Project 2 - PCG

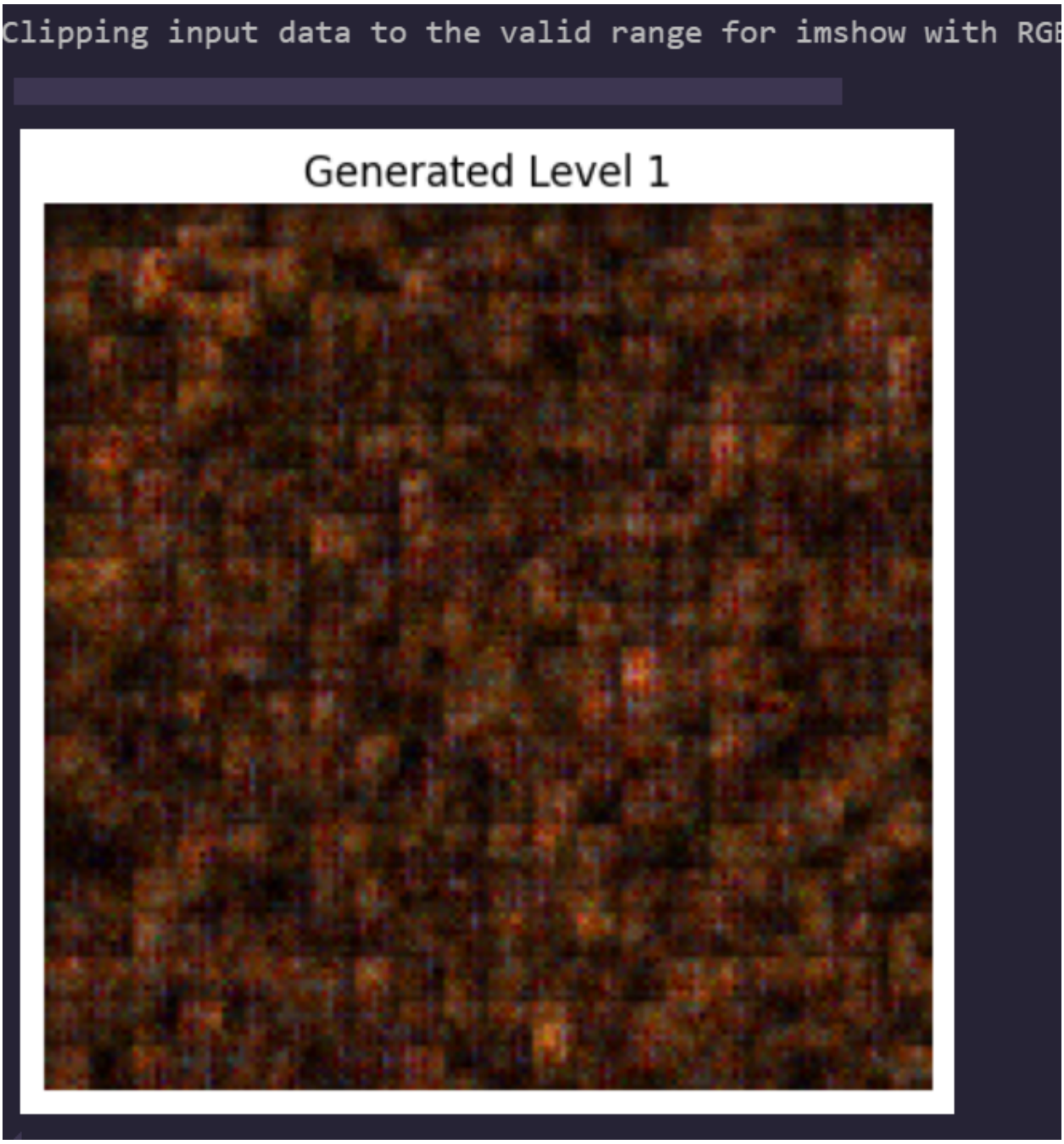
Roman Zvoda

Idea of the project was to generate a Sokoban levels based on the dataset of 1000 real levels using PCG. Implement autoencoder and sample the levels to create a new ones. I have had 2 different approaches with different libraries :

Approach 1 - Jax utilization

Initial attempt was to train CONV network of 3 layers to reconstruct the original levels. I split the dataset into 900 train levels and 100 test levels. I have made the model to train on 900 in bulk with no splitting into batches for 100 epoch which seemed to be not the most optimal solution as I ended up with 4 hour training process with a high CPU usage where 1 epoch took around 3 minutes to finish. Loss function seem to be working as there was a gradual decline ending up on a loss of 0.006 at the end with a pretty decent level of reconstruction quality. Player, boxes and goals had the correct outlines but the colour was not there yet.

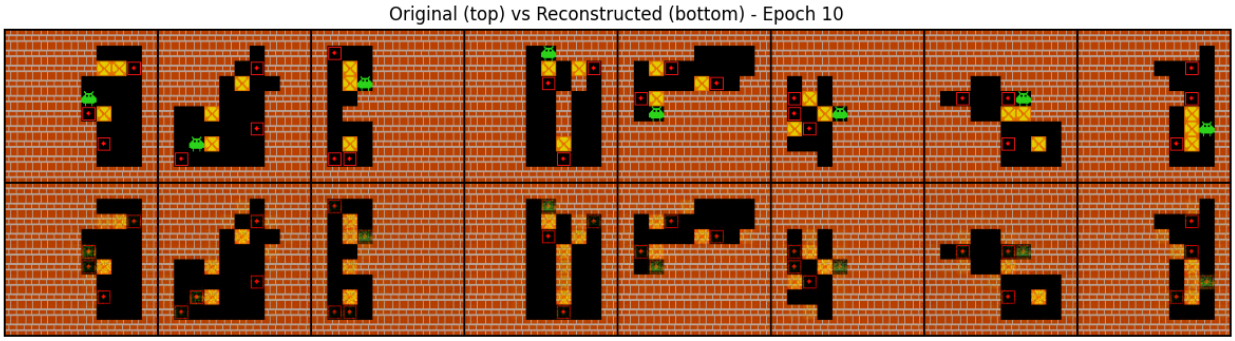
When generating new levels based on this model the output was not desirable and levels and colours did not match at all.



After spending a lot of hours on this approach I have decided to try something new and go with a tensor flow.

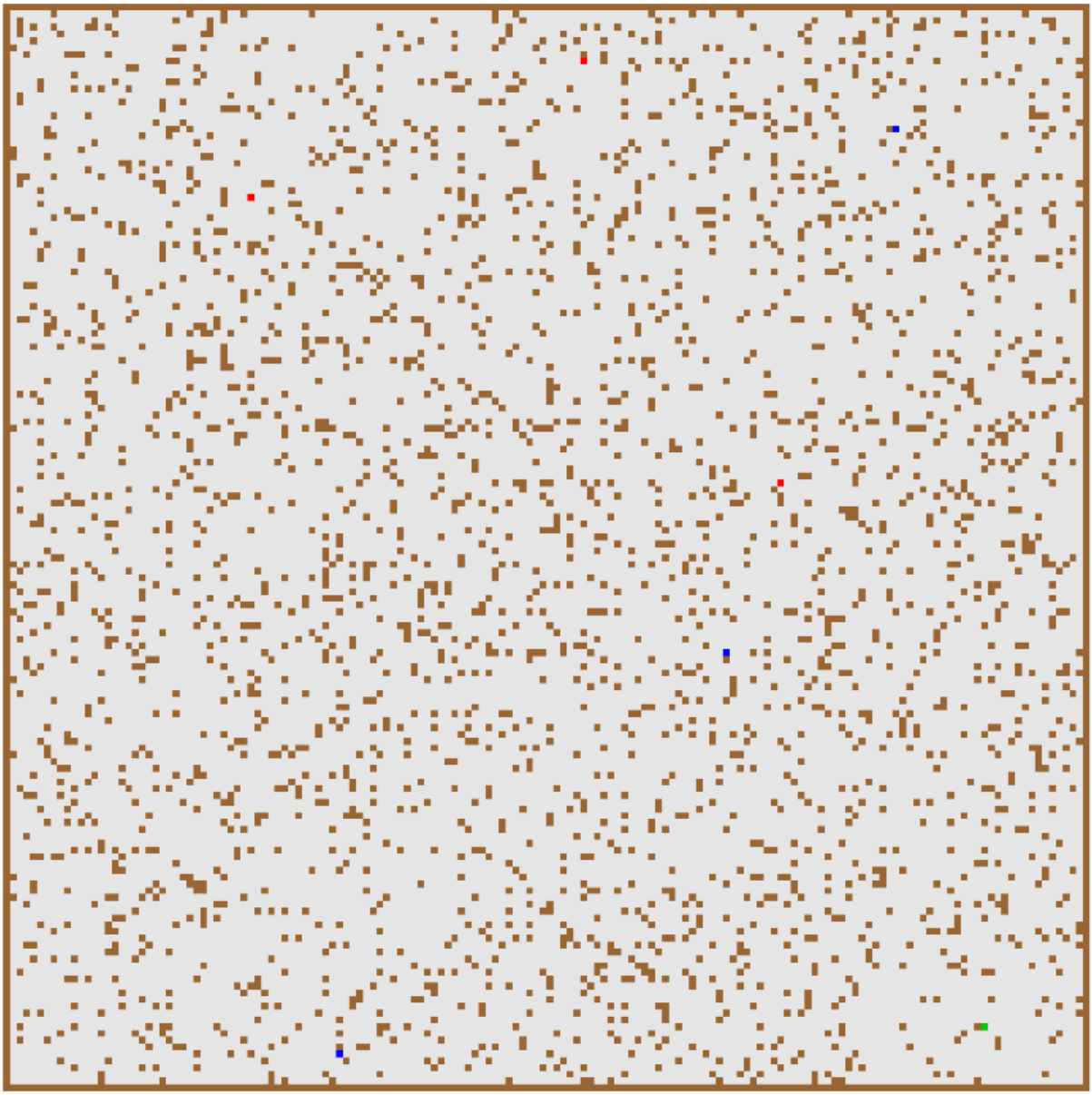
Approach 2 - TensorFlow utilization

TensorFlow and the model which I have managed to create was 100 times faster where training the model on 1000 level dataset was done in around 20 minutes with a loss function somewhere around 0.0002 after just 50 epochs. The reconstruction results were also much better:

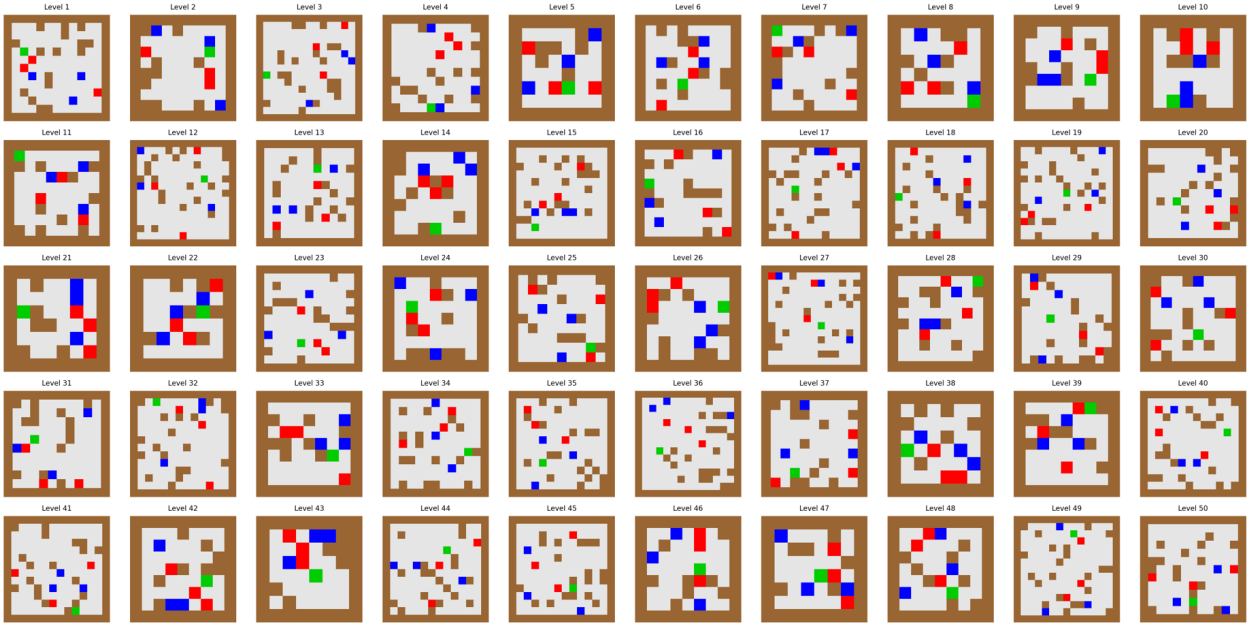


Walls, boxes and goals had a nice clean outlines only player model was lacking in clearance a bit.
Getting to the generating of new levels initial try was somewhat chaos due to no contrains being in use. In this example we can see one pixel representing one block from the game where the scale is much bigger than expected for a Sokoban level but at least all the elements of the game were present:

Level 3



Once constrains were put in place and the map size was scaled down I have managed to create 50 levels which made sense and were following all the principles of the game:



(Brown: wall, Green: Player, Red:Box, Blue: Goal)
Closer inspection of the levels shows most of the seem to be playable but the difficulty level is rather easy with a lot of free space to move and not so many walls places in the middle of the level. Also levels are represented as plain colours with no applied textures. This would be the next step in the process.

Results:

I have manged to create a model that is able to process dataset of levels and create a newly generated playable model using Autoencoder. The final result could be also much better in quality but due to lack of any example codes or ways to approach this I had to do everything from scratch which proved to be very time consuming in the end.