**Pattern Extraction**

This method extracts patterns from the levels that already exist in the framework to make predictions in the newly generated map. The algorithm works in the following way.
All the levels from which patterns are to be extracted are divided into three sections horizontally as shown in *Figure 2* and the patterns prediction characters of each section are stored separately.

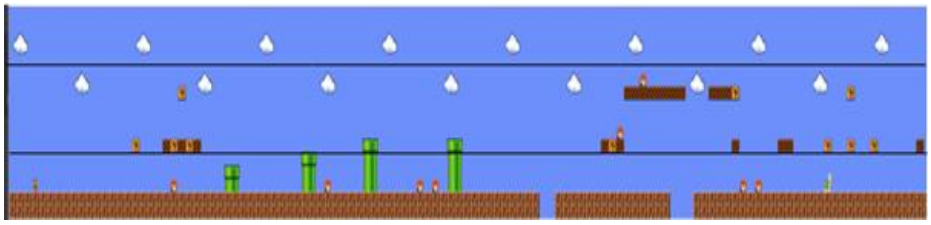

***Figure 2: Division of levels into horizontal sections (sky (0-4), intermediate (5-11), ground (12-15))***

The patterns are extracted row wise and each pattern consists of specified fixed length. The character that appears after this pattern is the prediction character which is stored in a list as shown in *Figure 3*. The patterns and respective prediction characters are stored in HashMap in java as key-value pairs where the pattern is the unique key and the value is an array of possible character values that could occur after this pattern. Whenever this pattern is encountered again the character array is updated.

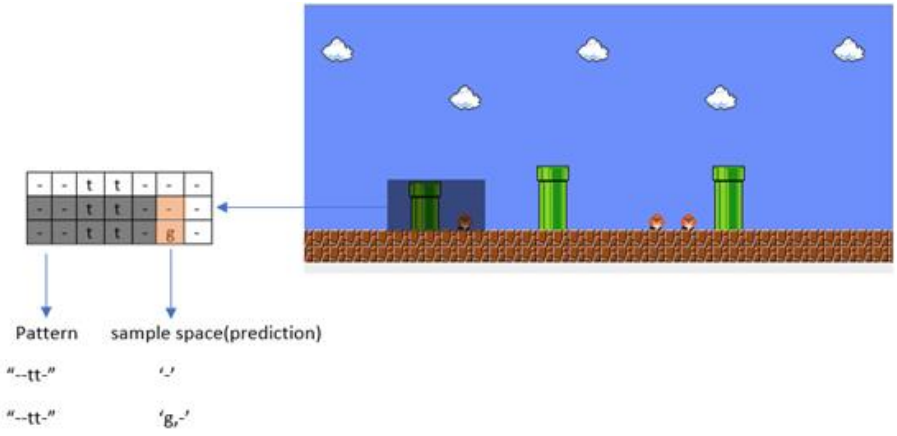


***Figure 3: Extraction of pattern and characters that represent tiles, from a sample level***

During level generation whenever a certain pattern is encountered in the newly generated level a random character is picked from the sample space to fill that tile position. The list does not keep count of each character and therefore the list can contain repetitive characters. For example, if a list contains 5 characters - ('X','X','X','-','g') and one character is picked at random then, there is a probability of 60% that the character picked is 'X' (ground) and the probability of '-' (empty) and 'g' (goomba) is 20% each. This is why frequency of each character in the sample space is important and therefore they are not discarded. A final level generated from extracted patterns is shown in *Figure 4.*

***Figure 4: final level generation from extracted patterns***