The main functioning of the generated level as discussed in the requirements is to be:

- infinite
- getting harder as the game progresses
- procedural

In this short text it is discussed if the current implementation is optimal to extend to these requirements.

In short a level consists of a queue of **Gameslice**s that the player must traverse. These **Gameslice**s are put in the queue by the **Director**. In the director logic can be put what slices would be interesting to put in next.

The things that I find good currently is that there is a director class that is able to randomely do things.
The things that I find can be better is that the director class does too much. The director should put gameslices into place, but the gameslices should decide for themselves how they configure themselves. The gameslices are actors that have to work together to create a coherent whole. So then there should be a few different gameslices that configure themselves differently have different elements or configure them in different ways. One gameslice can be about a climactic experience, a few little hops following a big jump, an other gameslice can be about just some random blocks etc. To allow for these different things Gameslice should have children that describe each one of such interesting elements. And very importantly they decide themselves their configuration, and talk with the gameslice that is closer to the player what they are able to be for the player to be able to progress.

The way I would like it to be: