SOFTWARE ENGINEERING PROJECT-II

IGEL Argern Game by Ronit Dahiya & Taranpreet Singh

APRIL 23, 2019

18204009- RONIT DAHIYA
18203372- TARANPREET SINGH

INTRODUCTION ABOUT THE GAME

Rules:

This is a multiplayer game with 2-6 players and a max of 24 hedgehogs(tokens). Each player is given 4 tokens of a particular colour, the player who gets 3 pieces of his own colour from the start to the ziel(goal) column wins.

The track on which they reach the goal doesn't matter.

In the beginning the all the players place their pieces in the starting column one plyer per turn, clockwise. Then pieces are stacked by players but you can't stack on your own piece. By the end the stacks are level and the game is ready to start.

Gameplay and coding:

The game is played clockwise and ends immediately when the requirements are met.

A move consists of three parts-

A) The player rolls a die. The rolled number indicates which track the player must make his move on at the end of his turn.

```
// 1. Roll the dice
int colFull;
//Randomising the roll outcome
int lucky = (rand() %  (5 - 0 + 1)) + 0; // Rand() creates a random number from 1-6 representing the number roled from the die.
printf("\nRolling Dice and your Lucky number is: %d\n", lucky );
```

This is the first basic step in our code, just basically a rand() acts as a die.

B) The player might or might not do a sidestep. He can move any of his pieces to do a sidestep.

```
// 2.  Sidestep one of players pieces (optional) i.e move any of players piece that is at top of the stack either below or above the row
int c = 0;
int sRow, sCol,dir=-1;

printf("\nDo you wanna sidestep one of your piece(1->YES 0-->NO)?");
scanf("%d",&c);

if(c == 1) {
  printf("\nEnter the row and column of the piece to be sidestepped and direction(0->Up and 1->Down): ");
  scanf("%d %d %d",&sRow,&sCol,&dir);
  token *st = board[sRow][sCol].stack_top;
  board[sRow][sCol].stack_top = board[sRow][sCol].stack_top->next;

  // Validation not done on dir
  //If input is 0 piece is moved up and 1 means down
  if(dir == 0)
  {
    st -> next = board[sRow-1][sCol].stack_top;
    board[sRow-1][sCol].stack_top = st;
  }

  else if(dir == 1) {
    st -> next = board[sRow+1][sCol].stack_top;
    board[sRow+1][sCol].stack_top = st;
  }
}

else {
  printf("\nNo Sidestep took!\n");
}
```

C) Take a step forward- The player must move a piece from the top of the stack on the track corresponding to the number on the die. If none of his pieces is free, he is forced to move the piece of any other player.

```
// 3. Move a piece from that numbered rolled from dice by 1 space (mandatory)
// To get the column of the token which needs to be moved.

int moveCol = -1;
for(k=0; k< NUM_COLUMNS; k++) {
  if(board[lucky][k].stack_top != NULL) {
    moveCol = k;
  }
}

// Token is being moved from one square to another (Obstacel case included too!).
  if(board[lucky][moveCol].type != OBSTACLE) {
      token *t = board[lucky][moveCol].stack_top;
      board[lucky][moveCol].stack_top = board[lucky][moveCol].stack_top -> next;
      t->next = NULL;
      moveCol++;

      board[lucky][moveCol].stack_top = t;

  }


  else {
      for(l=0; l<NUM_ROWS; l++) {
          if(board[l][moveCol].stack_top == NULL) {
            colFull = 0;
            break;
          }
      }

      if(colFull) {
          token *t = board[lucky][moveCol].stack_top;
          board[lucky][moveCol].stack_top = board[lucky][moveCol].stack_top -> next;
          t->next = NULL;
          moveCol++;
          board[lucky][moveCol].stack_top = t;
      }

      else {
          printf("\nSorry token is on Obstacle and can't move ahead!!!");
      }
  }
```

In our code we used two header files and made the use of stacks and functions.

The header files were **game_init** and **game_logic.**

Various functions(7 functions) were used for initialisation ,printing and game play.

OUR CONTRIBUTIONS:

A brief summary of what roles we did individually in our project.

The initialisation was done by Taranpreet and placing of tokens was done by Ronit.

Both of us contributed to the main file and the game logic.

Please do note that on GitHub all the files were pushed by Ronit and the work was done on his laptop as Taranpreet's laptop crashed. This issue has already been informed to the Professor.