 2017-05-14
* Testing 2017-05-14 – 2017-06-11
* Analyzing data 2017-05-14 – 2017-06-11
* Thesis opposition and defense 2017-05-22 – 2017-05-28

**7 Risk management**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Probability (1-5)** | **Severity (1-5)** | **Mitigation** |
| Procedurally generating a city is difficult and consumes too much time. | 3 | 5 | Only implement the most crucial features to answer the research question. |
| Failing to communicate with thesis partner. | 2 | 4 | Have daily meetings and work together when possible. |
| Technical issues | 2 | 5 | Make sure all work is on several hard drives. Use git as source control. |
| Collect data from the implementation wrong, twisting the results. | 3 | 4 | Carefully decide what parameters in the implementation to collect data from and in what form to collect this data. |
| Defective construction of the implementation. | 2 | 4 | Before coding the implementation make sure there is a good plan to follow. |