# **INFO6028 – Graphics 1 - Final Exam – Fall 2021**

Instructor: Michael Feeney

## The exam format:

* You may use any resources you feel are necessary to complete the exam, but you are to answer the questions **on your own**. I will be looking for plagiarism (i.e. copying) very carefully. There is *no possible way* that the specific code to answer these questions, or the output to the screen, would be very similar to the look of another student’s code. Remember, this is a test and there are very clear policies about cheating on tests.   
  + <http://www.fanshawec.ca/admissions/registrars-office/policies/cheating-policy>
  + <http://www.fanshawec.ca/sites/default/files/assets/Ombuds/cheating_flowchart.pdf>
* It is an “open book” exam. You have access to anything you book or internet resource you’d like
* The questions are ***NOT*** of equal weight. The exam has **eight (8)** questions and **nine (9)** pages. The questions involve submitting a working Visual Studio solution.

|  |
| --- |
| * The questions build on each other, to make a complete scene. However, you may decide that a different camera angle will better show the scene, so:   + You may submit a single solution & project if you thinks it’s appropriate   + If you submit multiple solutions and/or projects, please name them in some easy to understand way like “question\_1”, etc.   + Please include any information I’ll need (keyboard controls, etc.) in a readme file, and not buried in the source code somewhere. |

* I should not be expected to alter your code in any way to get it to build or run, so do not comment out your code, or expect me to alter it in any way at all. It should be ready to run as is.
* Your code should run under Windows 10 (i.e. a “win32 application”) with Release and x64 (“64 bit”) build.   
  I will *not* be building a “32 bit” application (note: “win32” means “windows API” and *not* “32 bit”).
* **If the solution does not build (and run), I will not mark it** (so you will receive zero on questions that can't be built and/or won't run). When I say "run", I'm not speaking about some, random, unforeseen bug, but rather something that you should have obviously dealt with, like memory exceptions, etc.
* No “**auto**” or the **boost library** (or any other 3rd party library we haven’t used or you haven’t cleared with me first).   
  If you use either, I will not mark your submission and you will receive a mark of zero.  
  No exceptions.

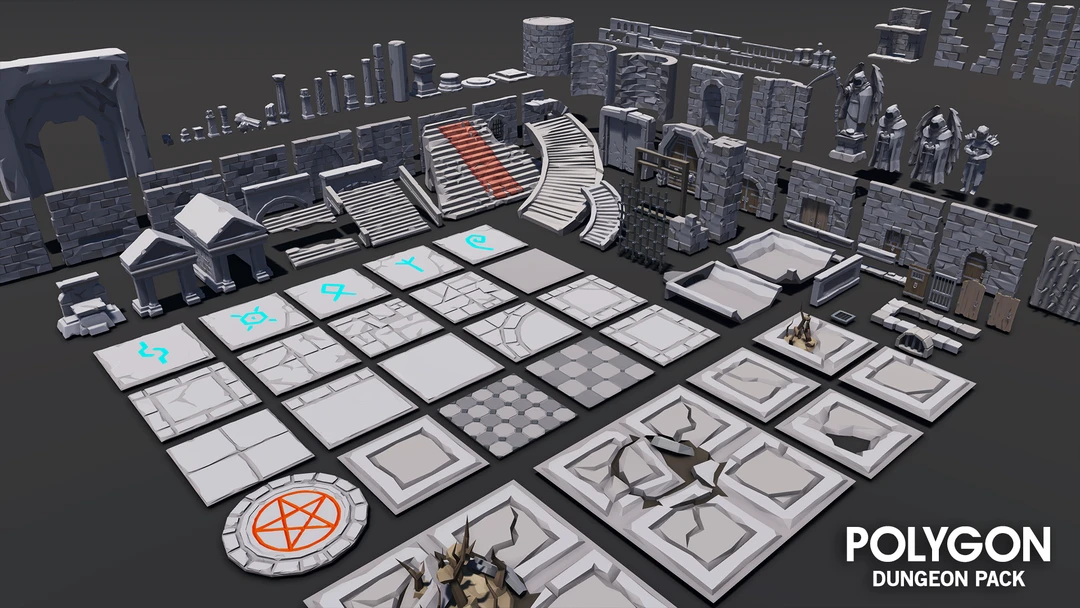
## NOTE: Unless otherwise indicated, you should be displaying the ply files that are in the Dungeon\_models folder.

## The Questions:

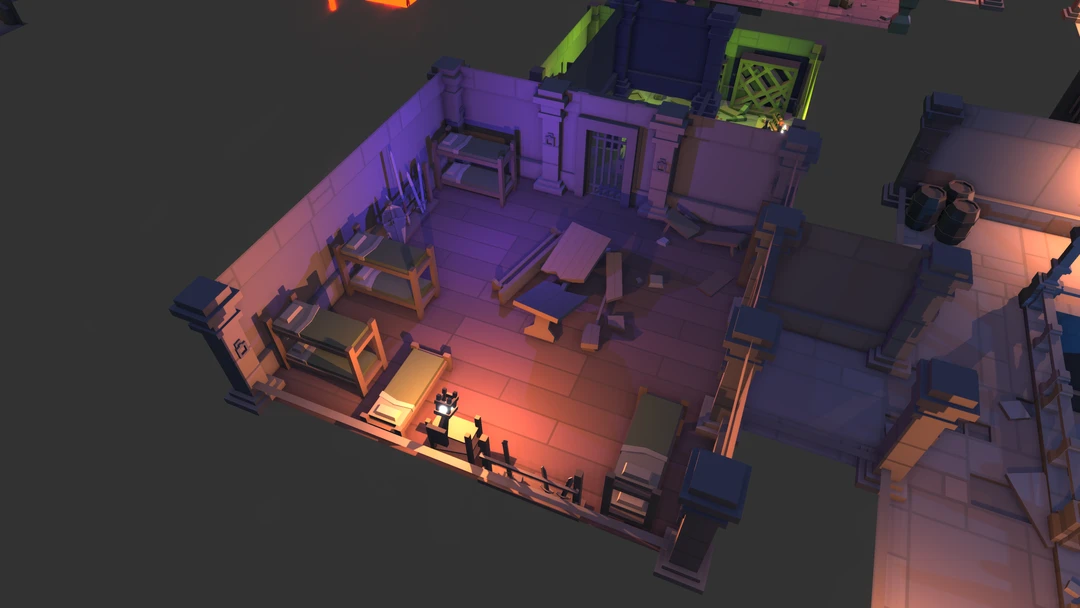
You are to create a small dungeon made up a several large open spaces and smaller corridors.

You will use the assets taken from the Synty Studios “POLYGON - Dungeon Realms” (<https://syntystore.com/products/polygon-dungeon-pack>) and “POLYGON - Pirate Pack” (<https://syntystore.com/products/polygon-pirate-pack>).

Many of the “environment” assets can be combined like LEGO bricks into any shape you’d like. They are all of similar size. In the picture below, you can see there’s a bunch of different “floor” models as well as a number of “wall” models.



When combined, you can make rooms like the one below. Also note that the small corridor is also made of floor and wall models:



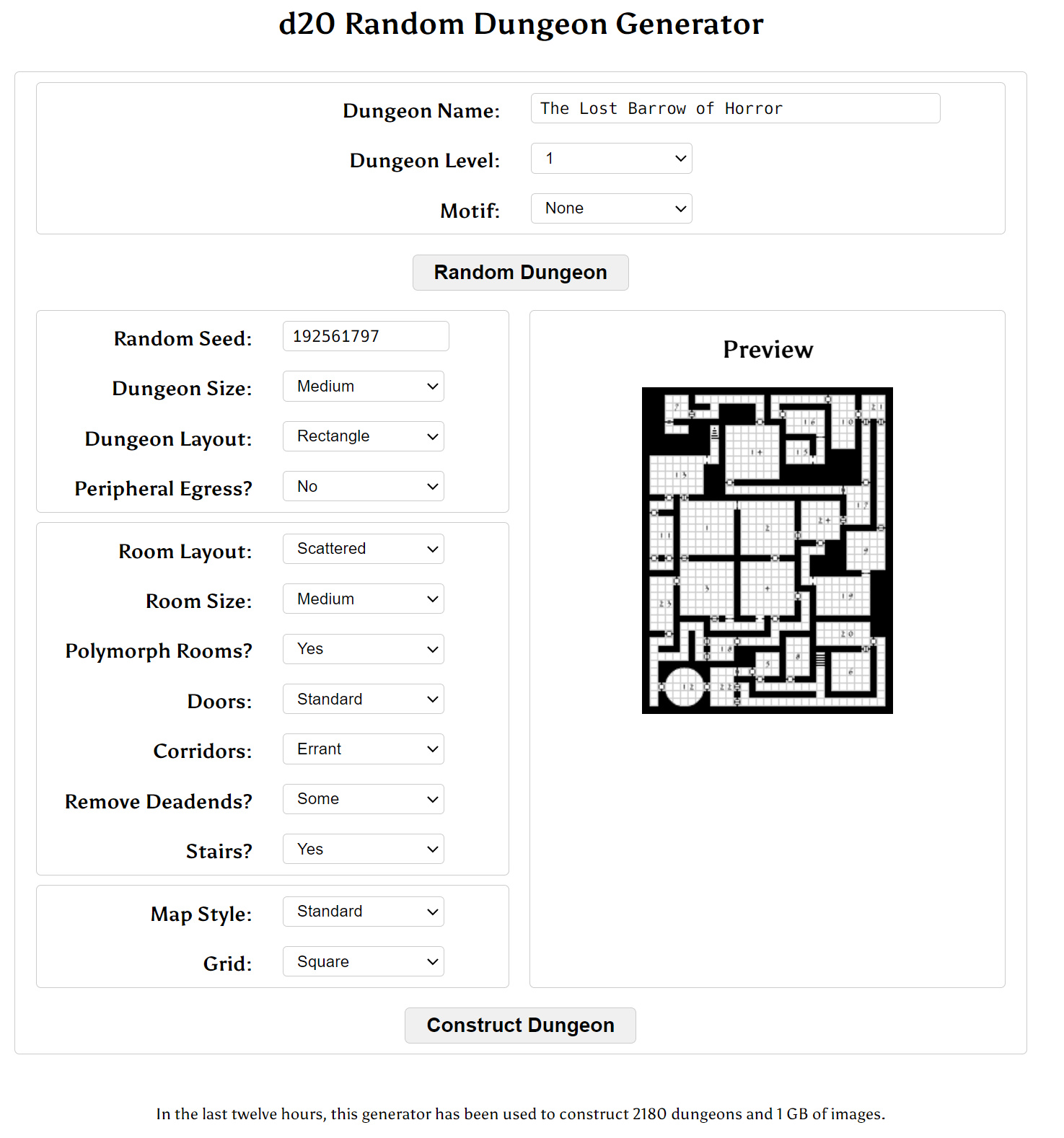
**Some notes about the models:**

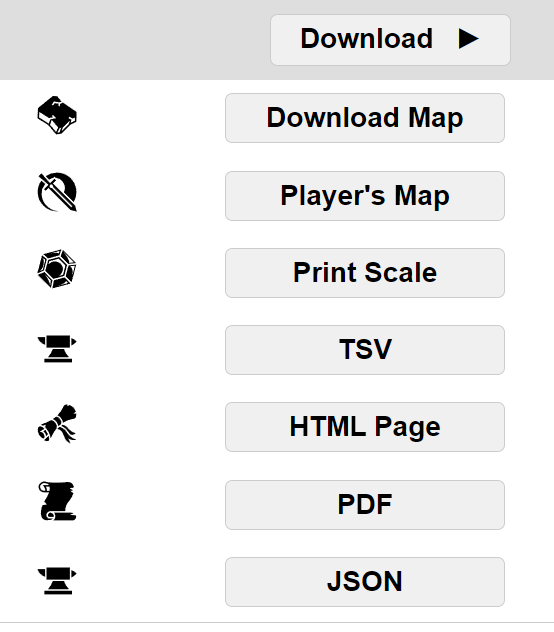
* They are aligned in one corner of the model. This allows you to “snap” them together more easily (MeshLab “Render”, then “Show Axis” will show this.)
* The textures that correspond to the model are shown in the ply file, for example, the SM\_Env\_Floor\_Grate\_01.ply file has the following:  
    
   comment TextureFile Dungeons\_2\_Texture\_01\_A.png   
    
  This means that you need the Dungeons\_2\_Texture\_01\_A.png file as a texture.
* I’m pretty sure I’ve converted them all to an xyz+normal+rgba+uv format, but there’s 100s of files in there; if I haven’t, then you should be expected to convert and/or load them yourself.

**You are going to make a dungeon based on the output of donjon website, specifically:**

[**https://donjon.bin.sh/d20/dungeon/**](https://donjon.bin.sh/d20/dungeon/)

The site should default to a “medium”, “rectangle” sized dungeon, but just in case it doesn’t, you can use the following settings:



1. (5 marks) Generate your own dungeon + accompanying files.

* Type your student number into the “Random Seed” text box.
* Hit enter (or click anywhere) then press the “Construct Dungeon” button at the bottom.
* It will generate a TON of stuff (which is pretty neat), as well as a set of files that you’ll need to submit.
* At the bottom of the screen, you’ll see a “Download” button. Click it and download all seven (7) files.
* **YOU WILL NEED TO SUBMIT ALL OF THESE FILES**.

1. (150 marks) Create the basic layout of the dungeon from the TSV file.

* Here’s an example of a TSV file and the matching image/map file.

F F F F F F F F F SD SDD

F

F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F

F F F D F F

F F F F F F F F F F F F F F F F F F F F F F F

D F F F F F F F F F F F F F F

F F F D F F F F F F F F F F F F F F F F F F F D F F F

F F F F F F F F F F F F F F F F F

F F F F F F F F F F F F F F F F F F F F F F F F F F F F

D F F F F F F F F F F F F F

F F F F F F F F F F F F F F F F F F F F F F F F

F F F F F F F F F F F F F F F F F F F F

F F F F F D F F F F F F F F F F F D F F F F F F F

F D F D D

F F F F F F F F F F F F F F F F F F F F F F F F F

F F F F F F F F F F F F F F F F F F F

F F F F F F F F F F F F F SUU SU F F D F F F F F

F F F F F F F F F F F F F F F F F F F F

F F F F F F F F F F F F F F F F F F F F F F F

F F D F F

F F F F F F F F F F F F F F D F F F F

F F F F F F F F F F F F F F F F F

F D F F F F F F F F F F F F F F F F

F F F F F F F F F F D

F F F D F F F F F F F F F F F F F F F F F F F F F F F

F F F F F F F F F D F F F F F F F F

F F F F F F F F F F F F F F F F F F F F D F F F F F D F F F

F D F F F F F F F F F F F F F

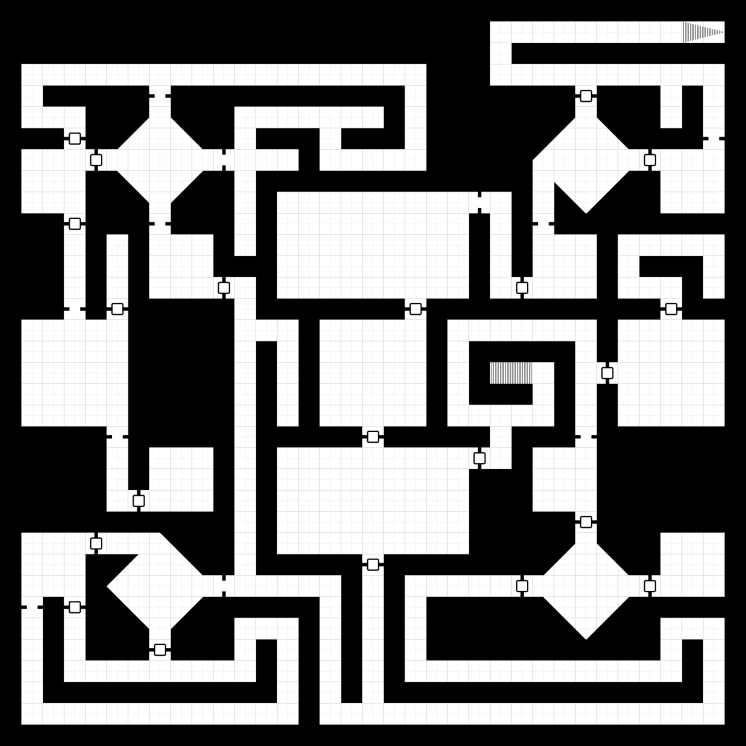
F F F F F F F F F F F F F F F F F F F F F

F F D F F F F F F F

F F F F F F F F F F F F F F F F F F F F F F F F F F F

F F F F F

F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F

****  
You’ll notice a few things about the TSV file:

* It’s on a grid, like the image map.
* The “F” indicates a corridor (white parts of the map) and everything else is a wall/rock/whatever (black parts).
* There’s a “D” where there’s a door.
* The “S” indicates a stair, with “U” meaning “up” and “D” meaning “down”. Note that the stairs take two “cells” on the grid.

----------------------------------

-----------------------FFFFFFFFF34

-----------------------F----------

-FFFFFFFFFFFFFFFFFFF---FFFFFFFFFFF

-F-----F-----------F-------D---F-F

-FFF-FFFFF-FFFFFFF-F-----FFFFF-F-F

---D-FFFFF-F---F---F-----FFFFF---F

-FFFDFFFFFFFFF-FFFFF-----FFFFFDFFF

-FFF-FFFFF-F-------------FFFFF-FFF

-FFF-FFFFF-F-FFFFFFFFFFF-FFFFF-FFF

---D---F---F-FFFFFFFFF-F-F--------

---F-F-FFF-F-FFFFFFFFF-F-FFF-FFFFF

---F-F-FFF---FFFFFFFFF-F-FFF-F---F

---F-F-FFFDF-FFFFFFFFF-FDFFF-FFF-F

---F-D-----F-------D-----------D--

-FFFFF-----FFF-FFFFF-FFFFFFF-FFFFF

-FFFFF-----F-F-FFFFF-F-----F-FFFFF

-FFFFF-----F-F-FFFFF-F-21F-FDFFFFF

-FFFFF-----F-F-FFFFF-F---F-F-FFFFF

-FFFFF-----F-F-FFFFF-FFFFF-F-FFFFF

-----F-----F-----D-----F---F------

-----F-FFF-F-FFFFFFFFFDF-FFF------

-----F-FFF-F-FFFFFFFFF---FFF------

-----FDFFF-F-FFFFFFFFF---FFF------

-----------F-FFFFFFFFF-----D------

-FFFDFFFFF-F-FFFFFFFFF---FFFFF-FFF

-FFF-FFFFF-F-----D-------FFFFF-FFF

-FFF-FFFFFFFFFFF-F-FFFFFDFFFFFDFFF

-F-D-FFFFF-----F-F-F-----FFFFF----

-F-F-FFFFF-FFF-F-F-F-----FFFFF-FFF

-F-F---D---F-F-F-F-F-----------F-F

-F-FFFFFFFFF-F-F-F-FFFFFFFFFFFFF-F

-F-----------F-F-F---------------F

-FFFFFFFFFFFFF-FFFFFFFFFFFFFFFFFFF

----------------------------------

* The file is tab (not space) delimited, and is a little odd in that it’s also using the tabs to indicate empty locations on the grid.

If you **don’t** want to just load that file as is, here’s what I would suggest you do to use this file:

* Open your TSV file in notepad++ and do find an replaces in this order:
  1. Make the tab symbols visible with “View”, then “Show Symbols”, then “Show whitespace and tab”.
  2. Select on of the “F” and the tab characters (the arrow). Copy it.
  3. Choose Repace, pasting that (the F+tab) and replace with “F”.
  4. Do the same for the “D+tab”, the “SU+tab”, etc.
  5. When you are all done, do a find and replace on the tab characters only, replacing them with something else, like a “-“ character. When you’re done, you’ll have something like this

1. (...continued...) Make the basic layout of the walls and floors using this “map”

* At this point, you *don’t* need the stairs (they happen later).
* You also don’t need to worry about the doors, yet (they happen later, too).
* You can use any “floor” and “wall” models you would like providing:
* There is some variation in the walls and floors. Like every room can’t have exactly the same floors and/or walls. They also can’t be just random – they should look like someone made them.
* The walls should be the same. You have some discretion about this, like the hallways being different than the walls, etc.
* Add a keyboard/menu control to switch *the entire scene* from solid to wireframe and back.   
  (So that I can see the entire scene)

1. (150 marks) Place a number of lit “torches” in the scene:

* Using the “torch” (light) model in the “Torches” folder, add some wall mounted torches.
* Place these torch models spread out evenly in the scene, attached to the walls, near the top. The idea is that they are torches with fire or candles, attached near the top of the wall.
* Place small, bright red-orange lights where the fire/candle of the torches would be.   
  These should light up the area nearby (like a small candle or flame would).
* Make the torches “flicker” by slightly perturbing the linear attenuation each from.   
  You do this by picking a small random number and adding it to the “base” attenuation of the light. In other words, the “base” would be the attenuation without flickering, then each frame you are adding a slightly larger number, making it slightly different per frame.
* There should be a “reasonable” number of torches so that it’s believable to see what’s going on.   
  So if there’s a long corridor *without* torches, how could you see anything, right?
* Remember: like question 2, you can do this procedurally or semi-procedurally (like place most of the torches automatically, and add a few manually).   
    
  For instance, torches can be placed at “corners” of rooms/hallways as long as they are “far enough” from each other. You could go through the entire map, looking for “corners” that don’t have torches yet, then see if there’s a torch “too close” – if not, place one there. Do this a number of times, like 100 times or whatever, and you’ll likely have a good spread.   
    
  And, of course, you can place them manually – the thing is on a grid after all.

1. (150 marks): Add the doors **and stairs**.

* The site will generate all sorts of doors, but you only have to deal with certain ones:
* Archways (with or without doors – your choice)
* Decide if they are opened or closed
* The “portcullis” is one of those doors that open vertically
* You can ignore the “locked”, “trapped”, and “secret” types.
* (BONUS 50 marks: Assign the number keys to specific doors so they open. Regular doors should “rotate” around their vertical hinges, but portcullis doors should “slide” up and down. Pressing a button should open the door. Pressing again should close.)
* Choose whatever door model you’d like.
* **Add ascending and descending stairs. Use whatever “stair” model you’d like.**

1. (150 marks): Place several semi-transparent, glowing crystals in the scene.   
   * Place at least five (5) of the “crystal” models (in the “Crystals” folder) throughout the scene. These should look like they have randomly been scattered or dropped or were just naturally there. Like they shouldn’t be dropped together or in a line, etc.
   * Make the crystals semi-transparent (so you can see through them).
   * Place small lights inside the crystals so they look like they are “glowing” slightly.  
     Note that you might have to change the “emissive” colour.
   * The transparency should be “order independent”, in that I can move the camera around and see all the crystals through all the other crystals.
   * Make the crystals very “shiny” (i.e. have a very high specular component).
2. (150 marks): Place a few plants that are growing in “holes” in the ceiling.   
   * Place three (3) groupings of plants in three (3) different rooms.
   * Each grouping consists of a single tree with a few smaller plants around them.
   * To mimic the “holes in the ceiling” by placing spot lights shining down on these plant groupings.
   * The spotlights should be large enough to light up all the plants in each grouping.
   * Point the spotlights at various angles, so that the light on the floor should be elliptical in shape (rather than circular, which it would be if the spots were *directly* above).
   * They should also be different elliptical shapes.
   * You can do this by *not* placing the spot lights directly above the plants, but having the spot direction pointing at the plants.
   * Use the FX\_SunShafts.ply model to mimic a “shaft of light” following the path of the spot light. Choose an appropriate texture for that.
3. (100 marks): Mimic the time of day changing.   
   * Gradually move the spot lights from one side of the mode to another.
   * When they get to the one side, gradually dim them (mimicking evening).
   * After this, make the overall lighting dimmer but *don’t* reduce the torch lights.   
     This will enhance the light from the torches.
   * After a little while, place the spot lights on the *other* side other model, gradually increase the overall lighting, and start moving the spot lights again.
4. (100 mark **bonus**): Mimic the ages passing with moss growing.   
   * Chose a “moss” texture from the internet.
   * Also choose a “vine covered wall texture” (like google that phrase).
   * Based on a key-press, mimic many years passing by doing the following:
     + The moss should slowly “creep” across the floor, starting from the centre of the rooms\*\* to the edges. You can do this by:
       - Picking a point in the centre of the room (pass this as a uniform), and sample the moss within a circle, very gradually expanding the radius. In other words, the “circle of moss” is applied on top of the floor.
       - If you don’t want to mess with the texture coordinates of the floor tiles, you can place a large quad, covering the entire map, but *just* above the floor, and apply the moss texture to this. It should be close enough you can’t quite tell it’s a separate mesh above the floor.
     + The vines should start to “grow” (i.e. appear) from the floor to the ceiling. You can do this by gradually applying a static texture to the existing walls, and drawing more of the vines over time (i.e. the “top” of the vines – where you sample – gets higher over time)

* + - \*\* ideally *all* the rooms, but it depends on your dungeon – it should be “most” of the “larger” rooms, though. Likewise ideally it should be “all” the walls, but it should be most of them – enough to give a reasonable effect.

**That’s it.**