* **Level Analysis**

The Input level is divided in GameObjects. The user can tag those objects (many types). Furthermore, objects are inside areas, defined by invisible Cubes or other Geometry types, so as to analyze data inside: how many times the controller died in each area, how areas are traversed...

* **Controller AI**

The controller has a set of parameters that can be inputted by the user.

Each controller reports the data in a separate file.

* **Visualization**

Said controller does not have to be a Unity 3D controller but rather an abstract controller symbolized by UI.

The 3D level can actually be “converted” to a scheme that makes it more accessible for visualization:



*Doom Eternal*

A navigation mesh would be ideal to achieve a first approach towards the above layout. The user would be able to see the player moving around and performing actions.

* **Automation**

Theoretically, there should be a different output window for each test that is being run and/or capture and save each simulation in a video.

If there is parallel execution, the controller must be able to try out different actions, otherwise it would be the same test. This goal can be accomplished by having some random factors and/or varying particular variables in the level or the controller for each test.

* **Feedback**

Each controller will have by this point generated a report and the program will calculate averages, medians....

If we alter the controller player type and data in each parallel simulation or between simulations, we can achieve more player-based conclusions, whilst if we vary some level data, eg by having an algorithm that rearranges pre-existing items, we achieve more level-based data.

**Useful Links**

[Editor Windows](https://docs.unity3d.com/Manual/editor-EditorWindows.html)

[How to make a Minimap in Unity](https://www.youtube.com/watch?v=28JTTXqMvOU)

[Unity Recorder User Manual | Package Manager UI website](https://docs.unity3d.com/Packages/com.unity.recorder@2.0/manual/index.html)