Conway’s Game of Life

* Requirements

After several days, our group 7 had finished the project Conway’s Game of Life.

From the Project Plan, we finished the basic function and most requirements: 1. Using Golang programing language; 2. Obeying the rules of cells activities (life or death); 3. Making the initial state; 4. Setting the models of specific results; 5. Initial state decided by users; 6. Error information display. However, we could not finish the page with OpenGL, because both members of our group are using MacOS, the version of our System cannot run the OpenGL. We try to figure out this issue, but the MacOS which is above Mojave cannot support OpenGL. So that we abandon this demand. Also, we have no time to finish the file input function.

* Design

The whole code of our program is design in a single file which is called Conway.

And the structure of the file separates three main function and one type struct. The first function is initialGound() function, which is used to set the initial live cells, and the different types of pattern also defined in this function through a switch statements. The second function is called nextState() function that is using to calculate the next state of the patterns through current state. The third function is main() function, which is used to set the size of the ground and offer an choice to user to set the pattern which their want to show in the ground. The type struct is used to define a two-dimantion array and several int variables.

* Citation

Wikipedia: Conway’s Game of Life, source: <https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life>

* Result

Screenshots: Figure 1 Step1, Figure 2 Step2-R, Figure 3 Step2-G, Figure 3 Step2-G and Figure 4 Step2-M

A screenshot of a social media post

Description automatically generated

Figure 1 step1

A screenshot of a social media post

Description automatically generated

Figure 2 Step2-R

A screenshot of a social media post

Description automatically generated

Figure 3 Step2-S

A screenshot of a social media post

Description automatically generated

Figure 3 Step2-G

A screenshot of a social media post

Description automatically generated

Figure 3 Step2-M

This program is significant for us, from this program, we have a better understanding for Golang programing language. Here are some details: we learnt how to use pointer and address to let sub programs communicate with each other. We also learnt how to create a matrix with slide that only support array. “:=” and “=” are totally different that if we choose the wrong one it will not work, “:=” using for define variables while “=” only use to change the value. When creating random number, if we want it to be different every time, we need a new seed. Time is a good way to get a new seed. Go can return multiple variables together directly from a sub program and also push in multiple as well. If you change the value of a matrix with another matrix while using a pointer, the order of run while cause mistakes. We can transfer the matrix into a string by write into buffer one by one and store to print it in a friendly way. We can also use os library to run command to the terminal. Here we use this way to clean the terminal to refresh the print.