Changed the way to structure the level

By building up the level using segments, we can randomly generate the next part of the level and connect it to the previous segment. This creates an illusion of an infinite level with different terrains.
We then divide the segments into slices which are randomly selected to give a feeling that even though the terrain is the same, it won't appear as the same terrain.

Changed texture atlas of a Segment

The issue at hand is that:
1. The texture size that can reside in the GPU is limited in older phones.
2. The most time-consuming process in the game loop is switching textures on the GPU, this is solved by having many images on the same texture and just drawing the portions we need.

We decided to change the way we structure the textures of the background. At the start we had slices which were rendered one by one to cover the whole screen. This would allow us to have many segments on a single draw call to the GPU which is very efficient.
This however didn't give us the quality for the rest of the level we require. We therefore chose to split the texture in rectangles more representing a landscape screen which is then scaled up to fit the size it's going to be used for.

We still don't know if it will be of sufficient quality if the phone screen has a higher resolution but it's a start and works for all phones.

Buildbot

Travis building bot compiles all the different branches every push and runs the unit tests on that branch. This makes testing simple and automatic which makes sure people don't forget.

Enums as definition

As a general rule in the system design we have enums used as definitions for concrete classes, if there exist several setups, with different properties but with the same behaviour, of the same class. These enums describes the properties of an object and are able construct such an object from it's properties.

For example: a class Enemy could have different amount of life points, different weapons and so on. Instead of creating different classes (For example Eenemy1, Enemy2), we have defined these properties in a definition enum so that the enum could create an Enemy with high life points and a special weapon.

These definition enums implements an interface so that they could be replaced by some other means of data definition, for example a specialized implementation, a json document or a remote database.