How to build a level in „Ooops! It’s Alive“

First of all open the „BasicLevel“ scene to see the basic structure of a level. To create a new one just save this under a new name.

# 1. Design the level layout

Since one of the main features of this game, is the interaction between the cells and the light (if photosynthesis is activated they grow in lighter and shrink in darker areas), you should first draw your light texture. This can be done in paint or any other image editor. The texture should have a size of 64 x 64 pixels. After this is done enable Read/Write in the texture options. In the „level“ object select your texture as the map component and choose a second background color in „Bg Color“. By pressing „Generate Lightmap“ the lightmap is displayed in the scene editor.

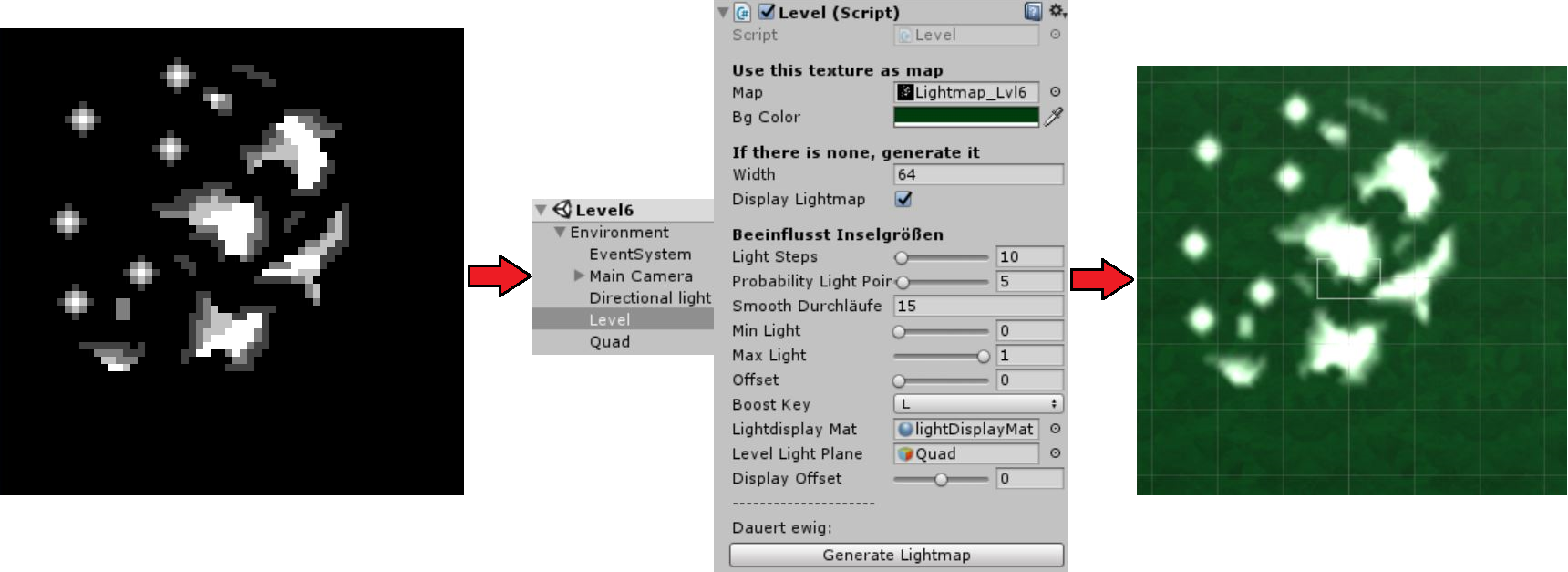


Abbildung The single steps necessary to create the lightmap for your level

The next step is to place walls in your scene to create single sections in your level. The wall prefab can be found under „prefabs“ and is basically a simple cube.



Abbildung Level background with walls

# 2. Place the player in the scene

The next step is to place and configure the player object in your scene. The important attributes for the gameplay are explained in the following. „Move Speed“ and „Acceleration“ influence the movement oft he player and are self-explaining. „Can Heal“ enables the Player to fire the healing beam (right mouse), which is used to regenerate the lifepoints of cells. „Can Spike“ enables the Player to use the spikes (left shift), which damage nearby cells. Finally „Start ATP“ sets the starting energy available tot he Player. The Player can grab cells using the space key.

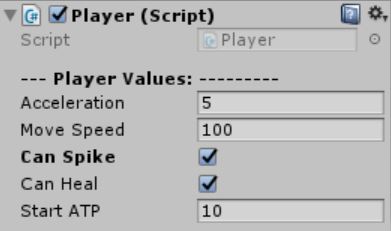


Abbildung Configurable Player attributes influencing the gameplay

# 3. Place the cells in the scene

First of all the Cell Type has to be defined. Neutral Cells are blue, Bad Cells are red and Good Cells are green. This matters in defining the win and loose conditions of a level. Then follows the section where the life energy attributes are defined. If „Use Photosynthesis“ is enabled the cells heals itself in light by the amount defined in „Heal Rate“. Contrary is „Consume Energy“, which makes the cell shrink in dark areas by the amount defined in „Energy Consumption“. If the cell can absorp atp it can also boost its energy by gathering energy points dropped by other cells. Lifepoints describe the start value of the cell and maximum lifepoints the highest amount a cell can have. The higher this amount the bigger the cell can grow in size.

If a cells has reached this threshold and “Cell Splits” is enabled the cell duplicates itself, producing two cells. The life points of those new cells are defined by the ratio in “Split Ratio”. If the cell was a neutral cell and “Mutates at Split” is enabled, it can either become a good or a bad cell. Cells can damage each other with absorption or spikes. The second picture shows a cell with activated absorption. It steals energy from other cells and thereby heals itself. Another way are spiky cells, which just do damage, but cannot be absorbed by other cells. Last but not least, the cell can drop something if it dies. Usually it drops Energy Points which the Player can gather to refill his own energy.



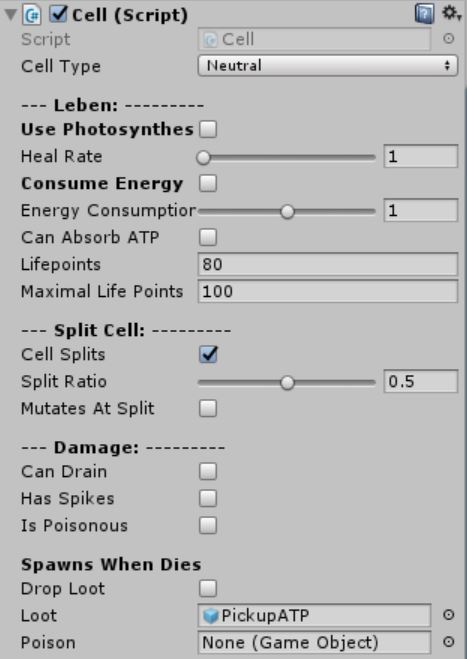


Abbildung configurable attributes of the cell and their visual counterparts

# 4. Place other objects

There exist other prefabs having influence on the gameplay.

## 4.1 DNA-Pickups

DNA-Pickups can change one or more attributes of the cell that gathered it. The Size describes how much attributes the pickup should change and the text can be used to explain the effect of the Pickup.



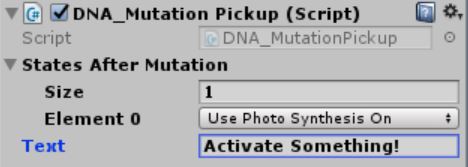


Abbildung configurable attributes of the DNA-Pickup

## 4.2 Robo-Cells

Robo-Cells attack cells specified in Kind (GOOD,BAD,NEUTRAL,ALL). The charge towards them with a speed defined in „Speed“ and deal the amount of damage defined in „Damage“



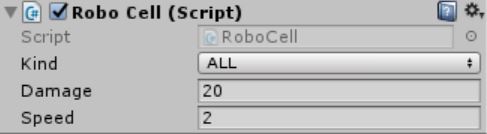


Abbildung configurable attributes of the robo-cells

## 4.3 Force Field, Laser Beams and Auras

Force fields transport everything in the direction of their flow with a force specified in „Force“. Beams can heal or damage cells that touch it. Their Length can be configured in „Distance and the power of the effect in „Ray Power“.



Abbildung configurable attributes of laser beams and force fields

The last prefab mentioned are the both auras. The fast aura doubles the movement speed oft he player and the slow aura halves it. However, they have no configurable attributes and are thus not further explained.

# 5. Setting Conditions

To complete or lose a level some conditions need to be defined. Therefore exist two condition triggers. One checks whether some cell counter is fulfilled, the other one whether the player has enough energy to complete the level. The example condition calls the specified function on a game object if exactly 10 cells exist which are good and have spikes activated. In this case GoalManager.Win is called that starts the next level and is a component of environment. The current goal can be shown to the User in the Canvas HUD.

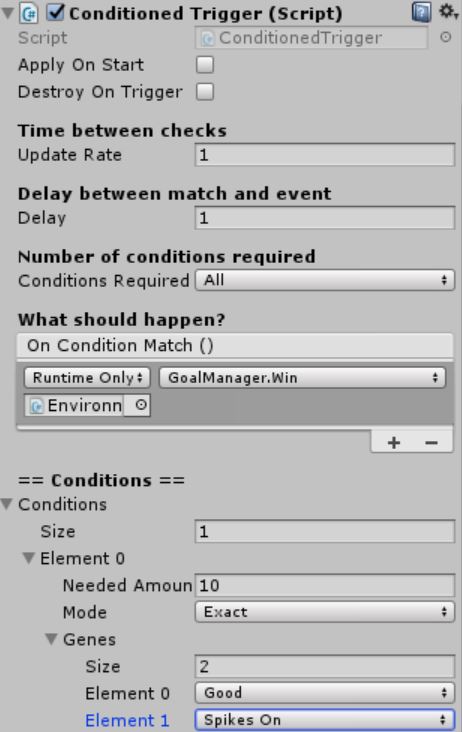


Abbildung example codition

More complicated setups can be created if one condition sets another one active or inactive. For examples refer e.g. to level 6.