CIS3161

Project Proposal - ‘The Development of a Video Game using Procedural Terrain Generation’

James Edwards – Computing (Games Programming)
Supervisor: Saad Saihi

Table of Contents

Table of Figures

Introduction

Procedural Content Generation (PCG) is a field of games development that involves parts of the games content being generated using algorithms (Liu, J. et al., 2020; A. Summerville et al., 2018). Developing new content for a game is very time consuming and expensive when done by hand, therefore PCG is useful as it can generate content faster and cheaper than creating it manually (F. Amato and F. Moscato, 2017). Due to this, many game developers and studios use procedural content generation when creating games. An example of games that have used PCG is game such as No man's sky (Hello Games, 2016), Minecraft (Mojang, 2011) and Astroneer (System Era Softworks, 2016).

Research around 3D terrain generation has mainly focused on generating aesthetically pleasing terrain, which leaves a gap in research exploring methods of generating gameplay focused procedurally generated terrain (Pech, Lam and Masek, 2020). Due to this, this project will be focused on generating mountainous 3D terrain that is focused on the gameplay of a ski racing game.

Project Background and Literature Review

Perlin noise is one of the most popular methods of 3D terrain generation. Perlin noise is a noise function that generates pseudo-random values that gradually increases and decreases (Unity, N.d.). An image of 2D perlin noise is shown below.

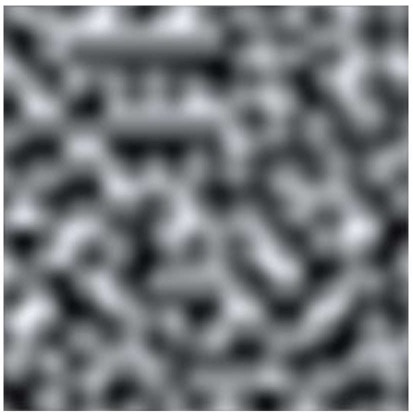


Figure 1 2D Perlin Noise (Unity, N.d.)

This 2D noise texture can then be used as a height map in order to generate 2D terrain. Games such as Minecraft (Mojang, 2011) use this technique to generate a height map that is used for its infinite voxel terrain. A benefit of Minecraft (Mojang, 2011) using this technique is that Perlin noise heightmaps can be reproduced by using the same seed value, this allows the terrain to be generated in chunks. These chunks allow it to be used for an infinite world. For a more realistic look on terrain multiple different planes of perlin noise can be stacked on top of each other to make the noise less smooth and rockier (Lague, 2016).

Another algorithm that is commonly used for PCG is the wave function collapse algorithm, this algorithm is used to generate content based on an input of rules (H. Kim et al., 2019). The real time strategy roguelite game 'Bad North' by Stålberg et al. (2018) uses the wave function collapse algorithm to generate the islands that are used in levels in the game. The terrain in this game plays a large effect on the gameplay and therefore shows that PCG can be used to generate gameplay focused terrain for video games. Although, a downside of the levels generated in 'Bad North' (Stålberg et al., 2018) is that they have a very blocky structure to them and do not look realistic, this works well in the context of the cartoony art style used in the game but in this project the terrain generated will be smoother and more realistic.

Other games such as the mountain bike racing game 'Descenders' by RageSquid (2019) use procedural content generation to generate realistic terrain for the mountain bike trails in the game. This project aims to use PCG to generate large mountainous terrain with a ski trail down it for the player to race down.

Literature Review

In a 2024 conference Jain, Sharma and Rajan (2024) proposed a framework for generating infinite 3D terrain using deep learning that can have different levels of detail. This method uses a Terrain Completion Module that is trained to generate infinite terrain as well as a Terrain Enhancement Module which is trained to enhance the terrains detail and produce multiple levels of details for the terrain. This approach is impressive as it can produce heightmaps that look very similar to real life heightmaps of terrain. A downside of this paper is that is compares it's results to the most basic form of perlin noise and does not show a comparison with terrain that is generated using more advanced techniques like fractal perlin noise and mixing in other noise functions to create more accurate terrain. This technique may also not be very applicable to games as although it can generate infinite terrain it does not include gameplay elements in the terrain.

In a 2019 conference paper Liu et al. (2019) used procedural content generation in order to generate levels for the game 'Kingdom Rush: Frontiers' by breaking down the game's levels into 3 different building blocks; roads, tower locations and monsters. This approach successfully generates levels for this game, but it is only applicable for generating levels for 'Kingdom Rush: Frontiers' as well as other similar tower defence games. Although, the approach of breaking down a games levels into different buildings blocks could be applied to many different genres and types of games.

In 2016 Golubev, Zagarskikh and Karsakov (2016) proposed a new method of terrain generation that was based on a modified version of Dijkstra algorithm. The terrain generated by this algorithm could be controlled using parameters which would give the designer more control on the result. This method of generating terrain can generate very aesthetically pleasing and realistic desert terrain using these parameters but the mountainous terrain that has been generated in this paper does not look very realistic.

Project Aim and Objectives

The Aim of the project is to: Create a skiing game that uses procedural content generation (PCG) to generate the terrain and the levels.

No	Objective Title	Objective Description
1	Create procedural skiing mountain terrain using PCG techniques.	Create a mountain by first procedurally generating a heightmap using PCG techniques such as fractal perlin noise then applying this heightmap to an Unreal Engine Landscape.
2	Sculpt terrain using Unreal Engine Terrain Splines.	Create a procedurally generated spline path and then apply this path to the landscape generated in objective 1. This spline path can later be used in objective 4 and 5.
3	Create a physics-based skiing character controller.	Create a physics-based skiing character controller that will ski down the procedural landscape.
4	Implement gameplay mechanics such as checkpoints, level start and finish and high score.	Add a start and finish line to the game as well as a checkpoints system to make a complete game. Also add a high score system that allows players to compete for the best time.
5	Add polish to the game e.g. GUI and decorative props	Add polish to the game by creating a main menu and a settings menu. Also procedurally add decorations to the scene to make it more interesting

Table 1 Project Objectives

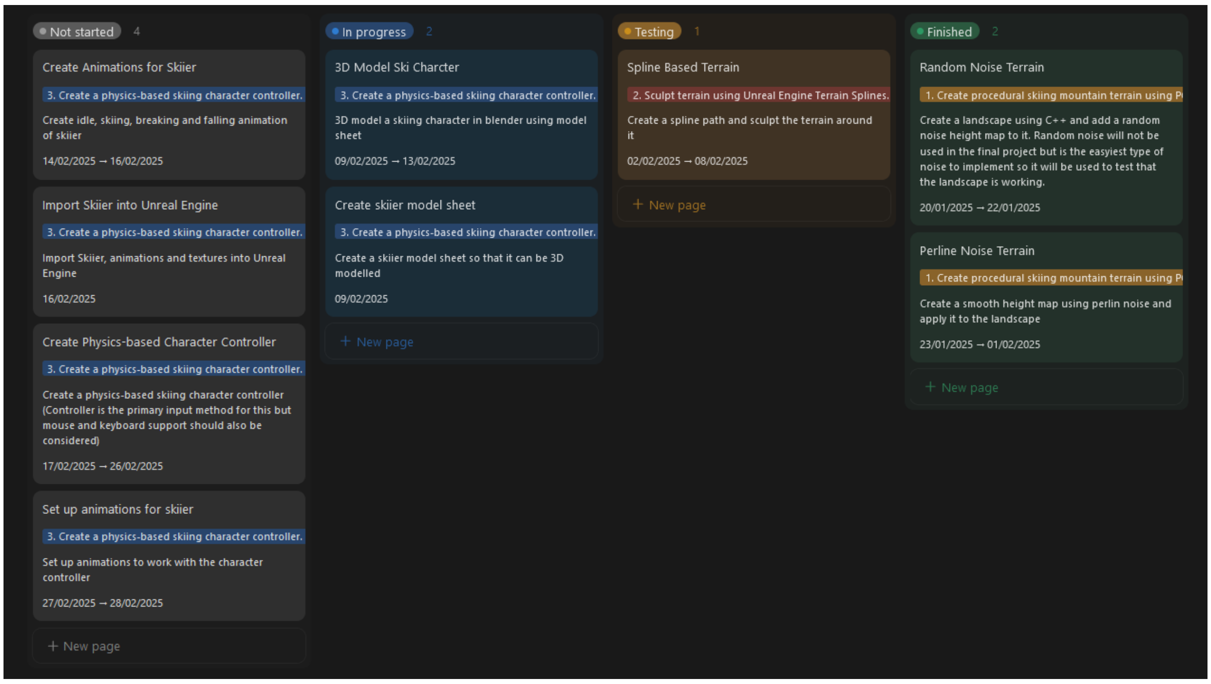
Methods

Development Methods

A software development model is an approach that dictates project workflow and how task are processed and completed in software development (Quiroz-Vázquez, 2024). The two most popular software development models are the Waterflow method and the Agile Method. According to Kivan, Lutkevich and Lewis (2024) the waterflow model is a linear approach to software development. This means that when using the waterflow method once a section of the work has been completed it cannot be changed. This means that once the design has been completed things in the design cannot be revised. Due to this the Agile method has been chosen for this project. Agile software development breaks up the project into smaller sections know as sprints (Quiroz-Vázquez, 2024) which allows developers to iterate on what they did in the last sprint by adding or editing tasks. Kanban is going to be used as the framework for implementing agile as it will allow the state of every task to be seen at any time (Radigan, N.d). The kanban board will be created using Notion (Notion

Labs, 2016). The tasks in the kanban board will be split into 3 different categories; not started, in progress, Testing and Finished. Additonally, on the Kanban board a description of the task, the date where the task should be worked on and the objective the task is forfilling will be displayed. A screen shot of the Kanban board is shown below.

Note that the tasks have been moved into the different status categories to show what the Kanban board will look like while the project is being completed. This does not reflect the current status of the project as it has not been started yet.



Not started 4

Create Animations for Skiier
3. Create a physics-based skiing character controller.
Create idle, skiing, breaking and falling animation of skiier
14/02/2025 → 16/02/2025

Import Skiier into Unreal Engine
3. Create a physics-based skiing character controller.
Import Skiier, animations and textures into Unreal Engine
16/02/2025

Create Physics-based Character Controller
3. Create a physics-based skiing character controller.
Create a physics-based skiing character controller (Controller is the primary input method for this but mouse and keyboard support should also be considered)
17/02/2025 → 26/02/2025

Set up animations for skiier
3. Create a physics-based skiing character controller.
Set up animations to work with the character controller
27/02/2025 → 28/02/2025

In progress 2

3D Model Ski Charcter
3. Create a physics-based skiing character controller.
3D model a skiing character in blender using model sheet
09/02/2025 → 13/02/2025

Create skiier model sheet
3. Create a physics-based skiing character controller.
Create a skiier model sheet so that it can be 3D modelled
09/02/2025

Testing 1

Spline Based Terrain
2. Sculpt terrain using Unreal Engine Terrain Splines.
Create a spline path and sculpt the terrain around it
02/02/2025 → 08/02/2025

Finished 2

Random Noise Terrain
1. Create procedural skiing mountain terrain using P
Create a landscape using C++ and add a random noise height map to it. Random noise will not be used in the final project but is the easyiest type of noise to implement so it will be used to test that the landscape is working.
20/01/2025 → 22/01/2025

Perline Noise Terrain
1. Create procedural skiing mountain terrain using P
Create a smooth height map using perlin noise and apply it to the landscape
23/01/2025 → 01/02/2025

Figure 2 Project Kanban Board

Testing

Testing is an important step in the development of software. It allows of bugs an errors to be identyfied (Magic Media, 2024), This will allow them to fixed. Due to a testing phase is of the states that a task can be in the projects Kanban board, due to this each task will be tested before it is marked as a finished task. During this testing phase performance testing will take place. This will test that the game is running fast an efficiently (Team EMB, 2024). The results of these tests can be shown using tables and graphs.

Additionally, there are multiple prototypes marked in the plan, these prototypes are after a section of tasks focusing on the same thing are finished. This prototyping phase will allow for relfection on how a section of the project has gone and allow of the plan to be changed if it needs to be (Simplilearn, 2024).

Data Gathering

Research papers will be used to decide which algorithms and methods to use in the project. Research papers will be used to create a literature review section in the project report to gain a better understanding of procedural content generation field. This is important as a literature review “can serve as the basis for knowledge development” (Snyder, 2019 339).

Resources Required for the Project

The project will be developed in C++ using Unreal Engine 5 (Epic Games, 1998). Unreal engine 5 (Epic Games, 1998) is a free to use for students (Epic Games, N.da) so there are no issues accessing it for the project. The minimum hardware requirements for Unreal Engine 5 (Epic Games, 1998)are shown in a table below as well as the hardware of the computer that will be used for this project.

	Recommended Hardware	Current Hardware
Operating System	Windows 10 64-bit version 1909 revision .1350 or higher, or versions 2004 and 20H2 revision .789 or higher	Windows 10
Processor	Quad-core Intel or AMD, 2.5 GHz or faster	Intel Six Core 2.9GHz
Video RAM	8GB VRAM	8GB VRAM
Graphics Card	DirectX 11 or 12 compatible graphics card with the latest drivers	DirectX 11 and 12 compatible.
Recommended Memory	32GB	32GB

Table 2 Unreal Engine Hardware Requirements (Epic Games, N.d.)

As shown in the table above the computer that will be used for the project meets the recommended hardware specifications of Unreal Engine 5 (Epic Games, 1998). Additionally, Blender (Blender Foundation, 1998) will be used for the 3D modelling assets in the project and Notion (Notion Labs, 2016) will be used as a CASE tool for project planning.

Legal, Social and Ethical considerations

A Social and Ethical consideration of procedural content generation is that because generating procedural content is faster a cheaper than it being created by a designer or artist (A. Summerville et al., 2018; Liu, J. et al., 2020), designers and artists may worry about being replaced by PCG algorithms. Designers and artists will not be replaced by PCG algorithms as PCG is designed as a tool to be used by designers and artists to create things that where not previously possible or to speed up the creation of assets for games (Epic Games, N.db).

The techniques chosen to procedurally generate the terrain in the project are noise and spline-based techniques. These methods have been chosen instead of machine learning based methods as a machine learning method needs vast quantities of data to produce good results for procedural content (Guzdial, Snodgrass and Summerville, 2022), this could open up the project to ethical issues as the data used for training the machine learning algorithm may not be intended to be used for AI. Additionally, it could open up the project to legal issues as there have been many lawsuits as a result of data being used as AI training data without permission such as Sony Music, Universal Music Group and Warner Records that claim that Suno and Udio have committed copyright infringement by using their music as training data for their generative AI models (Sherman, 2024). Due to this an approach that uses noise and spline-based techniques will be used instead.

Project Plan

To be able to achieve the objectives in the project they were broken down into 23 smaller tasks. These tasks are shown in the table below. Each task has a description of what it entails, they are labelled with what object or objectives they are working towards and what task it depends on being completed before it can be started.

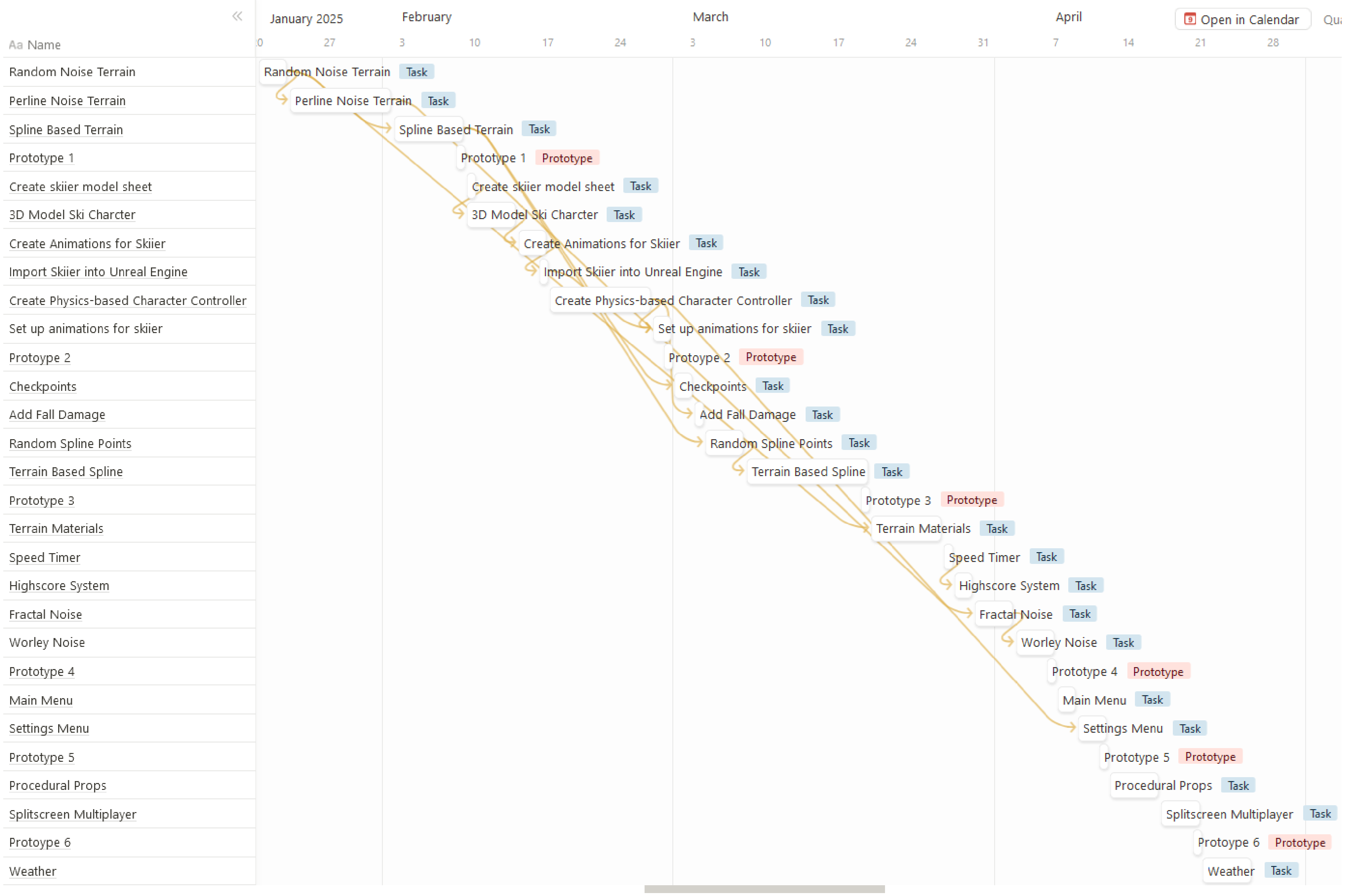
No	Name	Objectives	Description	Depends On
1	Random Noise Terrain	1	Create a landscape using C++ and add a random noise height map to it. Random noise will not be used in the final project but is the easiest type of noise to implement so it will be used to test that the landscape is working.	
2	Perlin Noise Terrain	1	Create a smooth height map using perlin noise and apply it to the landscape	Random Noise Terrain
3	Spline Based Terrain	2	Create a spline path and sculpt the terrain around it	Random Noise Terrain
4	Create skier model sheet	3	Create a skier model sheet so that it can be 3D modelled	
5	3D Model Ski Character	3	3D model a skiing character in blender using model sheet	Create skier model sheet
6	Create Animations for Skier	3	Create idle, skiing, breaking and falling animation of skier	3D Model Ski Character
7	Import Skier into Unreal Engine	3	Import Skier, animations and textures into Unreal Engine	Create Animations for Skier
8	Create Physics-based Character Controller	3	Create a physics-based skiing character controller (Controller is the primary input method for this but mouse and keyboard support should also be considered)	
9	Set up animations for skier	3	Set up animations to work with the character controller	Import Skier into Unreal Engine and Create Physics-based Character Controller
10	Checkpoints	4	Add a checkpoint system that uses the spline created in objective 2. It should also respawn the player on the most recent checkpoint on a button press or on death	Spline Based Terrain
11	Add Fall Damage	3 and 4	Add fall damage so that the player has to follow the paths	Create Physics-based Character Controller
12	Random Spline Points	2	Generate the spline based on random points	Spline Based Terrain

13	Terrain Based Spline	2	Procedurally generate the path based on the terrain heightmap	Random Spline Points
14	Terrain Materials	5	Use different terrain materials based on height and slope of the terrain	Random Noise Terrain
15	Speed Timer	4	Add the timer so that it appears in on an in-game HUD	
16	High score System	4	Add a timer and a saving system that saves the best time	Speed Timer
17	Fractal Noise	1	Improve the Perlin noise by implementing multiple perlin noise heightmaps	Perlin Noise Terrain
18	Worley Noise	1	Create a Worley Noise algorithm and add it as a layer of Fractal Noise	Fractal Noise
19	Main Menu	5	A main menu scene from the game. Should be able to close the game and open the settings menu	
20	Settings Menu	5	A settings menu into the game. This menu should include sensitivity settings, graphics settings and accessibility settings	Create Physics-based Character Controller
21	Procedural Props	5	Add drops like rocks and ramps	
22	Split screen Multiplayer	5	Add 2 player split screen multiplayer to the game	
23	Weather	5	Add weather effects to the game	

Table 3 Project Tasks

All of these tasks are reflected in in the Gant chart that was created to plan this task. Additionally, they are sorted in order of completion in table above. In this Gant chart there are also tasks labelled as prototypes these prototypes are placed after a different section of work has been completed e.g. creating the player character.

The CASE tool used to create this plan is Notion (Notion Labs, 2016). This is a free project planning tool that can be used in a browser or in a separate desktop or phone app. This app has been used as it can also be used to create a Kanban board that is linked to the Gant chart. An image of this project Gant Chart is shown below



Aa Name	
Random Noise Terrain	Task
Perline Noise Terrain	Task
Spline Based Terrain	Task
Prototype 1	Prototype
Create skiier model sheet	Task
3D Model Ski Charcter	Task
Create Animations for Skiier	Task
Import Skiier into Unreal Engine	Task
Create Physics-based Character Controller	Task
Set up animations for skiier	Task
Protoyope 2	Prototype
Checkpoints	Task
Add Fall Damage	Task
Random Spline Points	Task
Terrain Based Spline	Task
Prototype 3	Prototype
Terrain Materials	Task
Speed Timer	Task
Highscore System	Task
Fractal Noise	Task
Worley Noise	Task
Prototype 4	Prototype
Main Menu	Task
Settings Menu	Task
Prototype 5	Prototype
Procedural Props	Task
Splitscreen Multiplayer	Task
Protoyope 6	Prototype
Weather	Task

Figure 3 Project Gant Chart

Conclusions

In this document a skiing racing games that uses procedurally generated terrain to generate its levels has been proposed. Additionally, 5 objectives have been created that aim to achieve the aim of the project “Create a skiing game that uses procedural content generation (PCG) to generate the terrain and the levels. This project will use the Kanban Agile framework as its software development methodology. It will use C++ and Unreal Engine 5 (Epic Games, 1998) to create the project and Blender (Blender Foundation, 1998) will be used to create 3D assets in the project. Additionally, a project plan Gant Chart and Kanban board have been created using Notion (Notion Labs, 2016). This report has also taken into account the legal, social, and ethical considerations of the project.

Bibliography

A. SUMMERVILLE, S. SNODGRASS, M. GUZDIAL, C. HOLMGÅRD, A. K. HOOVER, A. ISAKSEN, A. NEALEN and J. TOGELIUS, 2018. Procedural Content Generation via Machine Learning (PCGML). *IEEE Transactions on Games*. 10 (3), pp. 257–270.

Blender Foundation., 1998. *Blender* https://www.blender.org: .

EPIC GAMES., N.d. *Hardware and Software Specifications [online].* Available from: https://dev.epicgames.com/documentation/en-us/unreal-engine/hardware-and-software-specifications-for-unreal-engine [Accessed 6 January 2025].

EPIC GAMES., N.da. *Frequently asked questions (FAQs) [online].* Available from: https://www.unrealengine.com/en-US/faq [Accessed 6 January 2025].

EPIC GAMES., N.db. *Procedural Content Generation Overview.* Available from: https://dev.epicgames.com/documentation/en-us/unreal-engine/procedural-content-generation-overview [Accessed 9 January 2025].

Epic Games., 1998. *Unreal Engine* Epic Games Launcher: .

F. AMATO and F. MOSCATO., 2017. Formal Procedural Content Generation in Games Driven by Social Analyses. *In: 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. pp. 674–679.

GOLUBEV, K., ZAGARSKIKH, A. and KARSAKOV, A., 2016. Dijkstra-based Terrain Generation Using Advanced Weight Functions. *Procedia Computer Science*. 101, pp. 152–160. Available from: https://www.sciencedirect.com/science/article/pii/S1877050916326862.

GUZDIAL, M., SNODGRASS, S. and SUMMERVILLE, A. J., 2022. An Introduction of ML Through PCG. In: M. GUZDIAL, S. SNODGRASS and A. J. SUMMERVILLE, eds. *Procedural Content Generation via Machine Learning: An Overview.* Cham: Springer International Publishing. pp. 23–33.

H. KIM, S. LEE, H. LEE, T. HAHN and S. KANG., 2019. Automatic Generation of Game Content using a Graph-based Wave Function Collapse Algorithm. *In: 2019 IEEE Conference on Games (CoG)*. pp. 1–4.

Hello Games., 2016. *No Man's Sky* Steam: .

JAIN, A., SHARMA, A. and RAJAN, K., S., 2024. Learning Based Infinite Terrain Generation with Level of Detailing. *In: 2024 International Conference on 3D Vision (3DV)*. pp. 1048–1058.

KIVAN, P., LUTKEVICH, B. and LEWIS, S., 2024. *What is a Waterfall model? Definition and guide.* Available from:

https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model [Accessed 6 January 2025].

LAGUE, S., 2016. *Procedural Landmass Generation (E01: Introduction) [online].* Available from: https://www.youtube.com/watch?v=wbpMiKiSKm8&list=PLFt_AvWsXl0eBW2EiBtl_sxmDtSgZBxB3 [Accessed 8 January 2025].

LIU, J., SNODGRASS, S., KHALIFA, • A., RISI, S., YANNAKAKIS, G. N. and TOGELIUS, • J., 2020. Deep learning for procedural content generation. *Neural Computing and Applications*. 33.

LIU, S., CHAORAN, L., YUE, L., HENG, M., XIAO, H., YIMING, S., LICONG, W., ZE, C., XIANGHAO, G., HENGTONG, L., YU, D. and QINTING, T., 2019. Automatic generation of tower defense levels using PCG New York, NY, USA: Association for Computing Machinery.

MAGIC MEDIA., 2024. *Game Testing 101: Basic Tips and Strategies [online].* Available from: https://starloopstudios.com/game-testing-101-tips-and-strategies/ [Accessed 5 January 2025].

Mojang., 2011. *Minecraft* PC: .

NOTION LABS., 2016. *Notion.* Available from: https://www.notion.com/ [Accessed 13 December 2024].

PECH, A., LAM, C. P. and MASEK, M., 2020. Quantifiable Isovist and Graph-Based Measures for Automatic Evaluation of Different Area Types in Virtual Terrain Generation. *IEEE Access*. 8.

QUIROZ-VÁZQUEZ, C., 2024. *What is software development [online].* Available from: https://www.ibm.com/think/topics/software-development [Accessed 6 January 2025].

RADIGAN, D., N.d. *Kanban [online].* Available from: https://www.atlassian.com/agile/kanban [Accessed 3 January 2025].

RageSquid., 2019. *Descenders* Steam: .

SHERMAN, N., 2024. *World's biggest music labels sue over AI copyright [online].* Available from: https://www.bbc.co.uk/news/articles/ckrrr8yelzvo [Accessed 9 January 2025].

SIMPLILEARN., 2024. *Prototyping In Design Thinking: Definition, Types & Benefits [online].* Available from: https://www.simplilearn.com/prototyping-in-design-thinking-article [Accessed 5 January 2025].

SNYDER, H., 2019. Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*. 104, pp. 333–339. Available from: https://www.sciencedirect.com/science/article/pii/S0148296319304564.

STÅLBERG, O., MEREDITH, R., KVALE, M. and Plausible Concept., 2018. *Bad North* Steam: .

System Era Softworks., 2016. *Astroneer* Steam: .

TEAM EMB., 2024. *Algorithm Testing: Tools and Best Practices [online].* Available from: https://blog.emb.global/algorithm-testing-tools/#5-performance-testing [Accessed 6 January 2025].

UNITY., N.d. *Mathf.PerlinNoise [online].* Available from: https://docs.unity3d.com/6000.0/Documentation/ScriptReference/Mathf.PerlinNoise.html [Accessed 7 January 2025].

Appendix

Approval for Low and Minimal Risk Research Projects

I can confirm that: ***(confirm you have read these)***

- **I have read the Departments Ethics Policy Document (see BB).**
- **I agree to abide by their principles**

Student	James Edwards
Date:	09/01/2025
Supervisor	Saad Saihi
Module leader	Ella Pereira