|  |  |
| --- | --- |
| Harry Luck - Design Test |  |
|  | Senior Level Designer |

**Question C - Additional Information**

So, I decided I wasn't happy with the overall structure and the way I presented my answer to this question and I also feel like there might be a slightly better, and more organic, way to design and implement an automated level creation tool.

First of all, I believe the biggest step I didn't go into enough detail about was how such a system might initially generate content. How would this be achieved? What constraints would need to be in place and how would the user decide how the level would be constructed. So my proposal would be content blocks.

Content blocks would essentially be a large macro scale flowgraph based system that describes the base zones the level would require and how they would ideally be linked together to describe the general level layout and connections for the automated system to adhere to. Here is an example flowgraph:



 **Lets take a closer look at some of these modules and how and what could be used to determine their structure and layout with a few more details.**

**Transition Corridor**



**By using a system like this you could control how these areas were connected, add extra connections if necessary, select the assets, or possibly asset groups, that should be used to generate the area as well as define sizes for the area. There could, and likely should, be many more options and constraints within here, some of which could be dynamically generated based on how the flowgraph is constructed initially or altered after the a simple layout has been generated.**

**Another thought I began to consider was based on how to prevent very angular and/or 2D layouts from being the end result of auto-generation using a set of modular geometry assets. So, I realised there might be the possibility of using spline based deformation on modular sections after their initial construction. See below for an example of how this might work:**



**By using the same modular assets as before to create a section but then adding a unique spline for deformation you can alter the possible variations of corridor like sections by a huge amount. As long as the start and end of these deformations remain as possible connections to subsequent sections then this could be an excellent way of reducing the similarity of each area.**

**Gameplay elements that are related to the actual layout and geometry pieces should, after some more thought, actually be clamps that are placed within the assets themselves, either prior to export from a 3D package or added as elements in another tool. For example, wooden supports could have climb clamps and swing clamps added to them. By clamps, I simply mean some kind of dummy object that is linked and placed into position onto the actual asset. The clamp itself would contain the information needed to determine how it functions and what it is. This could be aligned and placed in multiple locations on the actual asset. Then, when the level is being generated constraints about how far apart or how often those assets could be used, if the area is to consider gameplay elements, would be considered based on the clamps that are attached to the assets. For example:**



**Here we have included several possible locations for vertical swings, a horizontal swing around the column and an area to climb.**

**Other gameplay elements, like items, pickups (health, ammo, weapons etc) etc could simply have a set of simple rules that determine how many can be placed as possible locations in a given area, how close together than can be to one another and how many should actually end up being spawned in that area from the possible locations. This could also be setup through the flowgraph system as a separate set of dialogs.**

**I hope this is a good addition to the previous section I wrote on Question C. I just knew there was more I wanted to add and explain and I hope I have done that a little more here. Obviously this is a huge technical task that I alone would be practically incapable of designing without collaboration from programmers and artists. I just wanted to share some more thoughts.**