To begin with it is possible to think of each aspect of the game as its own individual systems, The deck of train cards to be drawn affects the whole game but can be considered modularly.

To start with I developed the card drawing system, it’s an atomic aspect to the game. Meaning, it’s completely necessary for every other aspect to work and at the same time it stands by itself. So the game will start out as a simple card drawing simulator. Players will take it in turns to draw train cards to their hands. Once this is complete the rest of the game will use this system to work.

Rather than represent the deck of cards as an overly abstract array where an index corresponds with a particular colour, instead I use a list of TrainCard objects, each TrainCard having a string value for its colour and an integer for the amount that are present. This makes the code a lot easier to read, and though it’s probably less efficient it’s unlikely that lack of processing power will be a problem in a piece of software of this scope.

The discard pile is also represented in this script, this way I can just add each discarded card immediately and when the deck runs out the discard pile is switched in for the active pile.

To begin with I know how many cards are in the deck in total, it begins at 45, there are 12 cards of each colour and 15 rainbow trains. Knowing this I can use a roulette selection as would be used in a GA to decide what train cards are drawn when drawing blind. This bypasses the need to shuffle the deck and represent each card in a particular order, which is unnecessary as the order the cards are in will never come into play and the blind draw is effectively random.