|  |  |  |
| --- | --- | --- |
|  |  | SNAKE GAME  USING LINKED LIST DATA STRUCTURES AND ALGORITHMJ-COMPONENT |
| Profile  * ***THARUN*** * ***19MID0031*** * [***tharun.2019@vitstudent.ac.in***](mailto:tharun.2019@vitstudent.ac.in) * ***8667606006*** |  | AIM To understand the application of Array through the famous snake game ABSTRACT Games are becoming a major part of technology which yields high money. 10-15 years back, when the cell phones were being used, no one would be without the knowledge of the famous game in it. The Snake Game!!! Either your child would be playing that, or you would be playing that when you are child. This project depicts a major application of a Data structure called linked list through this Game INTRODUCTION Snake is a game where the player maneuvers a line which grows in length, with the line itself being a primary obstacle. There is no standard version of the game. The concept originated in the 1976 arcade game Blockade, and its simplicity has led to many implementations (some of which have the word snake or worm in the title). ALGORITHM/PROPOSED METHOD The above game is implemented using the Data structure called linked Arrays.      These various AASCI characters are used to design the borders and body parts of snake and design the obstacle or frog. Two different AASCI characters are used to design the snake to differentiate the head and body. SOURCE CODE: #include<stdio.h>  #include<windows.h>  #include<stdlib.h>  #include<conio.h>  #include<time.h>  #define N 20  #define M 40  int i,j,Field[N][M],x,y,Gy,Head,Tail,Game,Frogs,a,b,var,dir,Score,HighScore,Speed=99;  FILE \*f;  void print(); /// TO CREATING GAME ENVIRONMENT  void snakeinitialization();/// TO PLACE THE SNAKE AT CENTER OF THE ENVIRONMENT  void ResetScreenposition();/// TO PLACE GAME ENVIRONMENT CONSTANTLY AT THE  void Random(); /// TO PLACE FROG IN RANDOM POSITION  int getch\_noblock(); /// TO DETECT IF A KEY IS PRESSED FROM KEYBOARD  void movement(); /// TO MODEL THE MOVEMENT OF THE SNAKE  void TailRemove(); /// TO ADJUST THE TAIL POSITION WHILE MOVEMENT  void main()  {  snakeinitialization();  while(Game==0)  {  print();  ResetScreenposition();  Random();  movement();  TailRemove();  Sleep(Speed);  }  }  void print()  {  ///CREATING A SQUARE FOR GAME ENVIRONMENT  ///1ST LINE  for(i=0;i<=M+1;i++)  {  if(i==0)  printf("%c",201);  else if(i==M+1)  printf("%c",187);  else  printf("%c",205);  }  printf("Current Score: %d High Score: %d",Score,HighScore);  printf("\n");  ///MIDDLE AREAS  for(i=0;i<N;i++)  {  printf("%c",186);  for(j=0;j<M;j++)  {  if(Field[i][j] == 0)  printf(" ");  if(Field[i][j]>0 && Field[i][j] != Head)  printf("%c",176);  if(Field[i][j] == Head)  printf("%c",178);  if(Field[i][j] == -1)  printf("%c",15);  if(j == M-1)  printf("%c\n",186);  }  }  ///LAST LINE  for(i=0;i<=M+1;i++)  {  if(i==0)  printf("%c",200);  else if(i==M+1)  printf("%c",188);  else  printf("%c",205);  }  }  void snakeinitialization()  {  f=fopen("HighScore.txt","r"); /// TO ACCESS HIGH SCORE FILE  fscanf(f,"%d",&HighScore);  fclose(f);  ///TO PLACE THE SNAKE AT CENTER OF THE ENVIRONMENT  for(i=0;i<N;i++)  {  for(j=0;j<M;j++)  {  Field[i][j]=0;  }  }  x=N/2; /// TO PLACE SNAKE AT MIDDLE OF THE ENVIRONMENT  y=M/2;  Gy=y; /// TEMPORARY ASSIGNMENT  Head=5; /// POINT STORING 5 => HEAD  Tail=1; /// POINT STORING 1 => TAILS  Game=0 ; /// OTHER PARTS ARE BODIES  Frogs=0; /// INITIAL NUMBER OF FROGS  dir='d'; /// FOR CONTINUOS MOVEMENT  Score=0; /// INITIAL SCORE  for(i=0;i<Head;i++)  {  Gy++;  Field[x][Gy-Head]=i+1;  }  }  void ResetScreenposition()  {  HANDLE hOut;  COORD Position;  hOut=GetStdHandle(STD\_OUTPUT\_HANDLE);  Position.X=0;  Position.Y=0;  SetConsoleCursorPosition(hOut,Position);  }  void Random()  {  srand(time(0)); ///TO PLACE FROGS AT DIFFERENT POSITIONS  a = 1+rand()%18;  b = 1+rand()%38;  if(Frogs == 0 && Field[a][b] == 0)  {  Field[a][b]=-1;  Frogs=1;  if(Speed>10 && Score != 0)  Speed-=5;  }  }  int getch\_noblock()  {  if(kbhit())  return \_getch();  else  return -1;  }  void movement()  {  var=getch\_noblock();  var=tolower(var);  if(((var =='a' || var =='s') || (var =='d' || var =='w')) && (abs(dir-var)>5))  dir=var;  if(dir == 'd')  {  y++;  if(Field[x][y] !=0 && Field[x][y] != -1) ///TO CHECK IF BODY OF SNAKE COLLIDES  GameOver();  if(y == M-1)  y=0;  if(Field[x][y] == -1)  {  Frogs=0; /// TO CREATE NEW FROG ON NEXT LOOP  Tail-=2; /// TO INCREASE THE SNAKE LENGTH  Score+=5; /// TO INCREASE THE SCORE  }  Head++;  Field[x][y]=Head;  }  if(dir == 'a')  {  y--;  if(Field[x][y] !=0 && Field[x][y] != -1) /// TO CHECK IF BODY OF SNAKE COLLIDES  GameOver();  if(y == 0)  y=M-1;  if(Field[x][y] == -1)  {  Frogs=0; /// TO CREATE NEW FROG ON NEXT LOOP  Tail-=2; /// TO INCREASE THE SNAKE LENGTH  Score+=5; /// TO INCREASE THE SCORE  }  Head++;  Field[x][y]=Head;  }  if(dir == 'w')  {  x--;  if(Field[x][y] !=0 && Field[x][y] != -1) ///TO CHECK IF BODY OF SNAKE COLLIDES  GameOver();  if(x == -1)  x=N-1;  if(Field[x][y] == -1)  {  Frogs=0; /// TO CREATE NEW FROG ON NEXT LOOP  Tail-=2; /// TO INCREASE THE SNAKE LENGTH  Score+=5; /// TO INCREASE THE SCORE  }  Head++;  Field[x][y]=Head;  }  if(dir == 's')  {  x++;  if(Field[x][y] !=0 && Field[x][y] != -1) ///TO CHECK IF BODY OF SNAKE COLLIDES  GameOver();  if(x == N-1)  x=0;  if(Field[x][y] == -1)  {  Frogs=0; /// TO CREATE NEW FROG ON NEXT LOOP  Tail-=2; /// TO INCREASE THE SNAKE LENGTH  Score+=5; /// TO INCREASE THE SCORE  }  Head++;  Field[x][y]=Head;  }  }  void TailRemove()  {  /// TO MAINTAIN THE SIZE OF THE SNAKE  for(i=0;i<N;i++)  {  for(j=0;j<M;j++)  {  if(Field[i][j] == Tail)  {  Field[i][j]=0;  }  }  }  Tail++;  }  void GameOver()  {  printf("\a");  Sleep(1500);  system("cls");  if(Score>HighScore)  {  printf(" !!! New HighScore %d !!!\n\n",Score);  system("pause");  f=fopen("HighScore.txt","w");  fprintf(f,"%d",Score);  fclose(f);  }  system("cls");  printf("\n\n !!! GAME OVER !!! SCORE: %d\n",Score);  printf(" Press ENTER to continue and Press ESC to exit...");  while(1)  {  var=getch\_noblock();  if(var == 13)  {  Game=0;  snakeinitialization();  break;  }  else if(var == 27)  {  Game = 1;  break;  }  }  system("cls");  } RESULT Thus, this project signifies the application of Array. It also uses the concept of files to store and update the high score. This project just shows the basic model of snake game and doesn’t depict the original game. The colors and other attributes are added and made attractive in other application platforms. The motive of this project to understand the linked list application is shown. |