MGAIA Assignment 1: Procedural Content Generation

Leiden University

February 2024

1 Introduction

In this assignment, we want you to familiarize yourself with Procedural Content Generation (PCG). Minecraft is a host to many interesting AI challenges, such as settlement generation [Sal+18][1] (see Figure 1), or co-operative natural language interaction [Kis+21]. You will be generating a House in Minecraft.

Figure 1: Entry from the Generative Design in Minecraft Challenge (GDMC). This settlement was constructed algorithmically and not built manually.

[1] https://gendesignmc.engineering.nyu.edu/

2 Required Setup

1. Install Minecraft 1.19.2 (Java Edition). In the Official Minecraft Launcher, go to `Installations` → `New Installation` → `Version` → `Release 1.19.2`. Before you go to the next step, press Play, otherwise it won't download. *Note that we of course discourage third-party launchers, such as TLauncher, that allow you to download the official client from the official minecraft website, without owning an account.*

2. Install Forge for Minecraft 1.19.2: https://files.minecraftforge.net/net/minecraftforge/forge/index_1.19.2.html. Note that you might need an up-to-date Java version to install Forge. To our knowledge `openjdk version 17+` suffices.

3. Install the GDMC HTTP interface mod (v1.0.0): https://github.com/Niels-NTG/gdmc_http_interface/releases/tag/v1.0.0. The release contains a *.jar file* that you need to install as a Forge mod[2].

4. Install GDPC (the Python package you will be using). You need to clone https://github.com/avdstaaij/gdpc and install the package by running `pip install .` in the root of the cloned repository. Alternatively, you can install it using `pip install git+https://github.com/avdstaaij/gdpc` . **Do not use "pip install gdpc"**, as that will give you an outdated version.

5. Open Minecraft **with** the Forge 1.19.2 profile selected. Open a world, set a build area[3], and try to run examples/emerald_city.py. If this works, you should now be ready.

6. Familiarize yourself with GDPC by going through the guides at examples/tutorials

3 Your Task

Your code will procedurally generate a building in the *build area*[3]. When testing your submission, we will use a build area of 100x100 blocks. Of course, you don't know where we will test it, so your building should smoothly integrate into the existing world. Do not just clear all blocks in a bounding box, but make sure your algorithm attempts to scan the selected environment, and places its structures in a fitting way. This is the difficulty and essence of Procedural Content Generation! Adaptability to the infinite possibilities that RNGesus might throw at you.

Moreover, you will be writing a scientific report. There you will make the intentions behind your architectural design clear, and reference inspirations and applied algorithms to achieve your goal.

Your generated structure should include *at least* the things below, but more can only be better.

- A house of a specific architectural style
- Interior decoration
- Appropriate adaptation to the terrain in the build area

[2] https://github.com/Niels-NTG/gdmc_http_interface/blob/master/docs/Installation.md

[3] In Minecraft, press `t` and type `/setbuildarea [x0] [y0] [z0] [x1] [y1] [z1]` (https://minecraft.fandom.com/wiki/Coordinates#Commands)

We expect you to attempt to implement these things in a way that the world is *believable*. Believable in the sense that, e.g., a house should not spawn directly in the middle of an ocean where it would not be reachable. And ideally, you do not "heavily" alter the area where it should be placed, but of course if you are in the middle of a jungle and there is no continuous free space to put your structure due to too many trees, you are allowed to make free space.

Again: take randomness into account! The result of your program shall not always produce the exact same thing 1:1 within a bounding box, but instead adapt and vary! Not 1:1 means your house should also vary, by placing some windows or other things a little different every time they are placed. The keyword is **Procedural Content Generation**, not place a given structure at point X,Y,Z.

4 Submission

Make sure to nicely document everything that you do in the report. Your final submission consists of:

- Source code for a python program that uses the GD**P**C Python package and integrates a House believably into the world.
  - This includes a `Readme` on how to run your code.
- A self-contained scientific pdf report with figures, **references**, etc. The page count we expect of you might vary depending on your layout (4-8 pages would be reasonable). This report contains:
  - An explanation of the techniques you applied, e.g.
    - How do you find an area to place your building? Include plots that visualize your terrain evaluation
    - What is variable / random in the building process?
  - How you addressed *believablity*.
  - **Multiple** example figures showing the *variation* as well as *adaptability* to different environments.
  - Overall conclusions.

If you have any questions about this assignment, please visit our lab sessions on Thursdays where we can help you out. In case you cannot make it in person, you can post questions about the contents of the course on the Discord discussion forum, where other students can also read and reply to your questions.

The deadline for this assignment is **09.03.2024 at 23:59**.

5 Grading Criteria

Your grade for the assignment will be made up of:

- 33% Believability & Adaptability
- 33% Randomness/Variation of Procedurally Generated Content
- 33% Report

References

[Sal+18] Christoph Salge et al. "Generative design in minecraft (GDMC) settlement generation competition". In: *Proceedings of the 13th International Conference on the Foundations of Digital Games*. 2018, pp. 1–10.

[Kis+21] Julia Kiseleva et al. "Neurips 2021 competition iglu: Interactive grounded language understanding in a collaborative environment". In: *arXiv preprint arXiv:2110.06536* (2021).

6 Examples from previous year

Note that these images are solely meant as inspiration and do not serve as a solution you should be attempting to copy exactly.

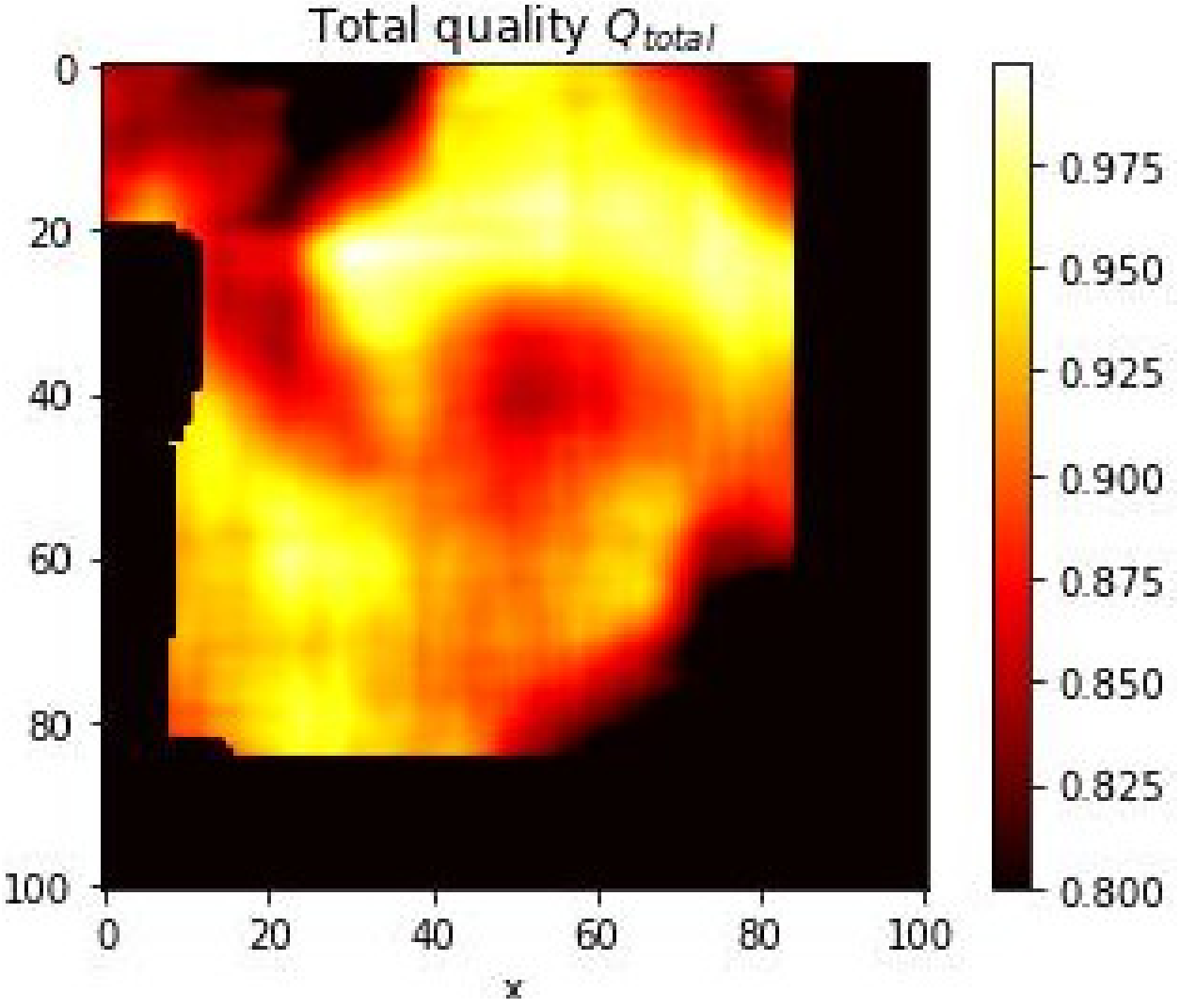

Figure 2: An example plot in which students visualize the final evaluated suitability of the building area. This was preceded by a few other individual plots that show individual features that have been combined in this plot.

Figure 3: An example in which students procedurally generate a tree including houses as leaves

Figure 4: An example in which the student highlights the variety of building styles and sizes