ECE2534 Lab4 Report

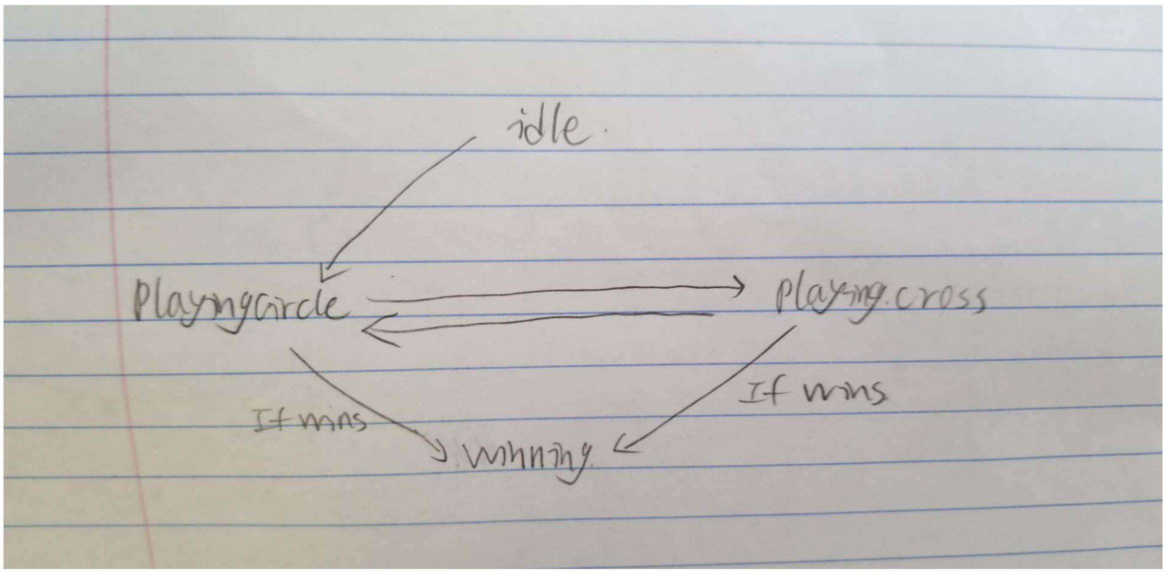
Section 1

In this lab, each student is required to design a Tic-Tac-Tone game, in which the game can be played by receiving a DTMF signal. On the start panel, the game is played against itself. In addition, the information about the game, include game name, score, and instruction, shows on the top. There are two mode in this game. The first mode is computer goes first , which is accomplished by pressing button one. The second mode is the user goes first, which is accomplished by pressing button two. After the computer makes a move, a sound will be played, and every time when the user or the computer wins the game, a short music will be played. However, the user could abort the game during the game, which is accomplished by pressing button one during the game. When the button is pressed, the rest empty cells will by filled by cross and a winner will be determined. After each round of the game, the game will return to the start panel and the score will be updated.

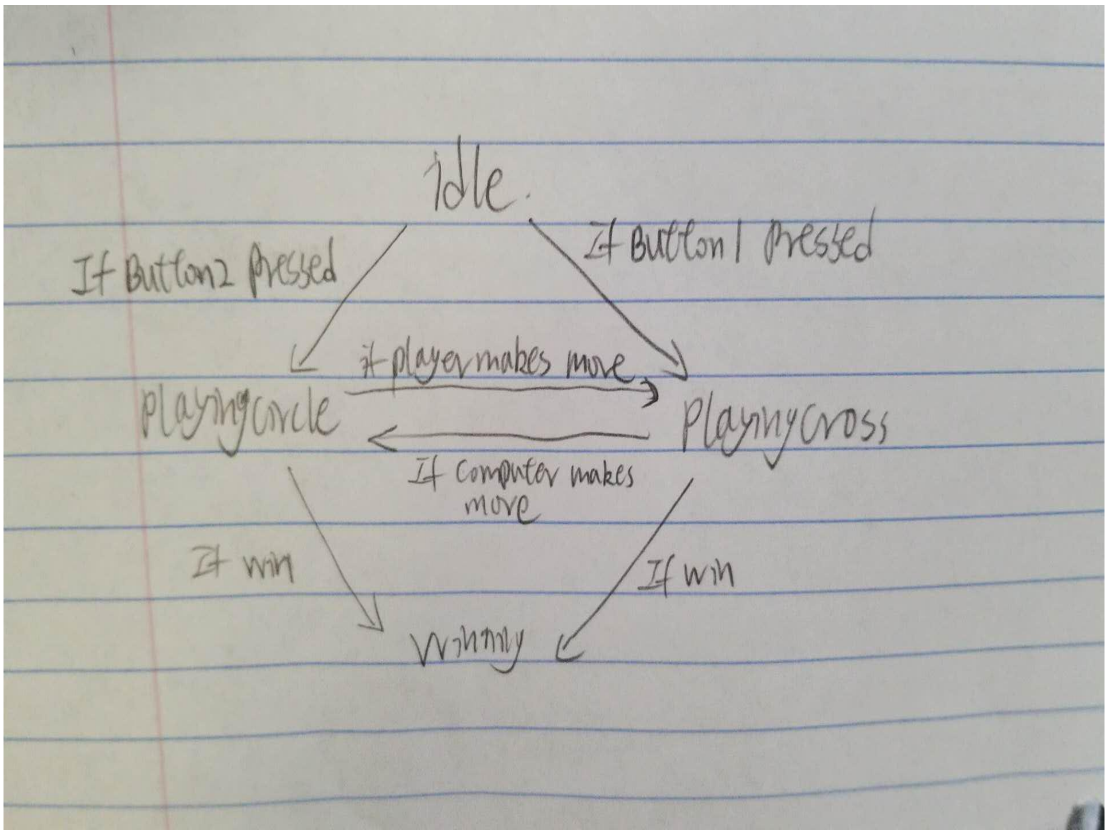
My application could accomplish all the functions above. But when my program listens the input, it is not very sensitive, I need to press the input a little bit longer, which is about two seconds. And in some cases, I need to press the input twice. In addition, when run the game, it requires absolute a quit environment.

Section 2

There are four state machines. The first one is the start game state machine. The start state is idle. Then it automatic goes to playing circle state . When the circle makes a move. It goes to playing cross state. When either of the circle or cross win the game. It goes to winning state, then it goes to idle state again.



The second state machine is when the program in the game mode.The start state is idle state. Then if butoon2 is pressed , the game goes to playingcircle, and if button1 pressed, it goes to playingcross state. After the circle makes a move, the state goes to playingcross state. If the cross makes a move, it goes to playingcircle state. If either the circle or cross wins, the state goes to winning state.



Section 3

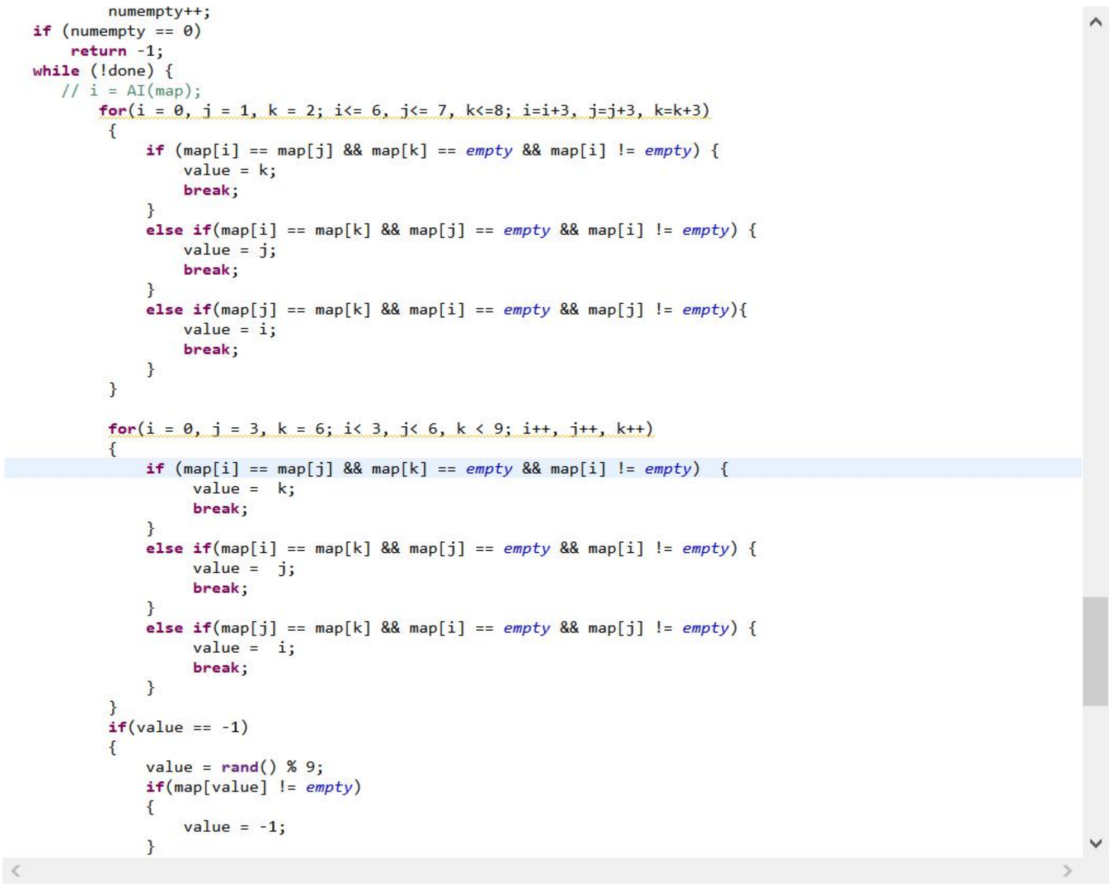
HAL functions

```
void PlaySound(tnote n, unsigned ms) //This function plays a note
void InitSound() //Initial the sound GPIO
int DTMFDisplay()  //check which input button is pressed, and return the value.
void DTMFAddSample(unsigned x) // sample the ADC result
void DTMFReset()   // reset the DTMF signal.
int PowerGoertzel(Gtap *t) //calculate the power of DTMF signal
void SampleGoertzel(Gtap *t, unsigned x) // calculate the sample result
void T32_INT1_IRQHandler() // The ISR function
void InitMicrophone() // initial the microphone
unsigned GetSampleMicrophone() // get microphone sample result
void InitADC()                //Initial ADC result
```

Section 4

My program is written in several files, which include goertzel.c, goertzel.h, dtmf.c, dtmf.h, sound.c, sound.h, maplogic.c, maplogic.h. Therefore, goertzel.c and dtmf.c are for implementing the listen input function. Sound.c is for playing musics and note. Maplogic.c is for checking if the player or computer wins and playing computer side.

However, I change the random play function. Therefore the computer will not play randomly. My implementation is shown below. Therefore, my algorithm will check all the columns and rows. If there are two same objects in the same row or column and the third cell is empty. Then the computer will plays in the third cell.



```
            numempty++;
    if (numempty == 0)
        return -1;
    while (!done) {
       // i = AI(map);
            for(i = 0, j = 1, k = 2; i<= 6, j<= 7, k<=8; i=i+3, j=j+3, k=k+3)
            {
                if (map[i] == map[j] && map[k] == empty && map[i] != empty) {
                    value = k;
                    break;
                }
                else if(map[i] == map[k] && map[j] == empty && map[i] != empty) {
                    value = j;
                    break;
                }
                else if(map[j] == map[k] && map[i] == empty && map[j] != empty){
                    value = i;
                    break;
                }
            }

            for(i = 0, j = 3, k = 6; i< 3, j< 6, k < 9; i++, j++, k++)
            {
                if (map[i] == map[j] && map[k] == empty && map[i] != empty)  {
                     value =  k;
                     break;
                }
                else if(map[i] == map[k] && map[j] == empty && map[i] != empty) {
                     value =  j;
                     break;
                }
                else if(map[j] == map[k] && map[i] == empty && map[j] != empty) {
                     value =  i;
                     break;
                }
            }
            if(value == -1)
            {
                value = rand() % 9;
                if(map[value] != empty)
                {
                    value = -1;
                }
```