CSE13s Winter 2022
Assignment 4: The Game of Life

Theodore Ikehara, tikehara@ucsc.edu

CSE 13S - Winter 2022

Due: February 2nd at 11:59 pm

Design Document

Introduction:

1. About: In this assignment we are going to be creating our own abstract data type that is going to be assisting us in making the game of life. The game of life is a zero player game that is played on a grid and each cell in the grid is considered either dead or alive and evolves on its own to simulate the process of evolution. This code will evolve on its own and come out with its own outcome.
2. Rules:
   The game progresses through generations and there are only three rules that determine the state of each cell these rules are:
   I. Any live cell with two or three live neighbors survives
   II. Any dead cell with exactly three live neighbors becomes a live cell
   III. All other cells die either due to loneliness or overcrowding

Layout/Structure:

1. The ADT (Abstract Data Type): The abstract data type we are creating is going to be the universe that the game is played on; this will be a finite 2d grid of cells. This allows the adt to keep track of the rules and position on its own. This allows the grid and cells to keep the information about its own position and state, this is important because the game needs to be able to play on its own and update itself. So as long as we implement the rules properly the game will work as intended.
2. How it works together: In the abstract data type we will have functions to get the data of the cells such as the col and row, and also the state, whether the cell is dead or alive. This is important as we would need to retrieve the data of each individual cell inorder to progress the game, we also need to be able to change the state of the cell. The adt would also need to be able to populate or generate its own grid. This adt would also need to be able to free

its own memory preventing memory leaks. The universe should also extend to infinite show there should be a function that solves the wrap around problem. Meaning we can only create a limited 2d space and thus cannot create an infinite plain. However, this should be more of a torus shape as in the edges of the 2d grid should wrap around and interact with the cells on the other side of the grid creating a grid that all the cells have adjacent cells to interact with. Without this the edges would be static after the initial run. Finally, the universe should also be able to print itself out, this means the entire grid should be printed to show the final results. Interacting with this adt will be *life.c* life is going to be where the main function is run. In the main function is where all the command line options are going to be placed. Life.c is where all the functions in the universe.c are going to be utilized. Here we will start by making a function call and creating an instance of the universe and populating the grid. Then we would assign the values for the grid either dead or alive depending on the command line options. The game then will be iterated and changed depending on the rules that are implemented. This should then print out the resulting grids using ncurses. After all is finished the memory should be freed and the program ends.

Pseudocode: