**Project title**

An investigation into techniques for procedurally generating realistic cities.

**Brief outline of the work**

I will explore and prototype some different techniques for procedurally generating cities. I will be evaluating these techniques against several criteria and use this to pick a single technique to move forward with and refine.

**Rationale:**

Previously I have worked with some procedural algorithms for dungeon generation in a dungeon crawler game and found this to be fantastic, I believe it allows for more re-playability as it means that the game isn’t the same every time it’s played. I’d like to research and further develop procedural generation techniques and this project allows me to do so.

I’d also like to see if the use of using rules/restrictions to control the outcome of procedural generation can ensure that what is created will always be suitable for gameplay purposes: if a minigame took place in a procedurally generated city, would I be able to control the generation to ensure that the level was not only just always completable and playable, but also fun for the player, and could I procedurally generate something of an acceptable quality when compared to a hand-crafted level, these are the questions I’d like to explore.

**Project timeline and Milestone deliverables:**

Milestone 1: End of October

To have successfully created one prototype capable of:

* Major roads
* Minor roads
* Building placement

Milestone 2: End of December

To have prototyped multiple systems capable of the above bullet points. To have evaluated these different techniques against a set of criteria, and then choose and justify my decision of which technique to continue onwards with.

Milestone 3: End of project

Using the technique that I chose to continue with, I would like to have a completed system that can create Major and Minor roads that conform to global bounds and local constraints, choose where to place buildings and procedurally generate these buildings to create a more diverse and visually pleasing city. I would also like to have implemented methods of producing my own maps for any required user input (Height maps, etc.).

Stretch goals:

If all above is achieved earlier than expected and I have additional time, I would like to further focus on methods of procedural building generation and look into procedurally texturing these buildings.

**Clear employability statement of how the completed project will demonstrate the relevant specialist skills:**

This project will allow me to demonstrate ability to procedurally generate “random” content in a controlled way.

Algorithms such as the Drunkard walk are a fully-random method of creating procedurally generated content, however, as fully-random methods have little-to-no condition checks or rules in place , they are not well suited for making realistic and believable worlds, and it also does not ensure that they are appropriate for gameplay. This project will allow me to demonstrate the ability to restrict and control the random elements in such a way that the content is created sensibly and produces realistic output.

I will also be demonstrating the use of systems such as L-systems which are used commonly in procedurally generated content for many reasons such as creating trees and plants for procedurally generated forests, and can also be used for creating textures.

Finally, I’ll be demonstrating my ability to pick up and learn new systems and methodologies, evaluate these and iterate on them to produce the best outcome. This ability to pick up and learn new systems and concepts is one that is highly valued in many job listings I have looked at.

**Examples of specific organisations and/or current jobs in the relevant industry sector in which the specialist skills will be of value:**

Jobs in which experience with programming and scripting is of great value.

|  |  |  |
| --- | --- | --- |
| **Job title** | **Essential and Desired skills/experience** | **Link** |
| Junior Game Developer | * 2:1 or 1st Degree in Computer Science or Computer Games related degree * Good OOP knowledge * Thirst for knowledge and keen to continue learning | <http://www.jobsite.co.uk/job/junior-game-developer-958177035> |
| Junior Unity Mobile Game Developer | * Unity Game Engine * Java or C# * Android/iOS * REST API’s * Virtual Reality * Strong communication skills * 2:1 Degree or above | <http://www.jobsite.co.uk/job/junior-unity-mobile-games-developer-unity--android--ios-20k--958217968> |
| Entry level / Junior / Graduate Unity Games Developer | * Unity Game Engine skills * C# scripting expertise * Degree in Games Development or related subjects | <http://www.jobsite.co.uk/job/entry-leveljuniorgraduate-unity-games-developer-958142348> |

**What do I wish to be marked on for the final project?**

Similar to my criteria for evaluating the different procedural generation methods, I propose to be marked on the following:

Realism

How “realistic” is the city. Does it resemble cities we see today? Does it *look* like a real place?

Scalability

Can the final system be scaled to produce small or large cities? Does it face any issues with this? If I produce a large city, is it obvious that there are repeating/reoccurring themes/segments which make it obviously procedurally generated?

Diversity

Similar to what was addressed in scalability, if a large city was produced, does this city have a range of diversity in its roads and buildings, or is there a lot of repeated content?

**Annotated bibliography**

Initial reading and research:

|  |  |  |
| --- | --- | --- |
| **No.** | **Item** | **Description/Annotation** |
| City focused | | |
| 1 | G Kelly, H Mccabe. (2006). "A Survey of Procedural Techniques for City Generation". In ITB Journal, No. 14. | This paper starts by introducing multiple procedural techniques such as fractal, l-systems and noise. It goes on to evaluate these techniques in city generation based on criteria such as realism, scale, input, efficiency, etc. |
| 2 | Sun, J. Yu, X. Baciu, G. Green, M. (2002). "Template-based generation of road networks for virtual city modeling". In Proceedings of the ACM symposium on Virtual reality software and technology (VRST '02). Acm, New Work, NY, USA, 33 - 40. | This paper talks about various road-templates such as population-based, raster and radial which are used in template-based generation. It explains each of these templates and goes onwards to talk about validity control in the creation of major and minor roads. |
| 3 | Parish, Y, I, H. Muller, Pascal. (2001). "Procedural modeling of cities". In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01). Acm, New York, NY, USA, 301 - 308. | In this paper, the authors present CityEngine which is a system capable of procedurally generating cities using user-controlled input data such as height maps and population density maps. They talk about using L-Systems and road patterns for creating the city layout and go on to talking about procedural building geometry (buildings) and textures. |
| 4 | Chen G. Esch G. Wonka P. Mueller P. Zhang E. (2008). "Interactive Procedural Street Modeling" *In Proceedings of SIGGRAPH 2008.* ACM Trans. Graph. Article 103: 1-10. | This paper discusses the use of tensor fields and focuses more on user interactivity. It talks about a system which creates a tensor graph and allows the user to edit/adjust this to edit the generated city. |
| 5 | Evans, M. (2015). Procedural Generation For Dummies: Road Generation. [Online] 11 December 2015. Available online: [http://martindevans.me /game-development/2015/12/11/ Procedural-Generation-For- Dummies-Roads/](http://martindevans.me/game-development/2015/12/11/Procedural-Generation-For-Dummies-Roads/) [Date of access:  19 May 2016] | This article also talks about the use of tensor fields and references the above paper. It talks about the use of global goals and local constraints when creating major and minor roads and shows how they created road networks using different road templates (radial, grid). |
| 6 | Ilangovan, K, P. (2009) *Procedural City Generaror,* MSc thesus, Bournemouth University. Available at: [https://nccastaff .bournemouth.ac.uk/jmacey/ MastersProjects/MSc09/Ilangovan /Thesis\_i7834000.pdf](https://nccastaff.bournemouth.ac.uk/jmacey/MastersProjects/MSc09/Ilangovan/Thesis_i7834000.pdf) (Accessed: 30 May 2017). | This paper talks about some techniques used for city generation such as L-systems, however this paper is useful to me as it talks about terrain generation; it takes grey scaled maps or contour maps as input and it uses these to create its own heightmap and can use this to create its own terrain. This is useful for me as I can use these techniques to find out where water will be in the world, and can also use these same techniques and treat them as a population density map which will define where my major roads are placed. |
| Building focused | | |
| 7 | Greuter, S. Parker, J. Stewart, N. Leach, G. (2003). "Real-time procedural generation of 'pseudo infinite' cities". In Proceedings of the 1st international conference on Computer graphics and interactive techniques in Astralasia and South Eas  t Asia (GRAPHITE '03) Acm, New York, Ny, USA, 87 - ff. | This paper focuses on creating cities with a diverse range of buildings, and thus focuses on building generation and not generating the city layout. It proposes a method of building generation by splitting the city up into cells and using a hash function to create a number for each cell to be used as seed. This seed determines the properties of the buildings in the cell (number of floors, height, number of shapes, etc). The building is created by creating and placing several shapes together within the confined space of the building and then extruding each shape to the height of a specific floor. This creates a diverse range of buildings. |
| 8 | Muller P. Wonka P. Haegler S. Ulmer A. Gool V L. (2006). "Procedural Modeling of buildings". *In Proceedings of ACM SIGGRAPH 2006.* ACM New York, NY, USA, 614 - 613. | This paper focuses on the procedural modelling and texturing of buildings. It talks about the use of shape grammars in procedural building creation. When it comes to procedural texturing, it talks about using occlusion for checking for intersections between shapes and then uses snapping to avoid texturing errors that previous procedural texturing methods have faced such as having a window placed where an intersection of the building occurs. |
| Technique specific | | |
| 9 | Martz, P. (1997). Generating Random Fractal Terrain. [Online] 1997. Available Online: [http://www.game programmer.com/fractal.html](http://www.gameprogrammer.com/fractal.html)  [Date of access: 09 May 2016] | This article speaks about creating fractal two-dimensional and three-dimensional terrains, and later goes onwards to show how these can create height maps. |
| 10 | Havey, D. (2008). *Tutorial #7: Voronoi diagrams* [Online] May 4, 2008. Available online: [http://donhavey.com/ blog/tutorials/tutorial-7-voronoi-diagrams/](http://donhavey.com/blog/tutorials/tutorial-7-voronoi-diagrams/) [Date of access: 30 May 2017] | This article talks about Voronoi diagrams, their use in creating real world maps and then goes on to talk about how these would be implemented, the logic behind how they’re created and what condition checking is necessary. |