Luke Tolchard	Procedural City Generator - Project Log		
	Tasks set	Brief Summary of Outcome	Tasks to be taken forward
14/10/19	• Begin collecting research papers for report • Build an understanding as to how to begin prototyping	Set up the project space within Unity and found some relevant papers on procedural generation algorithms, still need to find more to make a conclusive decision	More research needs to be done to consolidate ideas of how to begin prototyping
21/10/19	• Discover some methods of procedural generation to try • Make a basic prototype of a system	Continued on with research this week, found some interesting papers that discuss Wave Function Collapse, and L-Systems. Still not in complete understanding of their role to play but it seems to be good way of trying to tackle this problem. I have also managed to get a simple camera switch script working, which will allow a user to view the plane from a range of different camera angles, which are fully customisable and easily extendable when the time comes to do that. https://gyazo.com/6a550d5b21f4cfce6d15f483ba76529e	Try and make a basic implementation of WFC or L-Systems to understand how they work
28/10/19	• Begin work on a prototype • Continue with PCG method research	Following a meeting with Sean I have turned my research attention to how cities are actually designed in real life to give an idea of good functionality for the project	Look into saving objects from play mode in Untiy
4/11/19	• Research saving game objects in play mode	As the city would be built in play mode, as soon as I close out of it the city would be lost. I've been looking into trying to find a function that allows for the grouping and saving assets but found nothing that I can replicate in a test project yet	Keep looking for ways of saving objects, and look at how to manipulate unity terrain via code
11/11/19	• Hold a focus group to ask my cohort what they would want to see in a project like this	As this will be a development tool, I thought it a good idea to ask my cohort for their opinions on what would be important to them for a Unity tool	Publish results in a blog post and begin to rethink research focus based on response
18/11/19	Did not complete any work due to illness	Did not complete any work due to illness	Did not complete any work due to illness
25/11/19	• Research into Perlin noise	I've seen some work that suggested Perlin noise is a good way of clumping similar values together, which could be used as a way of making districts of similar buildings in a city	Further explore validity to Perlin noise, try and prototype something with it
02/12/19	• Research into UI and user experience • Consolidate Perlin Noise research	Perlin noise seems like a very simple way of giving the effect I'm after, a simple prototype using a GetPixel function to visualise the noise on a texture looked great	Hold another focus group to ask about UI, and begin work on the research report
09/12/19	• Complete another focus group and record results in a blog post • Continue work on the research report	I have completed another focus group with the same set of students as before. This was mainly aimed at trying to show them the implemented changes from before based off their feedback. The minutes can be found here: gamedevelopment676677755.wordpress.com/2019/12/05/minutes-for-focus-group-3-12-	Publish the results on a blog post and continue with report
16/12/19	• Continue with report	This week I nearly feel ready to try and make a proper prototype city generator, I need to think about what settings I want to be in this menu, and how roads will interact with each other.	Start to create a settings menu using the window functionality and begin to construct a base road class

	• Think about what settings could define a road • Research best way of generating roads		
23/12/19	Week of due to Christmas	Week of due to Christmas	Week of due to Christmas
30/12/19	• Create a settings menu • Construct a base road class	The menu is available in the project now but the Unity GUI code is confusing and difficult to work with so it's taking some time setting it up how I want it. I guess it's more important to make sure its functional first and pretty later so I'll focus on importing some settings	Research road generation and make some settings for the menu
06/01/20	• Work on report • Research Road generation	The method of road generation seems to be a bit of a crossroad… I could choose to create models of road segments and have them procedurally selected from a wave function collapse algorithm but risk the city looking a bit rigid in how it appears or I can do something simpler like lots of line renders overlapping and only view from afar so the stitches aren't visible, but have a more fluid looking city	Decide a road generation method and write settings for it
12/01/20	• Finish research report • Apply settings to road generation	I've decided to use the line renderers as the January demo is fast approaching and I don't want to experiment with WFC this soon. I have finished the research report and have started applying some settings to the roads, I need a way of defining where the generation starts from instead of always at (0,0)	Create a start point function that lets the user choose where the city starts.
20/01/20	• Refine scope after Demo feedback • Complete additional research into building generation that's easier on the GPU	The demo went well, the roads generated okay but randomly selecting building prefabs and instantiating them was very costly and cause a bit of lag, need to see if there's another way of doing that. I also need to start looking at ways of moving the generation to a 3D terrain. City start location is nearly implemented, and I want to make it a physical object you move around the scene with a giant gizmo model like a flag or something that gives a clear indication where it is	Research cheaper ways of creating lots of primitives and create flag gizmo for start location implementation
27/01/20	• Test creating buildings using vertex declarations • Rethink Gizmo design	The flag gizmo looks great and makes it much easier to see where the start is, however it isn't really helping in regards to what direction the generation is going to start in so I may need to redesign it to a big floating arrow or something. Also found a unity doc about rendering boxes using vertex definitions to edit a game objects mesh filter. Might be cheaper on a GPU to create a new empty game object and then edit its mesh filter to show a box. Should be relatively simple to make the min and max size for each building change procedurally as well, so that kills 2 birds with 1 stone	Test different road types deriving from base class, think about how roads need to interact with each other.
03/02/20	• Develop a system for road intersections	If roads branch out over the top of each other it looks really messy and horrible, I need to develop a function that kills the road generator if its going to come into contact with another road. But I don't want that to happen all the time as then no roads will ever cross and there'll be no junctions. Maybe a % chance to intersect, or stop just before	Experiment further with road intersections
10/02/20	• Split road type into 3 different roads	It's occurred to me that there are many different types of roads that have different functionality in	Develop menu further to make the GUI look better

	with individual properties • Update settings menu to reflect the change in road structure • Develop a system for road intersections	the real world. I don't want houses spawning next to motorways, and I don't want streets stretching on for miles and miles before stopping. Easiest way to stop this is to make sub classes for each type of road, and make more tabs on the menu to contain them all (Makes the menu look more important as well)	and carry on working on intersections
17/02/20	• Make the heightmap editable on the menu • Implement intersections	Added a texture map slot on the menu for the heightmap so the user doesn't have to use the inspector. I've found a way round the intersection problem by making the branch generator check the terrain for something it can intersect with, and if its going to hit something it deletes the last road segment when it collides, then it redraws to the road. The calculation is costly when every branch is doing the same though	Optimise the intersection calculation
24/02/20	• Redesign road generation classes to allow for 3D terrain implementation • Optimise intersections	I may have made a mistake by leaving the upgrade to 3D generation so late, the code is now difficult to rewrite so that it works vertically. I did manage to optimise intersections though by dividing the terrain into quads so a road only has to check for intersections within its quad rather than the whole map.	Keep working on 3D implementation
02/03/20	Didn't complete any work due to coronavirus related issues	Didn't complete any work due to coronavirus related issues	Didn't complete any work due to coronavirus related issues
09/03/20	Didn't complete any work due to coronavirus related issues	Didn't complete any work due to coronavirus related issues	Didn't complete any work due to coronavirus related issues
16/03/20	Didn't complete any work due to coronavirus related issues	Didn't complete any work due to coronavirus related issues	Didn't complete any work due to coronavirus related issues
23/03/20	Did not complete any work due to mental health issues	Did not complete any work due to mental health issues	Did not complete any work due to mental health issues
30/03/20	• Push for 3D implementation	As I've been sick and unable to work for several weeks I am now very behind schedule on most modules so do not have a lot of time to dedicate to this problem right now. Have acted on the contingency stated in my proposal and reduced all hours at my part time job to help me catch up and figure this implementation out	Keep working on 3D implementation
06/04/20	Did not complete any work due to mental health issues	Did not complete any work due to mental health issues	Did not complete any work due to mental health issues
13/04/20	• Keep working on 3D implementation	I don't have time to rewrite the road generation code, but I feel that I may have to in order to convert this project over to 3D terrains rather than 2D planes. Might have to draw up compromises on the final implementation In order to meet the deadline	Keep working on 3D implementation
20/04/20	• Prepare for resubmission of CTP	I no longer have time to finish this in time for the deadline due to my health issues, I do have a chance to finish my other modules however, so I will be shelving this project until the resubmission period begins where I shall go again and work hard to finish this as well as I can	-

RESIT PERIOD			
22/06/20	• Redesign camera system • Refresh memory on project and code	It has been a long time since I've opened this project, so I'm going to spend a week to refamiliarise myself with how it works and what I need to do to convert it to 3D. The camera system is currently annoying to use so I shall break myself back in gently by redesigning it to be more intuitive.	Begin working on a solution to the issue
29/06/20	• Develop a ray cast system for detecting terrain height	Its apparent to me that attempts to have the roads created at the correct height have all ended in failure, which has given me the idea to try and edit their height after all the generation is finished. That way I no longer have to try and back pedal on the code that is already in place, I just have to develop a new system that plugs in to a new road's x,z position and then adjusts its y.	Develop a formula for looping over a terrain and storing values in a list for retrieval
06/07/20	• Implement excel implementation to code	I've made a game object that loops over every pixel of the terrain before generation begins that shoots a ray cast straight down and records the distance to the terrain in a list. When a road created at that pixel needs to know the height it needs to be bumped to, it needs to access the right value in the list and adjust accordingly	Optimise that function in C#
13/07/20	• Bug fix the height tester	The height tester sort of works, but nowhere near well enough for use in the project. With a terrain of 1000x1000, unity freezes for approx. 10 seconds at the start of each generation as it stores the height values in the list. The values are also lacking precision, most are out of the true value by about +/- 10%. Research online suggests that Unity makes ray cast results imprecise under heavy duress in an effort to save on computation, so I suspect that dumping this workload on the system is causing it to immediately break. To intentionally slow it down even more so that it is more precise makes the system not worth using. It's back to the drawing board.	Contemplate more ways the 3D terrain system might work.
20/07/20	• Try and read a GetPixel value off the heightmap	I was trying to make it so much harder than it needed to be, and have resorted to using GetPixel to do the same job as the ray caster, just at a fraction of the computational effort. I do not know how to assign it to the roads however as the roads themselves are created at 0,0 of their 'parent' which is a generator prefab that is immediately deleted after its child is created.	Will have to begin the final report write up
27/07/20	• Work on writing the final report • Create showcase video • Create control readme	I have begun writing the final writeup report, it's a shame I didn't manage to get the project to where I wanted it to be but there is a lot in there that I was able to overcome and develop so I am proud of the implementation I will be presenting	Submit!
03/07/20	SUBMISSION DATE	SUBMISSION DATE	SUBMISSION DATE