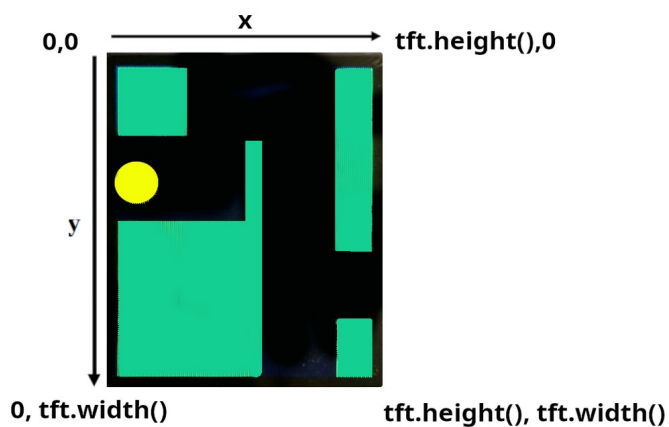
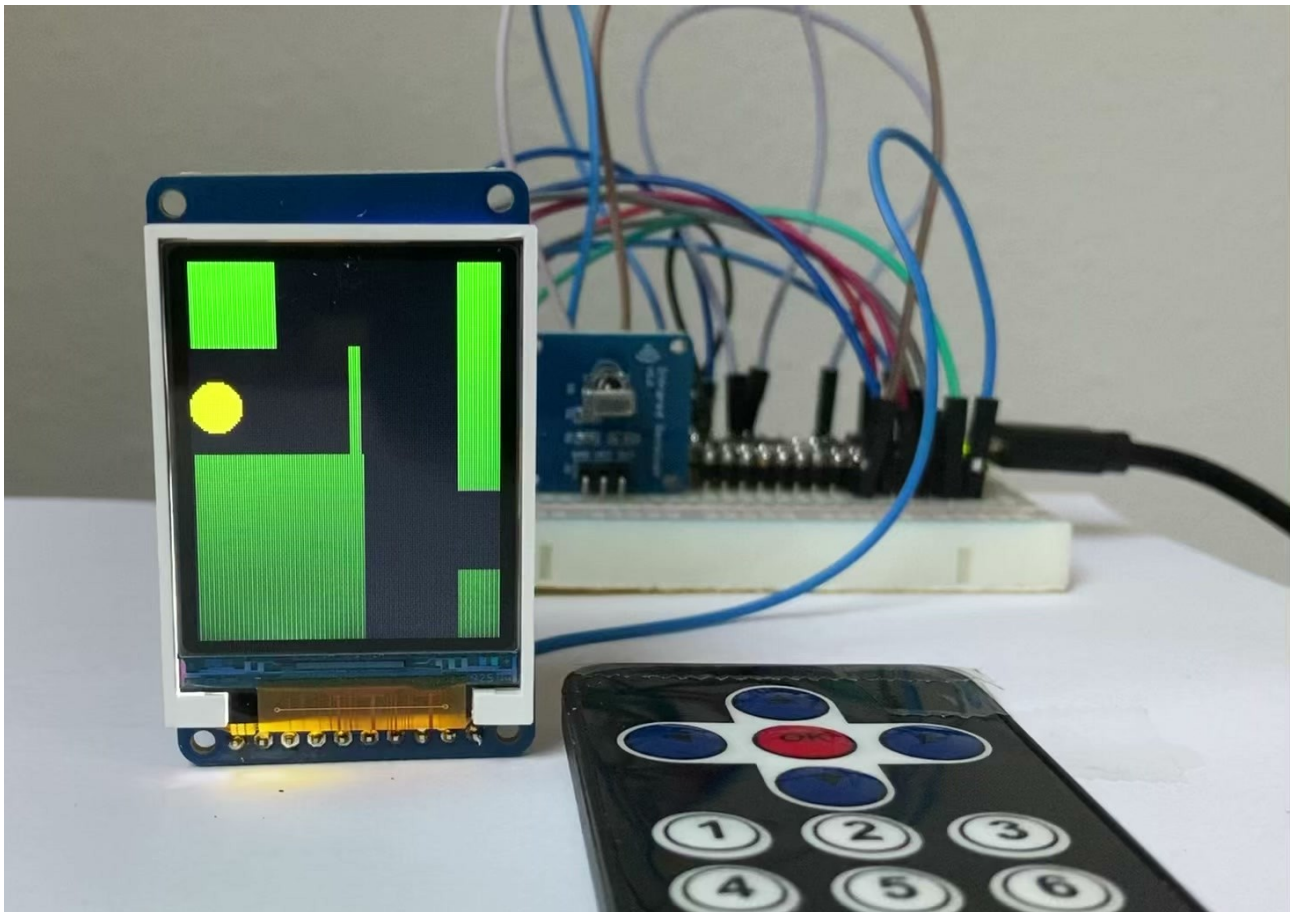


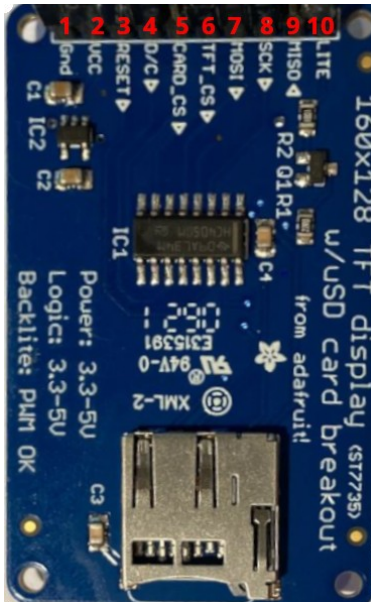
Mit Hilfe einer Fernbedienung wird auf einem TFT-Display ein Ball durch ein kleines Labyrinth bewegt. Am Ziel wird die Zeit gemessen.



1,77 und 1,8-Zoll große TFT-Module haben eine Bildschirmauflösung von 128x160 Pixeln.

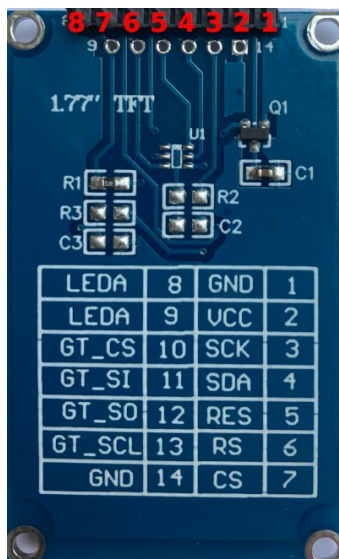
Benötigte Bauteile:

- ➔ Keyes Fernbedienung
- ➔ Adafruit 1,8 Zoll TFT/1,77 Zoll TFT
- ➔ Leitungsdrähte



Pinbelegung Adafruit 1,8 Zoll TFT

- 1 -> Gnd -> GND
- 2 -> VCC -> 5V
- 3 -> RESET -> 9
- 4 -> D/C -> 8
- 5 -> CARD_CS (nicht angeschlossen)
- 6 -> TFT_CS -> D10
- 7 -> MOSI -> 11
- 8 -> SCK -> 13
- 9 -> MISO (nicht angeschlossen)
- 10 -> 5V

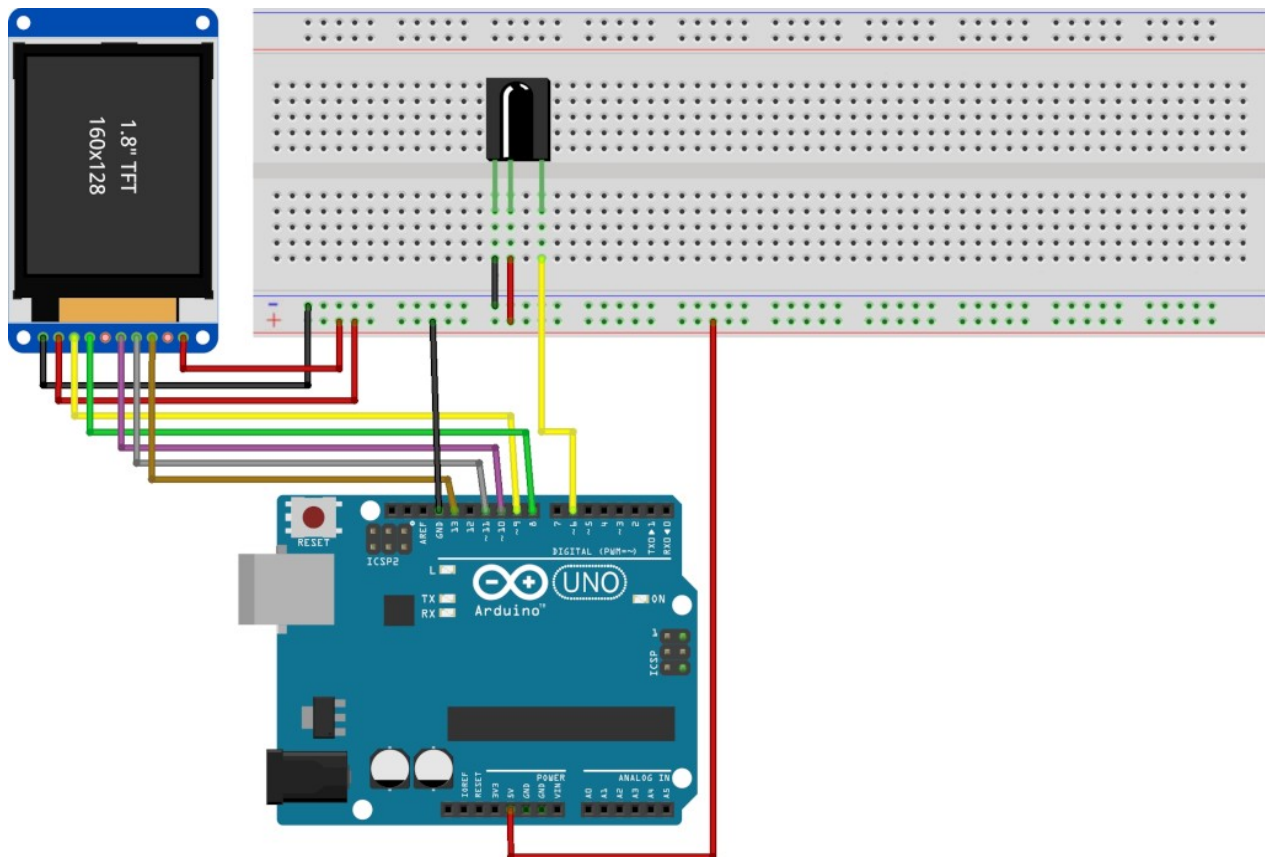


Pinbelegung 1,77 Zoll TFT

- 1 -> GND -> GND
- 2 -> VCC -> 5V
- 3 -> SCK -> 13
- 4 -> SDA -> 11
- 5 -> RES -> 8
- 6 -> RS -> 9
- 7 -> CS -> 10
- 8 -> LEDA -> 3,3V

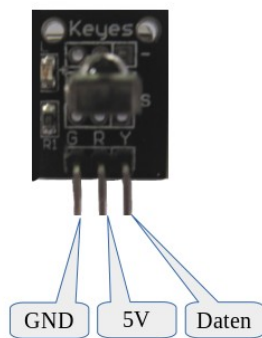
LEDA	8	GND	1
LEDA	9	VCC	2
GT_CS	10	SCK	3
GT_SI	11	SDA	4
GT_SO	12	RES	5
GT_SCL	13	RS	6
GND	14	CS	7

Baue die Schaltung auf.



fritzing

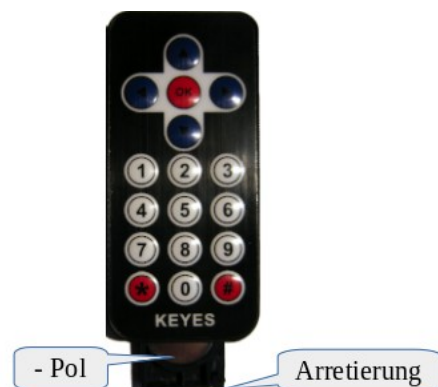
Achte auf die Pinbelegung der Infrarotempfänger.



Keyes-Empfänger



VS1838B



Achte darauf, dass die Batterie richtig eingelegt wurde. Der Minus-Pol zeigt nach oben.

Benötigte Bibliotheken:

BIBLIOTHEKSVERWALTER

Adafruit ST7735

Typ: Alle

Thema: Alle

Adafruit ST7735 and ST7789 Library von Adafruit
This is a library for the Adafruit ST7735 and ST7789 SPI displays. This is a library for the Adafruit ST7735 and ST7789 SPI displays.
[Mehr Information](#)

INSTALLIEREN

BIBLIOTHEKSABHÄNGIGKEITEN INSTALLIEREN

Die Bibliothek **Adafruit ST7735 and ST7789 Library:1.10.0** benötigt ein paar andere Abhängigkeiten, die derzeit nicht installiert sind:

- Adafruit BusIO
- Adafruit GFX Library
- Adafruit seesaw Library
- SD

Möchten Sie alle fehlenden Ressourcen installieren?

OHNE ABHÄNGIGKEITEN INSTALLIEREN

ALLE INSTALLIEREN

BIBLIOTHEKSVERWALTER

IRRemote

Typ: Alle

Thema: Alle

IRremote von shirriff, z3t0, ArminJo
Currently included protocols: Denon / Sharp, JVC, LG / LG2, NEC / Onkyo / Apple, Panasonic / Kaseikyo, RC5, RC6, Samsun...
[Mehr Information](#)

INSTALLIEREN

Überblick über die Methoden der Bibliotheken Adafruit ST7735/TFT

Schlüsselwort	Parameter	Aktion
begin();		TFT starten
initR(INITR_*TAB);	BLACKTAB GREENTAB REDTAB	Farbschema bestimmen
setRotation(Richtung);	Richtung = 0 → nicht drehen Richtung = 1 → 90° drehen Richtung = 2 → 180° drehen	Bildschirm ausrichten
fillScreen(Farbe)		Bildschirmhintergrund
drawLine(StartX, StartY, EndeX, EndeY, Farbe);		Linie zeichnen
drawFastHLine(StartX, StartY, Länge, Farbe);		horizontale Linie zeichnen
drawFastVLine(StartX, StartY, Länge, Farbe);		vertikale Linie zeichnen
drawRect(StartX, StartY, Breite, Höhe, Farbe);		Rechteck zeichnen
drawRoundRect(StartX, StartY, Breite, Höhe, Eckenradius, Farbe);		abgerundetes Rechteck zeichnen
fill.Rect(StartX, StartY, Breite, Höhe, Farbe);		ausgefülltes Rechteck zeichnen
drawCircle(MittelpunktX, MittelpunktY, Radius, Farbe);		Kreis zeichnen
fillCircle(MittelpunktX, MittelpunktY, Radius, Füllfarbe);		Ausgefüllten Kreis zeichnen
setCursor(x, y);		Cursor setzen
setTextSize(Textgröße);	Textgröße:	Textgröße bestimmen

Schlüsselwort	Parameter	Aktion
	1 - 4	
setTextColor(Farbe);		Textfarbe setzen
print("Text"); println("Text");		Text schreiben
setTextWrap(true/false);	false → Text fließt über den Rand des TFTs hinaus true → Text wird am Ende umgebrochen	Zeilenumbruch

Die Fernbedienung sendet beim Druck auf die Tasten einen Zahlencode.



Die Tastencodes beziehen sich auf die **Keyes-Fernbedienung**. Die Tastencodes anderer Fernbedienungen kannst du mit Hilfe des Seriellen Monitors herausfinden.

Die Tastencodes der Keyes Fernbedienung (hexadezimal und dezimal).

oben	links	rechts	unten	OK	1	2	3	4
0x46 70	0x44 68	0x43 67	0x15 21	0x40 64	0x16 22	0x19 25	0xD 13	0xC 12
5	6	7	8	9	*	0	#	
0x18 24	0x5E 94	0x8 8	0x1C 28	0x5A 90	0x42 66	0x52 82	0x4A 74	

Die Tastencodes kannst du mit folgendem Programm herausfinden. Sie werden im Seriellen Monitor angezeigt.

```
// benötigte Bibliothek einbinden
#include "IRremote.hpp"

// der Pin, an dem der Infrarot-Empfänger angeschlossen ist
int EmpfaengerPin = 2;

void setup()
{
  // Seriellen Monitor starten
  Serial.begin(9600);

  // Infrarot-Empfänger starten
  IrReceiver.begin(EmpfaengerPin);
}
```

```
void loop()
{
  // decode() -> Daten lesen
  if (IrReceiver.decode())
  {
    // kurzes delay, damit nur ein Tastendruck gelesen wird

    delay(200);

    // resume -> nächsten Wert lesen
    IrReceiver.resume();

    /*
      der Empfänger empfängt zwischendurch Signale,
      die nicht ausgewertet werden können
      es sollen deshalb nur die korrekt erkannten Tasten ausgewertet werden
      die Dezimalwerte der korrekten erkannten Tasten liegen zwischen > 0 und < 95
      es wird abgefragt, ob das empfangene Kommando decodedIRData.command
      zwischen 0 und (&&) 95 liegt
    */
    if (IrReceiver.decodedIRData.command > 0 && IrReceiver.decodedIRData.command < 95)
    {
      Serial.print("Dezimalwert: ");

      // IrReceiver.decodedIRData.command = Wert der gedrückten Taste
      Serial.print(IrReceiver.decodedIRData.command);
      Serial.print(" -> ");

      // Werte abfragen und anzeigen
      if (IrReceiver.decodedIRData.command == 22) Serial.println("Taste 1");
      if (IrReceiver.decodedIRData.command == 25) Serial.println("Taste 2");
      if (IrReceiver.decodedIRData.command == 13) Serial.println("Taste 3");
      if (IrReceiver.decodedIRData.command == 12) Serial.println("Taste 4");
      if (IrReceiver.decodedIRData.command == 24) Serial.println("Taste 5");
      if (IrReceiver.decodedIRData.command == 94) Serial.println("Taste 6");
      if (IrReceiver.decodedIRData.command == 8) Serial.println("Taste 7");
      if (IrReceiver.decodedIRData.command == 28) Serial.println("Taste 8");
      if (IrReceiver.decodedIRData.command == 90) Serial.println("Taste 9");
      if (IrReceiver.decodedIRData.command == 82) Serial.println("Taste 0");
      if (IrReceiver.decodedIRData.command == 66) Serial.println("Taste *");
      if (IrReceiver.decodedIRData.command == 74) Serial.println("Taste #");
      if (IrReceiver.decodedIRData.command == 68) Serial.println("Pfeil links");
      if (IrReceiver.decodedIRData.command == 67) Serial.println("Pfeil rechts");
      if (IrReceiver.decodedIRData.command == 70) Serial.println("Pfeil oben");
      if (IrReceiver.decodedIRData.command == 21) Serial.println("Pfeil unten");
      if (IrReceiver.decodedIRData.command == 64) Serial.println("OK");
    }
  }
}
```

```

Ausgabe  Serieller Monitor x
Nachricht(Enter um Nachricht für 'Arduino Uno' auf '/dev/ttyACM0' Sowohl NL als... 9600 baud

Dezimalwert: 70 -> Pfeil oben
Dezimalwert: 68 -> Pfeil links
Dezimalwert: 64 -> OK
Dezimalwert: 67 -> Pfeil rechts
Dezimalwert: 21 -> Pfeil unten
Dezimalwert: 22 -> Taste 1
Dezimalwert: 25 -> Taste 2
Dezimalwert: 13 -> Taste 3
Dezimalwert: 12 -> Taste 4
Dezimalwert: 24 -> Taste 5
Dezimalwert: 94 -> Taste 6
Dezimalwert: 8 -> Taste 7
Dezimalwert: 28 -> Taste 8
Dezimalwert: 90 -> Taste 9
Dezimalwert: 66 -> Taste *
Dezimalwert: 82 -> Taste 0
Dezimalwert: 74 -> Taste #

```

Binde die benötigten Bibliotheken ein und definiere die Variablen. Beachte die Kommentare.

```

#include "IRremote.hpp"
#include "Adafruit_ST7735.h"

/*
  Arduino
  SCK 13
  RST 9
  DC 8
  CS 10
  COPI 11
  #define TFT_CS 10
  #define TFT_RST 9
  #define TFT_DC 8
*/

#define TFT_CS 10
#define TFT_RST 9
#define TFT_DC 8

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);

/*
  Farben als hexadezimal definiert
  alternativ:
  int SCHWARZ = 0;
  int FARBE = 15;
  . . .
*/

```

```
// Pin des IR-Empfängers
int EmpfaengerPin = 6;

// Farben
#define SCHWARZ 0x0000    // dezimal 0
#define BLAU 0x000F       // dezimal 15
#define ROT 0xF800        // dezimal 40664
#define GRUEN 0x0E81      // dezimal 3713
#define CYAN 0x07FF       // dezimal 2047
#define MAGENTA 0xF81F    // dezimal 63519
#define GELB 0xAFE5       // dezimal 65504
#define WEISS 0xFFFF      // dezimal 65535
#define BRAUN 0xFC00      // dezimal 64512
#define GRAU 0xF7F0       // dezimal 63472
#define GRUENGELB 0xAFE5  // dezimal 45029
#define DUNKELCYAN 0x03EF // dezimal 1007
#define ORANGE 0xFD20     // dezimal 64800
#define PINK 0xFC18       // dezimal 64536

// Farbe der Blöcke
#define FarbeBloecke GRUEN

// Farbe des Kreises
#define Kreisfarbe ROT

// Farbe der Schrift
#define Schriftfarbe WEISS

// Spiel starten wenn * gedrückt wurde
bool SpielStart = false;

// Radius des Kreises
const int Radius = 10;
const int Abstand = Radius * 2;

// je höher, dest langsamer
const int Geschwindigkeit = 100;

// Vorwärtsbewegung des Kreises in Pixeln
const int Bewegung = 5;

// Startposition des Kreises
int CursorX = Radius;
int CursorY = tft.height() / 2 - Abstand;

// Variable für die Zeitmessung
long Start;
```

Der setup-Teil:

```
void setup()
{
  // Startbildschirm
  // schwarzes Farbschema horizontale Ausrichtung
  // Cursor setzen, Schriftgröße und -farbe definieren
  tft.initR(INITR_BLACKTAB);
  tft.setRotation(0);
  tft.fillScreen(SCHWARZ);
  tft.setTextSize(2);
  tft.setCursor(1, 10);
  tft.setTextColor(Schriftfarbe);

  tft.println("Start:");
  tft.print("-> *");
  Serial.begin(9600);
  IrReceiver.begin(EmpfaengerPin);
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  // nur ausführen, wenn SpielStart false
  if (IrReceiver.decode() && !SpielStart)
  {
    // nächsten Wert lesen
    IrReceiver.resume();

    // Start mit *
    if (IrReceiver.decodedIRData.command == 66)
    {
      // Spiel wird gestartet
      SpielStart = true;

      // Parcours bauen
      ParcoursBauen();

      // Zeitmessung starten
      Start = millis();
    }
  }

  // wenn die Taste * gedrückt wurde
  if (SpielStart)
  {
    if (IrReceiver.decode())
    {
      // nächsten Wert lesen
      IrReceiver.resume();
    }
  }
}
```

```

// nach oben
if (IrReceiver.decodedIRData.command == 0x46)
{
    // Kreis an der aktuellen Position "löschen"
    tft.fillCircle(CursorX, CursorY, Radius, SCHWARZ);

    // wenn der Bildschirmrand noch nicht erreicht wurde
    // rückwärts (Richtung x = 1) bewegen
    if (CursorY > Radius) CursorY -= Bewegung;
    tft.fillCircle(CursorX, CursorY, Radius, Kreisfarbe);
    delay(Geschwindigkeit);
}

// nach unten
if (IrReceiver.decodedIRData.command == 21)
{
    tft.fillCircle(CursorX, CursorY, Radius, SCHWARZ);
    if (CursorY < tft.height() - Radius) CursorY += Bewegung;
    tft.fillCircle(CursorX, CursorY, Radius, Kreisfarbe);
    delay(Geschwindigkeit);
}

// nach links
if (IrReceiver.decodedIRData.command == 68)
{
    tft.fillCircle(CursorX, CursorY, Radius, SCHWARZ);
    if (CursorX > Radius) CursorX -= Bewegung;
    tft.fillCircle(CursorX, CursorY, Radius, Kreisfarbe);
    delay(Geschwindigkeit);
}

// nach rechts
if (IrReceiver.decodedIRData.command == 67)
{
    tft.fillCircle(CursorX, CursorY, Radius, SCHWARZ);
    CursorX += Bewegung;
    tft.fillCircle(CursorX, CursorY, Radius, Kreisfarbe);
    delay(Geschwindigkeit);
}
}

// rechter Bildschirmrand erreicht -> Spielende
if (CursorX > tft.height() - Radius * 2)
{
    ErgebnisZeigen();
}
}
}

```

Jetzt fehlen noch die Methoden ErgebnisZeigen() und ParcoursBauen():

```
void ErgebnisZeigen()
{
    // Zeit berechnen
    int Sekunden;
    long VerstricheneZeit = millis() - Start;
    Sekunden = int(VerstricheneZeit / 1000);

    // Zeit anzeigen
    tft.fillScreen(SCHWARZ);
    tft.setTextSize(2);
    tft.setCursor(1, 10);
    tft.println("Zeit:");
    tft.println(String(Sekunden) + " s");

    tft.setCursor(1, 40);
    tft.setTextColor(ROT);
    tft.println();
    tft.println("Neustart:");
    tft.println("-> *");

    // Neustart
    SpielStart = false;
}

void ParcoursBauen()
{
    CursorX = Radius;
    CursorY = tft.height() / 2 - Abstand;
    tft.fillScreen(SCHWARZ);

    // Kreis anzeigen
    tft.fillCircle(CursorX, CursorY, Radius, Kreisfarbe);

    // Parcours "bauen"
    tft.fillRect(65, 35, 5, 45, FarbeBloেকে);
    tft.fillRect(1, 1, 35, 35, FarbeBloেকে);
    tft.fillRect(1, 80, 70, 80, FarbeBloেকে);
    tft.fillRect(110, 1, 70, 95, FarbeBloেকে);
    tft.fillRect(110, 130, 140, 160, FarbeBloেকে);
}
```

```
void ParcoursBauen()
{
  CursorX = Radius;
  CursorY = tft.height() / 2 - Abstand;
  tft.fillScreen(SCHWARZ);

  // Kreis anzeigen
  tft.fillCircle(CursorX, CursorY, Radius, KREISFARBE);

  // Parcours "bauen"
  tft.fillRect(65, 35, 5, 45, FARBE);
  tft.fillRect(1, 1, 35, 35, FARBE);
  tft.fillRect(1, 80, 70, 80, FARBE);
  tft.fillRect(110, 1, 70, 95, FARBE);
  tft.fillRect(110, 130, 140, 160, FARBE);
}
```