# Introduction:

10-15 years ago when 3d game players were not as available as recent time, we used to play 2D game players mostly. With the enhancement of technology, 3D games became more popular and we lost the taste of those 2D games. So we took a project to make a 2d video games which would bring back the memories of our childhood. Using the basic idea of 2D games and being inspired by the legendary snake game from the black and white era, we're making a 2d video game player with TFT display and Arduino.

# Equipment:

Both hardware and software equipment are used in this project.

## Hardware:

* TFT Screen (ILI9325D\_16) – used for getting the better GUI
* Arduino Mega R3 2560 – used as a microcontroller
* Mini Bread Boards – used to solve the common / extra connections
* Male/Female Jumper Wires – used for connection between two points/ports
* Resistances – used with the switches
* Battery – use to supply voltage
* Push button – use to make the controller (up, right, down, left)

## Software:

* Arduino IDE – for burning code on Arduino Mega R3 2560.
* Proteus –to design the hardware component model.

# Features:

* portability & hand-compatibility
* visualization of the game-paly on a TFT display which helps to get better GUI
* 2 axis movement.
* On the runtime user get different types of food (normal, bonus) which effects the score.
* Built in easy, medium & hard difficulties which will increase the speed of the snake.
* On the runtime when any level is passed successfully then another level will be visible on the screen where only barrier/walls change their place.
* Green led use to indicate the food eating of the snake & red led use to indicate the dead of snake.
* Buzzer helps to give the bonus-food sounds (Highly) & to give the snake dead-sounds (slightly).
* Controlled by the push switch (up, down, left, right, pause / ok).
* Game can be paused and restarted
* Game over state occurs when the snake gets into collision with its tail or with the walls
* Top 5 scores are saved

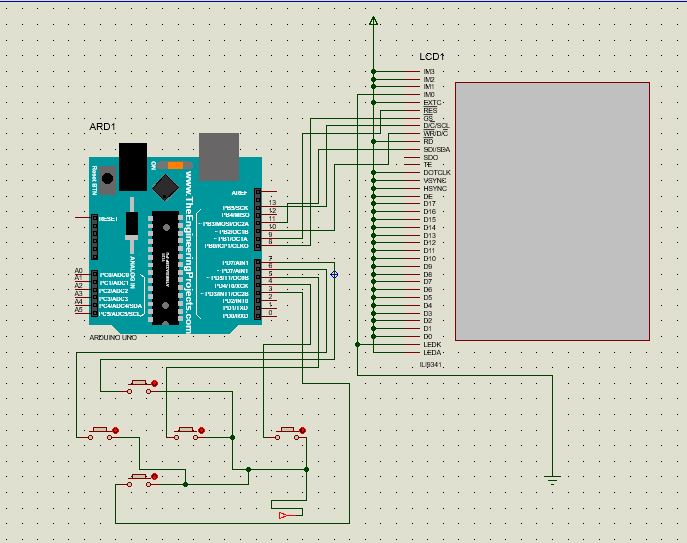
# Working Principle:

Basically the display works with pixels. Every pixel has its own co-ordinates starting from the upper left corner of the screen. All the things displayed in the game are drawn using this pixels. There are some certain functions which can be used to make any specific geometric shape like circle, rectangle, triangle etc.

In the starting, 5 circles are drawn subsequently to make the snake. In every frame, center of all the circles are moved in same direction by 10 pixels which makes the snake a moving one.

The direction of the snake is hardcoded for the starting. The direction can be changed anytime using the buttons. Buttons basically send Arduino a HIGH signal which determines the axis that should be increased or decreased.

# Circuit Diagram:



**Figure:** a snapshot from the design of the project in Proteus.

# Figures of the Project:

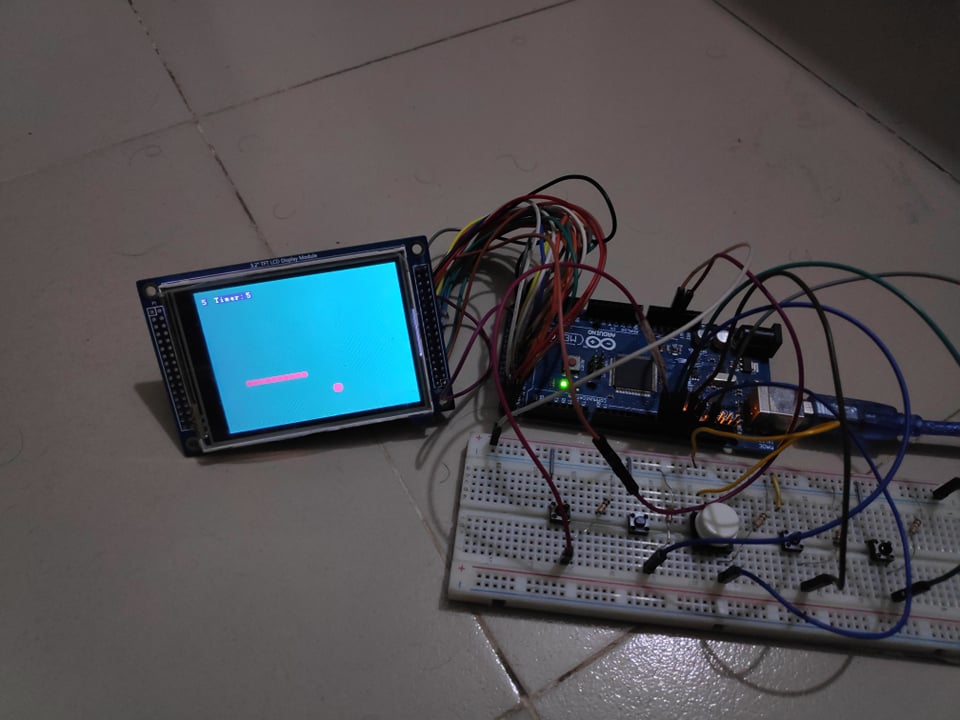
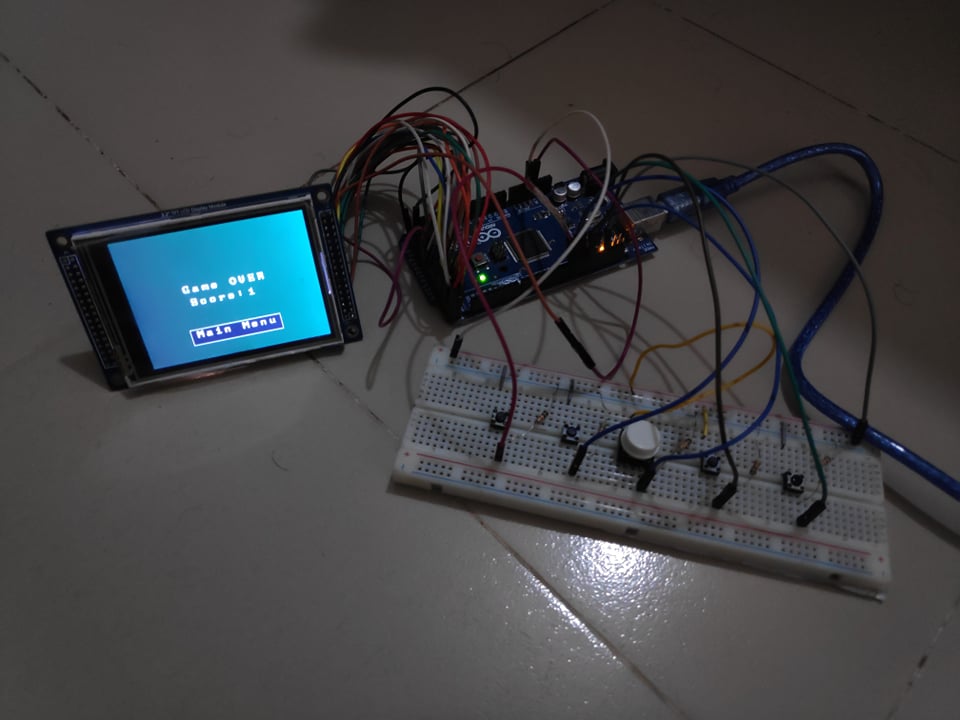
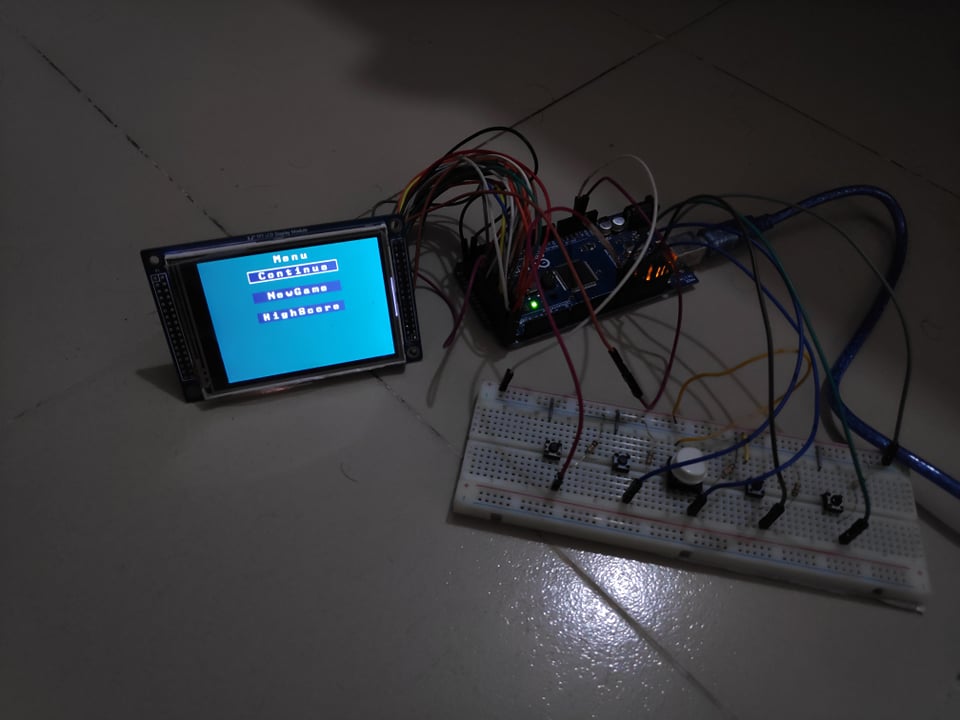


Figure: Game-play Figure: Game over screen



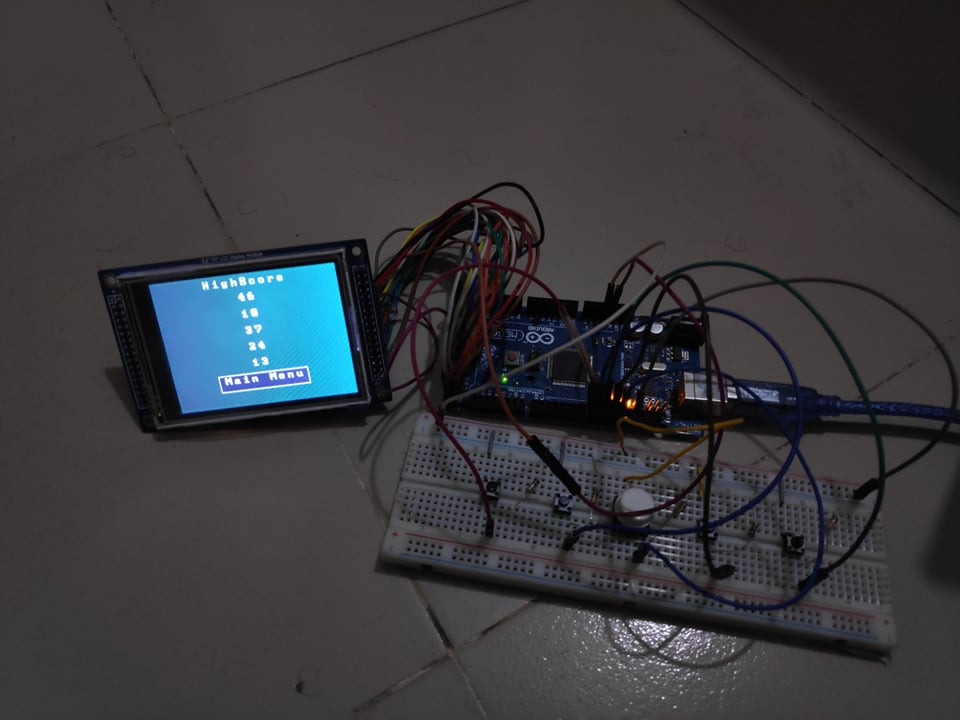


Figure: pause screen Figure: High Score screen

# Constrains:

* Main constrain of the project was to make the TFT Screen work and find the suitable library for the display.
* Working with push buttons made the project harder to control the direction-signals because all the input pin in the Arduino were getting false pulse.
* We were not familiar with the soldering process, so most difficult part was the Soldering part.

# Dos and Don’ts:

A comparison between the expected things and finished things of the project is given here.

* The project expected to have buzzers and LEDs to indicate different functions of the project. The intensity of the project was to be controlled. But LED and buzzers were not connected as it took all the times to solve the problems of basic game play.
* The basic feature of the project include changes of levels after a fixed amount of score. Different levels supposed to have different orientation of walls. Now the game player have only one level. It would only take about an hour to design a new layout.
* Difficulties (easy, medium, high) were not included to the game player because the snake now moves with the highest speed possible. So more speed can’t be added. Moreover, slowing down the snake looks very odd as frame rate goes very low.

# Conclusion:

Basically it is a demo of a video game player which we used in our childhood. This project helped to figure out the problem of using TFT Screens. Also we got idea on use of Arduino too.

# Appendix:

The code part of the project is given here:

|  |  |
| --- | --- |
| #include <EEPROM.h>  // UTFT\_Demo\_320x240 //  #include <UTFT.h>  extern uint8\_t SmallFont[];  extern uint8\_t BigFont[];  UTFT myGLCD(ILI9325D\_16,38,39,40,41);  struct item{  int x;  int y;  };  item snake[100];  item food;  int snake\_position = 1;//if 0 means that going left,1 means that going right,2 means that going down,3 means that going up  int snakesize = 5;  int point = 0;  int highscore[5];  int timer = 50;  int foodtype = 0; // if 1 then bonus otherwise normal  int bonusfoodstate = 1;  int currentsnakestate;  int test\_speed = 10;  myGLCD.InitLCD();  myGLCD.setFont(SmallFont);    //myGLCD.fillScr(0,106,78);  //backgound\_with\_board();  background();  draw\_snake();  generate\_food();  for(int i = 0;i<5;i++)  {  highscore[i] = EEPROM.read(i);  }  }    void loop() {  //backgound\_with\_board();    if(state == 0)  {  game\_running();  }  else if(state == 1)  {  gameover\_to\_menu();  gameover();  delay(100);  }  else if(state == 2)  {  myGLCD.setBackColor(0,106,78);  myGLCD.setColor(255, 255, 255);  myGLCD.printNumI(EEPROM.read(0),CENTER,32);    myGLCD.setBackColor(0,106,78);  myGLCD.setColor(255, 255, 255);  myGLCD.printNumI(EEPROM.read(1),CENTER,62);    myGLCD.setBackColor(0,106,78);  myGLCD.setColor(255, 255, 255);  myGLCD.printNumI(EEPROM.read(2),CENTER,92);    myGLCD.setBackColor(0,106,78);  myGLCD.setColor(255, 255, 255);  myGLCD.printNumI(EEPROM.read(3),CENTER,122);  myGLCD.setBackColor(0,106,78);  myGLCD.setColor(255, 255, 255);  myGLCD.printNumI(EEPROM.read(4),CENTER,152);  myGLCD.setColor(45,45,78);  myGLCD.fillRect(80 ,180 ,239,210);  myGLCD.setBackColor(45,45,78);  void checkmenuposition()  {  if(digitalRead(11) == 1 && menuposition == 0)  menuposition++ ;  if(digitalRead(9) == 1 && menuposition == 1)  menuposition --;  }  void selection\_menu()  {  if(digitalRead(12) == 1 && menuposition == 0)  {  state = 0;  }  if(digitalRead(12) == 1 && menuposition == 1)  {  state = 3;  highscoremenuposition = 1;  }  }  void menu(int y1,int y2)  {  background();  myGLCD.setFont(BigFont);  myGLCD.setBackColor(0,106,78);  myGLCD.setColor(244, 244, 244);  myGLCD.print("Menu",CENTER,5);  myGLCD.setColor(45,45,78);  myGLCD.fillRect(80 , 30 , 239 ,50);  myGLCD.setBackColor(45,45,78);  myGLCD.setColor(244, 244, 244);  {  Serial.println("1 menu");  ypos1 = 70;  ypos2 = 90;  menu(ypos1,ypos2);  // menuupblock = 0;  }  }  void check\_from\_pause\_menuposition()  {  if(digitalRead(11) == 1 && pausemenuposition != 2)  pausemenuposition++ ;  if(digitalRead(9) == 1 && pausemenuposition != 0)  pausemenuposition --;  }  void check\_from\_pause\_to\_selection\_menu()  {  if(digitalRead(12) == 1 && pausemenuposition == 0)  {  state = 0;  }  if(digitalRead(12) == 1 && pausemenuposition == 1)  {  snakesize = 5;  snake\_position = 1;  myGLCD.setColor(244, 244, 244);  myGLCD.print("Continue",CENTER,32);    myGLCD.setColor(45,45,78);  myGLCD.fillRect(80 ,70 ,239,90);  myGLCD.setBackColor(45,45,78);  myGLCD.setColor(244, 244, 244);  myGLCD.print("NewGame",CENTER,72);    myGLCD.setColor(45,45,78);  myGLCD.fillRect(80 ,110 ,239,130);  myGLCD.setBackColor(45,45,78);  myGLCD.setColor(244, 244, 244);  myGLCD.print("HighScore",CENTER,112);    myGLCD.setBackColor(45,45,78);  myGLCD.setColor(255, 255, 255);  myGLCD.drawRect(80, y1, 239, y2);//y1 = 30,y2 =50 starting block  }  void pause\_control\_menu()  {  if(pausemenuposition == 0)  {  Serial.println("0 menu");  ypos1 = 30;  ypos2 = 50;  pause\_menu(ypos1,ypos2);  // menuupblock = 0;  }  taking\_input();  running\_the\_snake();  if(snake\_position == 0)  snake[0].x = snake[0].x - test\_speed;    else if(snake\_position == 1)  snake[0].x = snake[0].x + test\_speed;    else if(snake\_position == 2)  snake[0].y = snake[0].y + test\_speed;    else if(snake\_position == 3)  snake[0].y = snake[0].y - test\_speed;  //delay(10);  for(int i = 0;i<snakesize;i++)  {    myGLCD.setColor(244, 42, 65);  myGLCD.fillCircle(snake[i].x,snake[i].y,radius);  }  backgound\_with\_board();  drawfood();  delay(100); }  void draw\_snake(){  int radius = 5;  int snakeheadx = random(10,220); | int state = 2; //0 means game running,1 means gameover,2 means menu,3 means highscore,4 means pause menu  int menuupblock = 1;  int menudownblock = 0;  int menuposition = 0;//if 0 means that Newgame,1 means that Highscore  int pausemenuposition = 0; //if 0 means that Continuegame,1 means that NewGame,2 means that HighScore  int highscoremenuposition = 0;  int gameovermenuposition = 0;  int gamepauseposition = 0;  int comefrompausemenu = 0;  int ypos1 = 30;  int ypos2 = 50;  int normal\_food = 0;  int bonus\_food = 0;  void setup()  {  Serial.begin(9600);  pinMode(8,INPUT);//left  pinMode(9,INPUT);//up  pinMode(10,INPUT);//right  pinMode(11,INPUT);//down  pinMode(12,INPUT);//ok  randomSeed(analogRead(0));  selection\_menu();  checkmenuposition();  control\_menu();  delay(100);  }  else if(state == 3)  {  highscore\_to\_menu();  Highscore();  delay(100);  }  else if(state == 4)  {  check\_from\_pause\_to\_selection\_menu();  check\_from\_pause\_menuposition();  pause\_control\_menu();  // pause\_menu(30,50);  delay(100);  }    }  void Highscore()  {  background();  myGLCD.setFont(BigFont);  myGLCD.setBackColor(0,106,78);  myGLCD.setColor(244, 244, 244);  myGLCD.print("HighScore",CENTER,5);  myGLCD.setColor(255, 255, 255);  myGLCD.print("Main Menu",CENTER,182);  myGLCD.setBackColor(45,45,78);  myGLCD.setColor(255, 255, 255);  myGLCD.drawRect(80, 180, 239, 210);    }  void highscore\_to\_menu()  {  if(digitalRead(12) == 1 && highscoremenuposition == 1)  {  if(comefrompausemenu == 1)  {  state = 4;  comefrompausemenu = 0;  }  else  state = 2;  }  }  myGLCD.print("NewGame",CENTER,32);  myGLCD.setColor(45,45,78);  myGLCD.fillRect(80 ,70 ,239,90);  myGLCD.setBackColor(45,45,78);  myGLCD.setColor(244, 244, 244);  myGLCD.print("HighScore",CENTER,72);    myGLCD.setBackColor(45,45,78);  myGLCD.setColor(255, 255, 255);  myGLCD.drawRect(80, y1, 239, y2);//y1 = 30,y2 =50  }  void control\_menu()  {  if(menuposition == 0)  {  Serial.println("0 menu");  ypos1 = 30;  ypos2 = 50;  menu(ypos1,ypos2);  // menuupblock = 0;  }  if(menuposition == 1)  snakesize = 5;  point = 0;  timer = 50;  foodtype = 0;  bonusfoodstate = 1;  normal\_food = 0;  bonus\_food = 0;  draw\_snake();  state = 0;  }  if(digitalRead(12) == 1 && pausemenuposition == 2)  {  state = 3;  highscoremenuposition = 1;  comefrompausemenu = 1;  }  }  void pause\_menu(int y1,int y2)  {  background();  myGLCD.setFont(BigFont);  myGLCD.setBackColor(0,106,78);  myGLCD.setColor(244, 244, 244);  myGLCD.print("Menu",CENTER,5);  myGLCD.setColor(45,45,78);  myGLCD.fillRect(80 , 30 , 239 ,50);  myGLCD.setBackColor(45,45,78);  if(pausemenuposition == 1)  {  Serial.println("1 menu");  ypos1 = 70;  ypos2 = 90;  pause\_menu(ypos1,ypos2);  // menuupblock = 0;  }  if(pausemenuposition == 2)  {  Serial.println("1 menu");  ypos1 = 110;  ypos2 = 130;  pause\_menu(ypos1,ypos2);  // menuupblock = 0;  }  }  void game\_running()  {  myGLCD.setFont(SmallFont);  int radius = 5;  if(digitalRead(12) == 1 && gamepauseposition == 0)  {  state = 4;  }  score\_board();  collition();  int snakeheady = random(40,210);  snake[0].x = snakeheadx;  snake[0].y = snakeheady;  for(int i = 1;i<snakesize;i++)  {  snake[i].x = snake[i-1].x+radius + radius;  snake[i].y = snake[i-1].y;  }    for(int i = 0;i<snakesize;i++)  {  myGLCD.setColor(244, 42, 65);  myGLCD.fillCircle(snake[i].x,snake[i].y,radius);  }    }    void running\_the\_snake()  {  for(int i = snakesize-1;i>0;i--)  snake[i] = snake[i-1];  } |