K.MANOJ KUMAR

DEPARTMENT OF COMPUTER SCIENCE,

SIDDHARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY,

PUTTUR,ANDHRA PRADESH.

[MANOJ2ROX@GMAIL.COM](mailto:MANOJ2ROX@GMAIL.COM)

K.BHAVYA

DEPARTMENT OF COMPUTER SCIENCE,

SIDDHARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY,

PUTTUR,ANDHRA PRADESH.

[KBHAVYA1976@GMAIL.COM](mailto:KBHAVYA1976@GMAIL.COM)

**ABSTRACT**

In this paper, we describe an engaging puzzle game called 2048 and outline a variety of exercises that can leverage the game’s popularity to engage student interest, reinforce core CS concepts, and excite student curiosity towards undergraduate research. Exercises range in difficulty from CS1-level exercises suitable for exercising and assessing 1D and 2D array skills to empirical undergraduate research in Monte Carlo Tree Search methods and skilled heuristic evaluation design.

**INTRODUCTION**

In this paper, we will describe this engaging game and point out ways that it may be used to advantage in the Computer Science curriculum.

The outline of the paper is as follows: We begin by describing the 2048 game, noting work already in progress to leverage the popularity of this game in CS education, and describing the scope of our unique contribution. Next, we discuss the modeling of the game state itself from simple fragments of the specification suitable for CS1 to a full game state model suitable for CS2. We then turn our attention to the creative challenge of heuristic game state evaluation.

**Game Description**

The 2048 game is played on a 4-by-4 square grid that is partially filled with tiles labeled with powers of 2. The primary goal of the game is to merge randomly generated tiles in order to create a tile labelled 2048 (211) as shown above. Beyond this primary goal, however, players may continue play so as to achieve higher scores and tiles with higher powers as well. The highest possible tile would be 131,072 (217), although this would be improbable to achieve.

Initially, the board has two randomly generated tiles. Randomly generated tiles are distributed uniformly over all empty cells, but such tile values are either 2 (with probability .9) or 4.

In each turn, a player may choose to move left, right, up, or down. A move in a given direction is legal if it results in a change to the grid. When no legal move exists, the game is over. Change occurs through tiles merging and/or sliding into different cells. Thus, the grid must contain at least one empty cell or at least one pair of identical, successive tiles along a row or column for the game to continue.

When the player chooses a direction, the tiles in all rows or columns aligned with that direction merge and slide in that direction if possible. Let us refer to a single row/column as a line, with the front of the line being the end farthest in the direction of motion. From the front to the back of the line, successive identical tile pairs merge into a new tile with their sum. (Merged tiles cannot merge again in the same movement.) Then, all tiles (merged or not) slide as far as possible in the direction of motion. Consider the following grids before (left) and after (right) a downward move:

In the rightmost column, both bottom and top pairs of vertically adjacent 2s merge into two 4s that slide downward. Note that these merged 4s do not then merge with each other in the same movement. In the next column to the left, note the 3 vertically adjacent 8s. The two bottommost 8s merge into a 16, being farthest in the direction of motion. In the next column to the left, there are no possible merges; the 2 simply slides down. In the leftmost column, two 2s merge into a 4.

The upper-left corner 2 is a randomly generated tile. After each legal move, a single tile is randomly generated as described above. Scoring is simple: Each time a merged tile is created, the value of that tile is added to the score. Thus, as the board is seeded with 90% 2s and 10% 4s after each move, one seeks to move so as to maximize one’s expected future score.