

RESUMO

O projeto Game in a Box, referente ao quarto período do curso de Engenharia de Computação da Pontifícia Universidade Católica do Paraná, propõe o desenvolvimento de um jogo semelhante aos fliperamas antigos, como os pinballs, onde o jogador controla um boneco com o objetivo de evitar os obstáculos contidos em uma esteira.

Palavras-chave: projeto, jogo, obstáculos, eletroímã.

SUMÁRIO

1 INTRODUÇÃO	3
2 OBJETIVOS	4
2.1 OBJETIVO GERAL	4
2.2 OBJETIVOS ESPECÍFICOS	4
3 MATERIAIS UTILIZADOS	5
4 DESCRIÇÃO GERAL	6
4.1 HISTÓRIA DO PROJETO	6
4.1.1 Eletrônica	7
4.1.2 Mecânica	14
4.1.3 Software	22
5 CÓDIGO FONTE	24
5.1 MAIN	24
5.2 LEITURA DE MOEDA	27
5.3 MÚSICA	28
5.4 LOOPING PRINCIPAL	33
5.5 LEITURA DE SENSORES	35
5.6 LEITURA DE COMANDOS	36
5.7 FIM DE JOGO	38
6 PROBLEMAS ENCONTRADOS	39
7 CONCLUSÃO	40
8 REFERÊNCIAS	41
9 GLOSSÁRIO	32

1 INTRODUÇÃO

Frente a um mundo onde a era digital prevalece sobre os antigos instrumentos analógicos, criou-se a concepção entre o mundo dos “gamers” de que jogos ditos bons eram os jogos que possuíam alta tecnologia e gráficos exorbitantes. Entretanto, os novos celulares smartphones trouxeram de volta a simplicidade e a elegância dos primeiros jogos digitais a serem desenvolvidos.

Diante disso, os alunos tiveram a ideia de trazer o projeto G.I.B (Game in a box), que consiste em uma esteira com diversos obstáculos, onde um jogador, no controle de um personagem, tem por objetivo desviar destes obstáculos, seja pulando ou desviando.

2. OBJETIVOS

Para uma boa compreensão do projeto é necessário conhecer suas principais características e todos os requisitos que o mesmo deverá atender.

2.1. OBJETIVO GERAL

Desenvolver um projeto para os programas de aprendizagem das disciplinas de Física IV, Resolução de Problemas em Engenharia II e Sistemas Digitais II que consista em jogo analógico a fim de trazer diversão aos seus usuários.

2.2. OBJETIVOS ESPECÍFICOS

Abaixo serão listados os objetivos que regeram o desenvolvimento do projeto:

- Desenvolver uma esteira contendo diversos obstáculos, na qual se passará o jogo;
- Criação do controle para o usuário;
- Criação do eletroímã;
- Criar algoritmo para controle do jogo;
- Desenvolver circuitos elétricos para controlar os motores e o eletroímã;
- Utilizar os conhecimentos em Sistemas Digitais II para construir um contador de pontos;

3. MATERIAIS UTILIZADOS

- 1 Arduino Mega;
- 5 Fins-de-Curso Mecânicos (Sensor de Toque);
- 1 Carrinho de Impressora com Motor DC;
- 1 Tapete felpudo;
- 1 Dobradiça;
- 1 Barra de metal;
- 8 Conectores PVC (pequenos canos de PVC);
- 1 Motor com caixa de redução
- Aproximadamente 100M de fio de Cobre esmaltado 22awg
- 1 transistor TIP32
- Resistores de 10k, 220R, 1k e 820k
- 1 Sensor infravermelho;
- 1 Sensor Eletromagnético;
- 1 Manete de joystick;
- 1 Botão de fliperama;
- Imãs de ferrite;
- Papel com espuma;
- Linha de Nylon;
- 1 Buzzer contínuo de 3V;
- Placas de Fenolite;
- Madeira;
- Pregos
- Grampos de Pressão;
- Parafusos;
- Tinta preta;

4. DESCRIÇÃO GERAL

4.1. HISTÓRIA DO PROJETO

Inicialmente a ideia do projeto era criar uma esteira com obstáculos em três dimensões e com tamanhos diferenciados onde o personagem do jogo obrigasse a saltar ou desviar do mesmo, fazendo com que prendesse mais a atenção do jogador. Porém, devido a problemas técnicos, a ideia de obstáculos em 3D foi abandonada, pois traria diversas complicações e a falta de tempo não nos permitiu lidar com estas.

Após revisar e estipular os conceitos do projeto o mesmo foi oficializado em forma de plano de trabalho e entregue ao professor da disciplina de Resolução de Problemas em Engenharia como proposta de trabalho para o semestre.

4.1.1. ELETRÔNICA

O projeto eletrônico iniciou-se com a definição dos módulos necessários para viabilizar todas as funcionalidades propostas. Foram eles:

- Sensores;
- Controles do usuário;
- Controle de motores;

O primeiro passo a ser seguido foi o de projetar o circuito da ponte-h, responsável pela alimentação do motor da esteira e do carrinho de impressora que movimenta o boneco.

Inicialmente foi feito um projeto usando o CI I298, porém não obtivemos êxito. O CI I298 trouxe bastante dificuldade, inclusive em experiências passadas, pois exige montar um circuito relativamente grande para controlá-lo se comparado a outros CI's de ponte h, além da facilidade com que este CI queima.

Após avaliarmos as soluções, decidimos por utilizar o CI I293D. O CI I293D é um circuito de ponte h dupla (Permite o controle de dois motores independentemente) de fácil utilização, por isso a escolha deste. Devido à sua facilidade de utilização, não foi necessário fazer um projeto no software Eagle

Cadsoft, bastando somente seguir um esquema encontrado na internet.

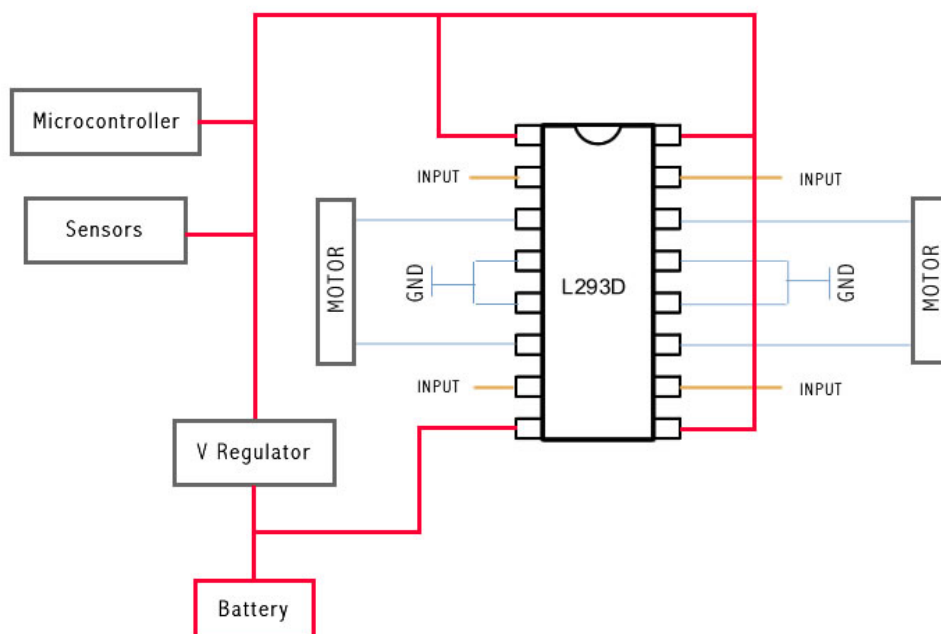


Figura 1 - Esquema utilizado para a construção da placa do L293D.

A segunda etapa do projeto eletrônico consistiu na elaboração do eletroímã que seria responsável por fazer o movimento de pulo do personagem. Esta etapa foi bastante difícil, pois ninguém fazia ideia do quão forte o eletroímã teria que ser para erguer o boneco, vencendo a gravidade, o peso do boneco e da estrutura que o ergueria e o atrito.

O primeiro modelo de eletroímã construído deu-se de maneira simples: Um emaranhado de fio de cobre esmaltado enrolado em um prego. Feito os testes, constatamos que este não era suficiente para o que precisávamos, porém este modelo foi utilizado para dar andamento a outros testes enquanto não desenvolvíamos outro eletroímã eficiente.

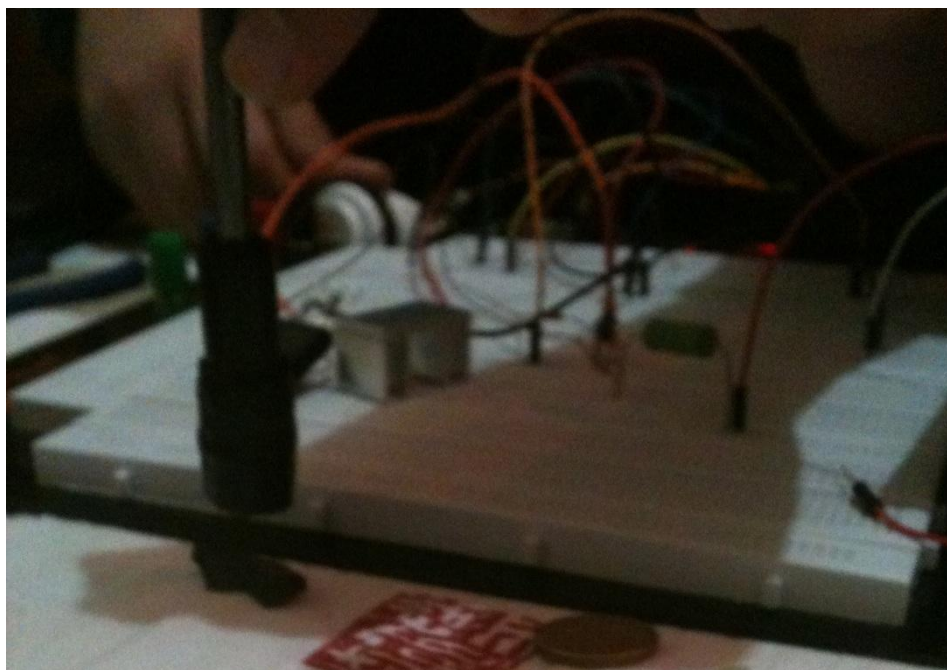


Figura 2 - Primeiro eletroímã sendo testado.

O grupo chegou a um consenso de que o fato do dispositivo estar muito fraco era devido a ter pouco fio de cobre, o que resultava em um campo magnético muito fraco. Sendo assim, partimos para a compra de mais fio de cobre para produzir um segundo modelo, modelo este que também não deu certo, mesmo contendo mais fio que o primeiro.

Após algum tempo, surgiu a ideia de criar não mais um eletroímã comum, mas sim um solenoide, que ao ser energizado puxaria um prego para dentro dele, prego este que estaria fixo à estrutura que levantaria o personagem principal, como mostra a figura 3.



Figura 3 – Solenóide.

Feito isto, partimos então para a construção de um sistema de contagem de pontos para o jogador. Por ser apenas um projeto protótipo, o contador de pontos conta somente até 99.

O CI contador utilizado foi o CD4040, que é na verdade um CI divisor por 4096. Foram encontrados outros CIS contadores mais simples da família TTL, mas como os outros CIS a serem utilizados eram da família CMOS, optou-se por utilizar o CD4040 para manter uma única tecnologia. O contador de pontos funciona da seguinte maneira: É utilizado na placa um NE555 responsável por gerar um clock de aproximadamente 0,93HZ, que seria o clock do 4040 responsável por representar as unidades dos pontos. Após passar pelo CD4040, o sinal passaria por um decodificador BCD para 7 segmentos e então para o display. Ao formar nas suas saídas o número binário 1010 (dez em decimal), uma porta AND do CI 4081 manda sinal para reiniciar este contador e gerar um pulso de clock para o contador de dezenas, até que este chegue a nove e zere ambos os contadores. Além disso, fez-se necessário o uso de clear e enable dos contadores para que o arduino possa mandar sinal para controlar esta contagem, como mostram as figuras 4 e 5.

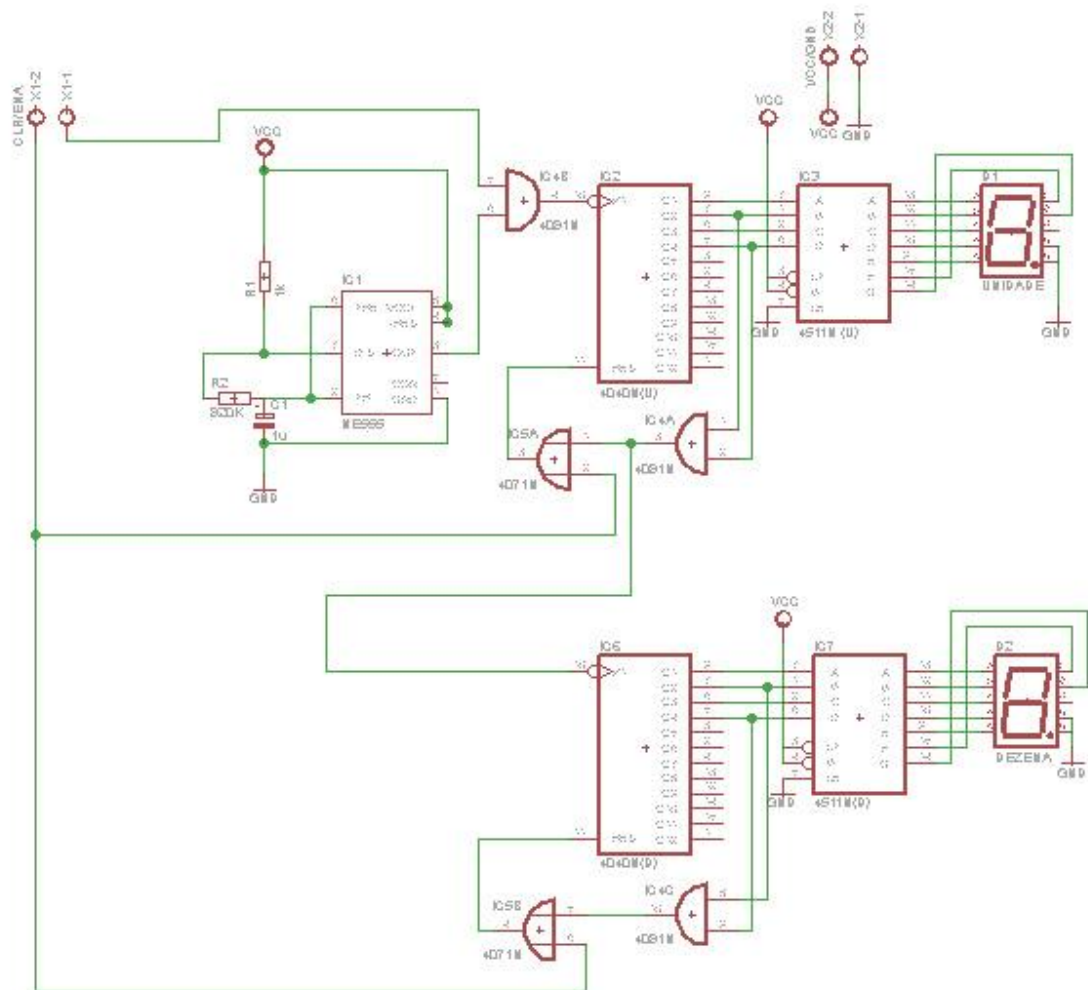


Figura 4 - Esquemático display de pontos.

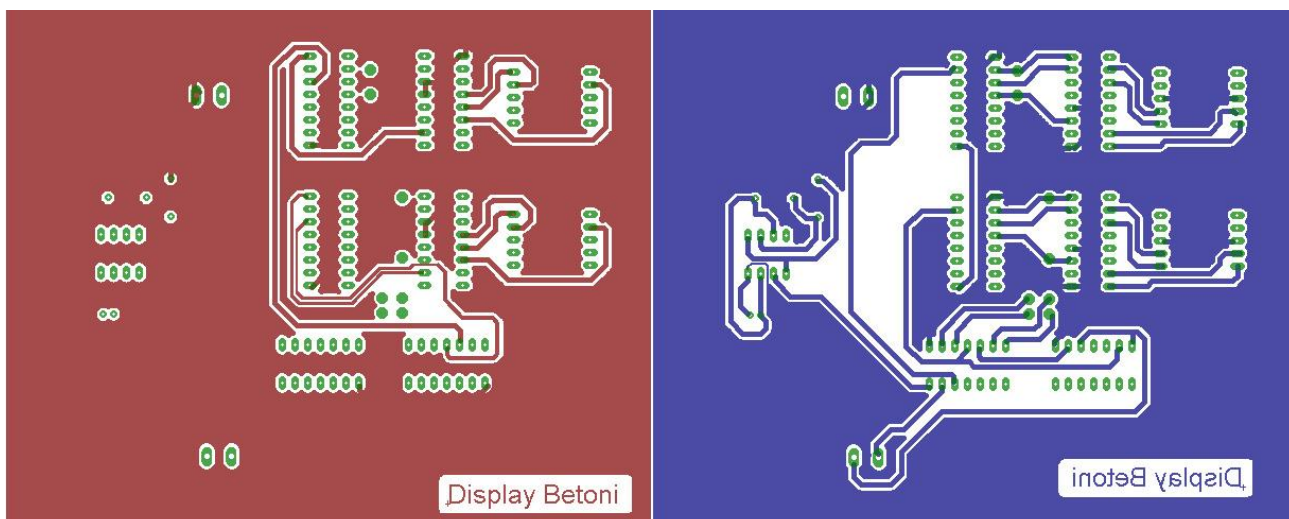


Figura 5 - Board do display de pontos. Camada superior à esquerda e inferior à direita.

As duas últimas placas a serem construídas, finalizando assim a parte eletrônica do projeto foram: Placa com buzzer para tocar música e placa de controle do eletroímã.

A primeira é feita de maneira bem simples, contendo somente um buzzer e os fios para conectá-lo ao arduino, juntamente com fios e resistores de 220 Ohms para os leds que piscarão com a música.

A placa de controle do eletroímã é bem simples, e conta somente com um resistor de 330 Ohms, um transistor TIP35 e um diodo, como mostra a figura 6.

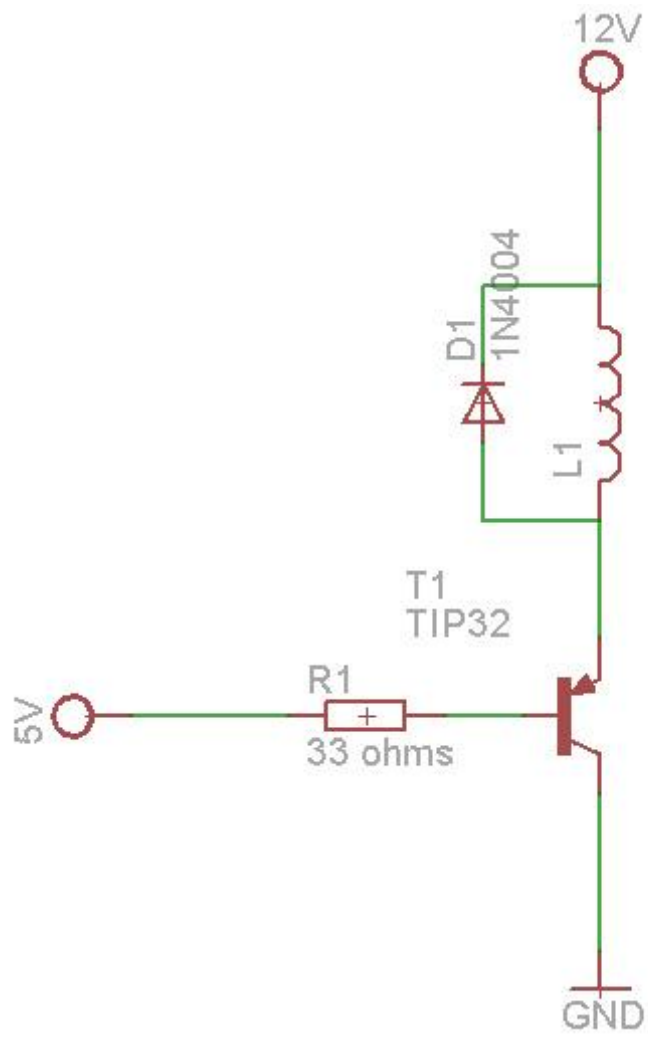


Figura 6 - Esquema para controle do eletroímã.

4.1.2. MECÂNICA

O projeto mecânico partiu da definição dos conceitos e formas básicas da estrutura da esteira, em forma de CAD. (Figura 9).

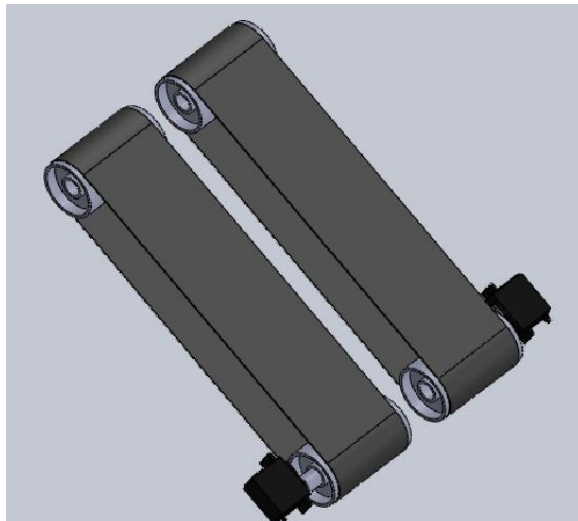


Figura 7 - Desenho feito em CAD de como deveria ser a esteira.

Como mostrado na imagem acima (Figura 9), a primeira ideia era de fazer duas esteiras finas com obstáculos que girassem em velocidades diferentes fazendo assim com que o jogo ficasse mais difícil, mas devido à complexidade de criar dois eixos para a movimentação das mesmas e o excesso de motores para a movimentação ficou decidido que seria uma esteira mais larga com obstáculos em três dimensões.

Para a criação da esteira utilizou-se um tapete felpudo com uma espessura de mais o menos onze centímetros a fim de aperfeiçoar o cenário do jogo (Figura 8).



Figura 8 - Esteira do jogo, feita de tapete.

O próximo passo foi à criação e a fixação dos primeiros obstáculos, feitos em papelão. Foram confeccionados dois tipos básicos de obstáculos, onde em um destes tipos o jogador seria obrigado a pular, pois seria impossível desviar destes obstáculos, e outro que obrigaria o jogador a desviar, pois a altura do obstáculo era maior que o pulo máximo do boneco (Figura 9).



Figura 9 - Obstáculos em 3D de tamanhos diferentes

Como já foi mencionado, os obstáculos em 3D trouxeram diversos problemas e fez-se necessário cortar estes papelões para que se tornassem obstáculos em duas dimensões (Figura 10).



Figura 10 - Obstáculos cortados

Em um primeiro momentos, estes obstáculos foram mantidos para serem utilizados para testes, sendo colocados ímãs em cada um dos obstáculos e uma figura em cima que representasse um “inimigo” do personagem (Figura 11).

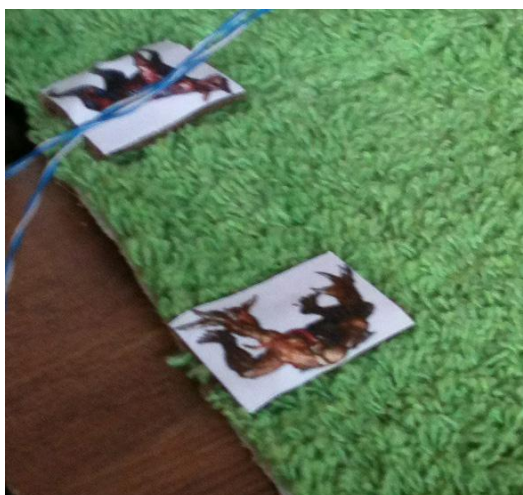


Figura 11 - Obstáculos

Por motivos estéticos todos os obstáculos foram refeitos, partindo do zero, dando vez à versão final dos obstáculos. Optou-se por utilizar personagens do jogo “Super Mario” como adversários, bem como utilizar o personagem Mário como personagem principal controlado pelo jogador. Diferente da primeira vez, estes obstáculos foram feitos com um papel que possui um tipo de espuma dentro com espessura aproximada de 1 cm, o que facilitou a colocação dos ímãs dentro destes, como mostra a figura 12.



Figura 12 - Versão final dos obstáculos

Finalizados os obstáculos, estes foram costurados na esteira com linha de nylon para pesca. O motivo da escolha da linha de pesca foi que esta era forte o suficiente para prender os obstáculos e maleável o suficiente para trabalhar junto com a esteira sem que arrebentasse (Figura 13). Além disso, foi fixado o personagem do Mário na estrutura móvel, colocando um sensor magnético dentro dele para detectar os obstáculos (Figura 14).

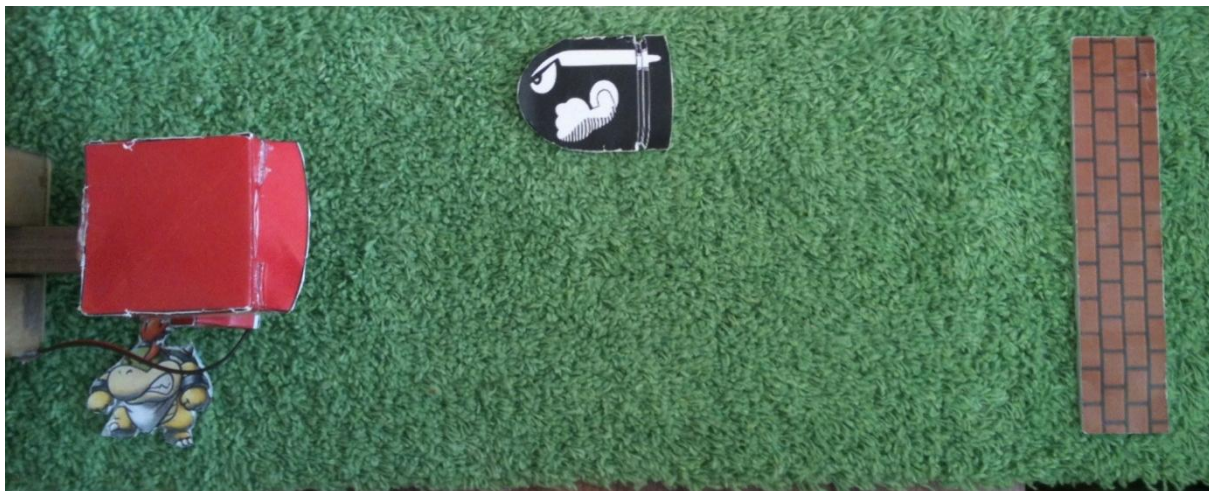


Figura 13 - Obstáculos fixados na esteira

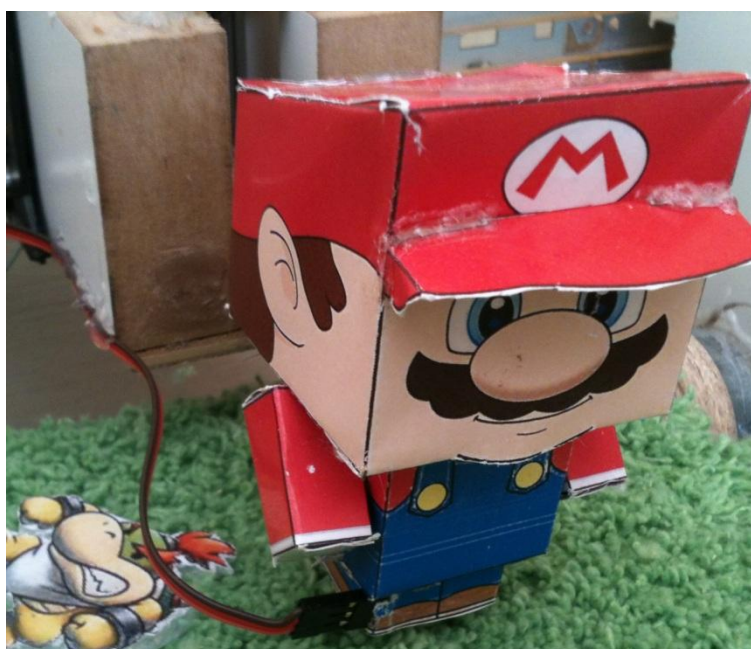


Figura 14 - Personagem principal

Com a esteira pronta, o próximo passo seria projetar e construir a parte de fixação da esteira, ou seja, os dois eixos e os quatro pilares para sustentação dos mesmos. Os dois eixos foram feitos com quatro conectores PVC colados um a um com cola PVC, onde seriam coladas nas suas extremidades rodinhas retiradas de impressoras para que a esteira não corresse para fora (Figura 15). Feitos os testes, detectamos que o motor girava em falso dentro destas rodinhas, nos obrigando a buscar outra maneira de fazer com que a esteira não corra para o lado. A solução foi construir rodinhas

de madeira, com o encaixe do motor feito sob medida, que nos trouxe desempenho satisfatório.



Figura 15 - Eixo da esteira com rodinhas de impressora

O passo seguinte consistiu em confeccionar a estrutura de movimentação do boneco. Para o movimento na vertical do boneco foi criado um eixo onde seria fixado o personagem do projeto, onde também foi fixado o eletroímã que tem a função de atrair o eixo que contem uma parte metálica. Já para a movimentação na horizontal, o eixo foi fixado a um carrinho de impressora com motor DC com dois fins de curso para delimitar as extremidades da esteira fazendo assim com que ele consiga desviar ou saltar os obstáculos (Figura 16).



Figura 16 - Eixo de fixação do boneco

Para que o usuário pudesse controlar o personagem principal foi criada uma caixa contendo um joystick para a movimentação do carrinho de impressora e um botão que aciona o eletroímã, fazendo com que o boneco salte. (Figura 17).



Figura 17 - Caixa de controle do usuário

Finalizando a parte mecânica, foi criada uma caixa para colocar todos os componentes do projeto, atendendo assim o propósito do projeto, que se denomina game in a box. Dentro desta caixa, foi feito uma caixinha onde ficam depositadas as moedas inseridas pelos usuários. Feito isso, o último passo foi fazer o fundo falso para esconder todos os fios e placas dentro da caixa. A figura 18 mostra a caixa finalizada.

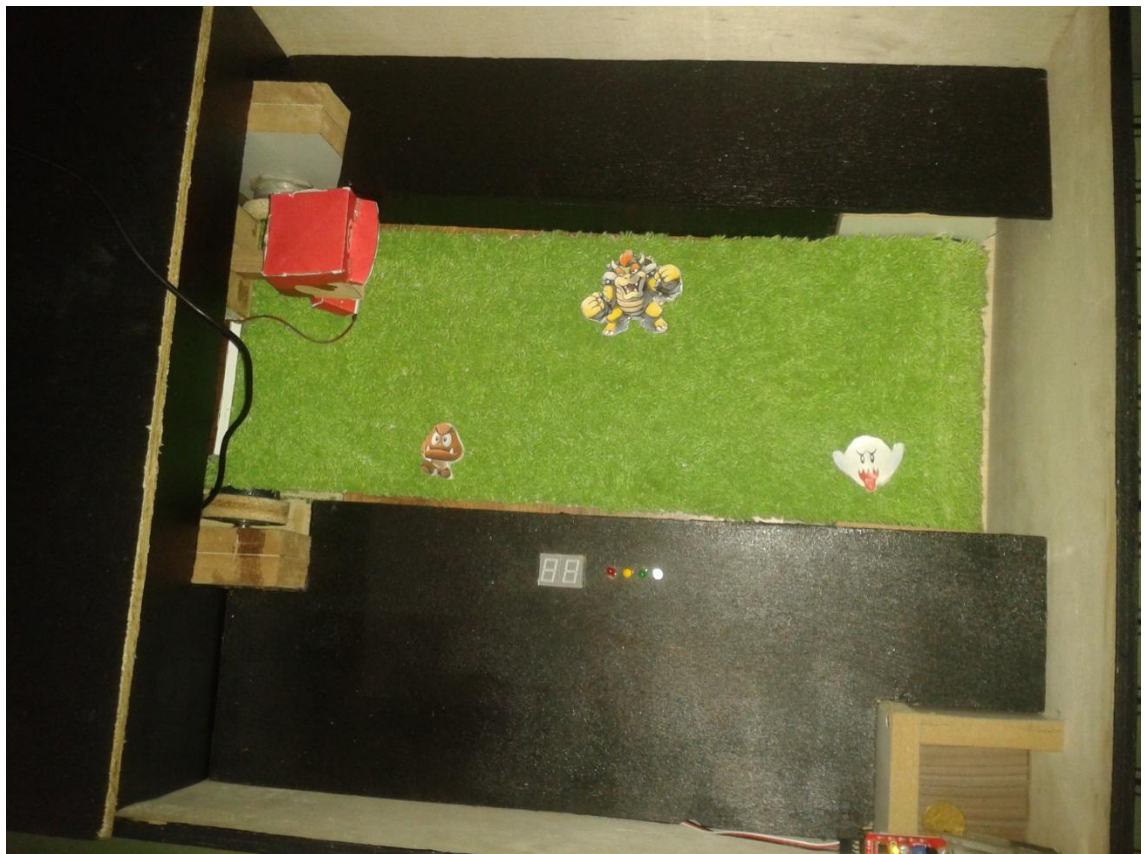


Figura 18 - Caixa finalizada

4.1.3 SOFTWARE

O passo inicial para o desenvolvimento do software foi a criação de um fluxograma que retratasse as principais características que o programa deveria ter. (Figura 19.)

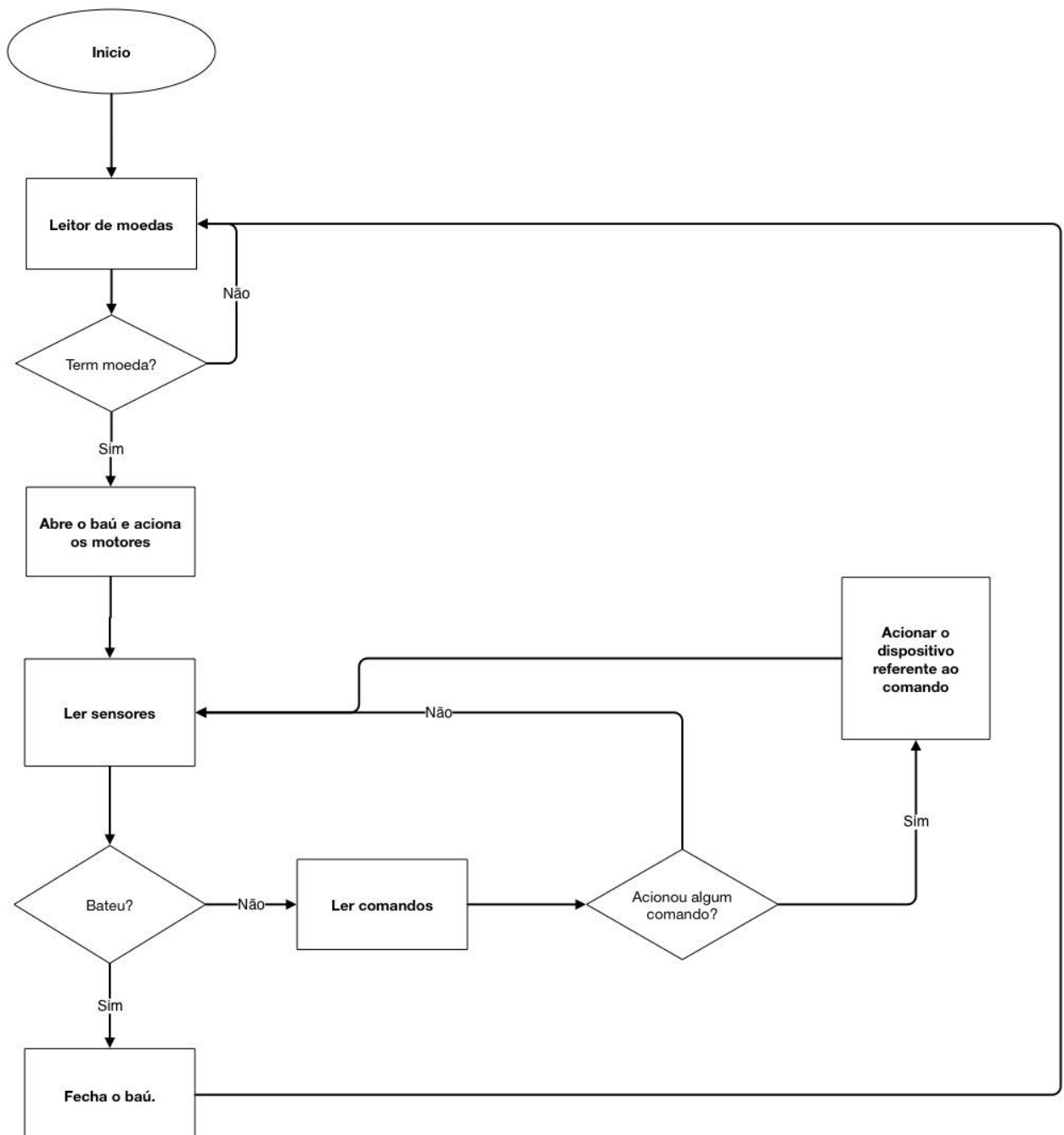


Figura 19

O software do projeto é bastante simples, porém houve uma modificação da ideia inicial devido a modificações dos outros módulos também.

Inicialmente, a ideia era de que ao colocar uma moeda, seriam acionados servo-motores que abririam a caixa do jogo, permitindo assim que o jogador jogue. Porém, devido às proporções inesperadas que os módulos mecânicos acabaram tomando se tornou impraticável esta abertura com motores, visto que um único servo, e talvez até dois, não daria conta de erguer a tampa. Por isso a tampa se tornou de abertura manual mesmo, sem que seja controlado por software.

Explicando objetivamente, o software funciona da seguinte maneira: Ao ligar o arduino, o programa fica dentro de um looping fazendo leitura do sensor infravermelho que identifica se foi inserida a moeda para dar início ao jogo. Inserida a moeda, é tocada a música de início do jogo e são ligados os motores que movimentam a esteira e os displays de pontos do jogador. Feito isso, o programa fica dentro de uma estrutura de repetição que faz a seguinte sequência: Lê o sensor magnético para detectar se houve colisão. Caso o sensor não a identifique, o código passa a ler os comandos do usuário, que podem ser de ir pra frente, pra trás ou pular os obstáculos, voltando a ler os sensores após o movimento. Caso haja a colisão, o software indica fim de jogo, parando a contagem de ponto e as esteiras, voltando assim a ler o sensor infravermelho para detectar se há moedas para o início de um novo jogo.

5 CÓDIGO FONTE

5.1 MAIN

```
#define sensmoed 43 // Sensor infra vermelho detector de moeda

#define sensmag 45 // Sensor magnetico de obstaculos

#define fdcf 35 // Fim de curso da frente do carrinho

#define fdct 33 // Fim de curso de tras do carrinho

#define botao 41 // Botao de pulo do boneco

#define joyf 39 // Fim de curso frente do joystick

#define joyt 37 // Fim de curso tras do joystick

#define mt1 27 // Sinal do motor

#define mt2 25 // Sinal do motor

#define car1 29 // Carrinho de impressora

#define car2 31 // Carrinho de impressora

#define ledverm 47 // Led vermelho da musica

#define ledama 49 // Led amarelo da musica

#define ledver 51 // Led verde da musica

#define ledaltb 50 // Led de alto brilho da musica

#define buzzer 52 // Buzer que vai tocar a musica

#define eletroima1 23 // Pinos do eletroima

#define gndb 53 // Gnd digital

#define vccd 42 // Vcc display de pontos

#define clrd 46 // Clear do display de pontos

#define enad 44 // Enable do display de pontos

int tempo;

int tempoant;

boolean firstjump;
```



```
void setup()
{
  pinMode(sensmoed, INPUT);
  pinMode(sensmag, INPUT);
  pinMode(fdcf, INPUT);
  pinMode(fdct, INPUT);
  pinMode(botao, INPUT);
  pinMode(joyf, INPUT);
  pinMode(joyt, INPUT);
  pinMode(mt1, OUTPUT);
  pinMode(mt2, OUTPUT);
  pinMode(car1, OUTPUT);
  pinMode(car2, OUTPUT);
  pinMode(ledverm, OUTPUT);
  pinMode(ledama, OUTPUT);
  pinMode(ledver, OUTPUT);
  pinMode(ledaltb, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(eletoima1, OUTPUT);
  pinMode(gndb, OUTPUT);
  pinMode(vccd, OUTPUT);
  pinMode(clrd, OUTPUT);
  pinMode(enad, OUTPUT);
  Serial.begin(9600);
  digitalWrite(clrd, HIGH);
  digitalWrite(enad, LOW);
}

void loop()
{
  tempoant=0;
  firstjump = true;
  delay(2000);
```

```
Leiturnoeda();  
Tocamusica();  
digitalWrite(clrd, LOW);  
digitalWrite(enad, HIGH);  
digitalWrite(vccd, HIGH);  
Loopingprincipal();  
}
```

5.2 LEITURA DE MOEDA

```
void LeituraMoeda()
{
    boolean leitura = false;
    do{
        if(!digitalRead(sensoMoeda))
        {
            leitura = true;
        }
        else
        {
            leitura = false;
        }
    }while(!leitura == true);
}
```

5.3 MÚSICA

```
void Tocamusica()
{
  digitalWrite(gndb, LOW);
  Serial.println("Tocando musica");
  tone(buzzer, 440, 300);
  digitalWrite(ledverm, HIGH);
  delay(200);
  tone(buzzer,0, 2000);
  digitalWrite(ledverm, LOW);
  delay(100);
  tone(buzzer, 440, 300);
  digitalWrite(ledverm, HIGH);
  delay(200);
  tone(buzzer,0, 2000);
  digitalWrite(ledverm, LOW);
  delay(100);
  tone(buzzer, 493.8833, 300);
  digitalWrite(ledama, HIGH);
  delay(200);
  tone(buzzer, 0, 2000);
  digitalWrite(ledama, LOW);
  delay(100);
  tone(buzzer, 554.3653, 300);
  digitalWrite(ledver, HIGH);
  delay(200);
  tone(buzzer, 0, 2000);
  digitalWrite(ledver, LOW);
  delay(100);
  tone(buzzer, 440, 300);
  digitalWrite(ledverm, HIGH);
  delay(200);
  tone(buzzer, 0, 2000);
  digitalWrite(ledverm, LOW);
```

```
delay(100);  
tone(buzzer, 554.3653, 300);  
digitalWrite(ledver, HIGH);  
delay(200);  
tone(buzzer, 0, 2000);  
digitalWrite(ledver, LOW);  
delay(100);  
tone(buzzer, 493.8833,300);  
digitalWrite(ledama, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledama, LOW);  
delay(100);  
tone(buzzer, 329.6276, 300);  
digitalWrite(ledaltb, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledaltb, LOW);  
delay(100);  
tone(buzzer, 440, 300);  
digitalWrite(ledverm, HIGH);  
delay(200);  
tone(buzzer,0, 2000);  
digitalWrite(ledverm, LOW);  
delay(100);  
tone(buzzer, 440, 300);  
digitalWrite(ledverm, HIGH);  
delay(200);  
tone(buzzer,0, 2000);  
digitalWrite(ledverm, LOW);  
delay(100);  
tone(buzzer, 493.8833, 300);  
digitalWrite(ledama, HIGH);  
delay(200);
```

```
tone(buzzer, 0, 2000);  
digitalWrite(ledama, LOW);  
delay(100);  
tone(buzzer, 554.3653, 300);  
digitalWrite(ledver, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledver, LOW);  
delay(100);  
tone(buzzer, 440, 2000);  
digitalWrite(ledama, HIGH);  
delay(500);  
tone(buzzer,0,2000);  
digitalWrite(ledama, LOW);  
delay(100);  
tone(buzzer,415.3047, 2000);  
digitalWrite(ledaltb, HIGH);  
delay(500);  
tone(buzzer,0,2000);  
digitalWrite(ledaltb, LOW);  
delay(100);  
tone(buzzer, 440, 300);  
digitalWrite(ledverm, HIGH);  
delay(200);  
tone(buzzer,0, 2000);  
digitalWrite(ledverm, LOW);  
delay(100);  
tone(buzzer, 440, 300);  
digitalWrite(ledverm, HIGH);  
delay(200);  
tone(buzzer,0, 2000);  
digitalWrite(ledverm, LOW);  
delay(100);  
tone(buzzer, 493.8833, 300);
```

```
digitalWrite(ledama, HIGH);  
delay(200);  
tone(buzzer, 0, 2000);  
digitalWrite(ledama, LOW);  
delay(100);  
tone(buzzer, 554.3653, 300);  
digitalWrite(ledver, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledver, LOW);  
delay(100);  
tone(buzzer, 587.3295, 300);  
digitalWrite(ledaltb, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledaltb, LOW);  
delay(100);  
tone(buzzer, 554.3653, 300);  
digitalWrite(ledver, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledver, LOW);  
delay(100);  
tone(buzzer,493.8833,300);  
digitalWrite(ledama, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledama, LOW);  
delay(100);  
tone(buzzer,440, 300);  
digitalWrite(ledverm, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledverm, LOW);
```

```
delay(100);  
tone(buzzer,415.3047, 300);  
digitalWrite(ledaltb, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledaltb, LOW);  
delay(100);  
tone(buzzer,329.6376, 300);  
digitalWrite(ledverm, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledverm, LOW);  
delay(100);  
tone(buzzer,369.9944, 300);  
digitalWrite(ledama, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledama, LOW);  
delay(100);  
tone(buzzer,415.3047,300);  
digitalWrite(ledver, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledver, LOW);  
delay(100);  
tone(buzzer, 440, 2000);  
digitalWrite(ledverm, HIGH);  
delay(500);  
tone(buzzer,0,2000);  
digitalWrite(ledverm, LOW);  
delay(100);  
tone(buzzer, 440, 300);  
digitalWrite(ledverm, HIGH);  
digitalWrite(ledama, HIGH);
```



```
digitalWrite(ledver, HIGH);  
digitalWrite(ledaltb, HIGH);  
delay(200);  
tone(buzzer,0,2000);  
digitalWrite(ledver, LOW);  
digitalWrite(ledama, LOW);  
digitalWrite(ledver, LOW);  
digitalWrite(ledaltb, LOW);  
delay(500);  
tone(buzzer, 440, 2000);  
digitalWrite(ledver, HIGH);  
delay(500);  
tone(buzzer, 0, 2000);  
digitalWrite(ledver, LOW);  
delay(500);  
tone(buzzer, 440, 2000);  
digitalWrite(ledama, HIGH);  
delay(500);  
tone(buzzer, 0, 2000);  
digitalWrite(ledama, LOW);  
delay(500);  
tone(buzzer, 659.2551, 2000);  
digitalWrite(ledver, HIGH);  
delay(1100);  
tone(buzzer, 0, 2000);  
digitalWrite(ledver, LOW);  
Serial.println("Fim da musica");  
}
```

5.4 LOOPING PRINCIPAL

```
void Loopingprincipal()
{
    boolean bateu = false;    // Booleano utilizado para testar se o sensor magnetico identificou obstaculos, o que
    indica colisao

    Serial.println("Rodando motor");
    digitalWrite(mt1, HIGH);
    digitalWrite(mt2, LOW);
    do
    {
        bateu = Lersensores();
        if(!bateu)
        {
            Lercomandos();
        }
        if(bateu)
        {
            Fimdejogo();
        }
    }while(!bateu);
}
```

5.5 LEITURA DE SENSORES

```
boolean Lersensores()  
{  
  if(digitalRead(sensmag))  
  {  
    Serial.println("bateu");  
    return true;  
  }  
  else  
    return false;  
}
```

5.6 LEITURA DE COMANDOS

```
void Lercomandos()
{
  if(digitalRead(joyf))
  {
    do
    {
      digitalWrite(car1, HIGH);
      digitalWrite(car2, LOW);
    }while(!digitalRead(fdcf));
    digitalWrite(car1, LOW);
    digitalWrite(car2, LOW);
  }

  if(digitalRead(joyt))
  {
    do
    {
      digitalWrite(car1, LOW);
      digitalWrite(car2, HIGH);
    }while(!digitalRead(fdct));
    digitalWrite(car1, LOW);
    digitalWrite(car2, LOW);
  }

  if(digitalRead(botao))
  {
    if(firstjump)
    {
      digitalWrite(eletoirma1, HIGH);
      delay(600);
      digitalWrite(eletoirma1, LOW);
      firstjump = false;
    }
  }
}
```

```
else
{
  tempo = millis();
  if(tempo-tempoant>=500)
  {
    Serial.println(tempo-tempoant);
    digitalWrite(eletoirama1, HIGH);
    delay(600);
    digitalWrite(eletoirama1, LOW);
    tempoant=tempo;
  }
}
}
```

9.7 FIM DE JOGO

```
void Fimdejogo()
{
    digitalWrite(mt1, LOW);
    digitalWrite(mt2, LOW);
    digitalWrite(enad, LOW);
    digitalWrite(clrd, HIGH);
    digitalWrite(vccd, LOW);
    Serial.println("Fim de jogo");
    //
    delay(2000);
    digitalWrite(mt1, HIGH);
    digitalWrite(mt2, LOW);
    delay(600);
    digitalWrite(mt1, LOW);
    digitalWrite(mt2, LOW);
    // Parte do código responsável por "rebobinar" a esteira
}
```

6 PROBLEMAS ENCONTRADOS

PROBLEMAS APRESENTADOS	SOLUÇÕES ENCONTRADAS
A ideia inicial era de construir obstáculos em 3D, porém notou-se que traria diversos problemas de fixação e de detecção do sensor de obstáculos.	Fazer obstáculos em 2D.
Definir qual sensor seria utilizado para detectar os obstáculos.	Após testar sensores Infravermelho, ultrassom, Idr e sensor magnético decidiu-se utilizar o sensor magnético e ímãs em cada obstáculo.
Dimensionar o eletroímã para que tivesse força suficiente para realizar o pulo do personagem.	Ao invés de fazer um eletroímã que empurrasse o boneco para cima foi decidido por construir um solenoide que puxaria o boneco através de um pino metálico no seu interior.
Fontes não conseguiam alimentar todos os motores e eletroímã.	Utilizar uma fonte de computador que fornece até 6A.
Motor girando em falso dentro do eixo da esteira	Construir um eixo com maior área de contato entre o motor e o rolo da esteira
Torque insuficiente do motor.	Utilizar um motor com caixa de redução.
Ineficiência da placa de acionamento do eletroímã.	Descartar projeto anterior e fazer um projeto mais simples e objetivo.
Ruídos provocados pelas chaves mecânicas.	A solução seria adicionar eliminadores de Bounce para cada chave, porém isto não foi feito devido à falta de tempo.

7 CONCLUSÃO

Aplicando os conhecimentos obtidos até o presente momento foi possível alcançar os objetivos traçados para o projeto. A principal proposta do projeto, expressa pela disciplina de RPE, foi percebida de maneira extremamente clara, visto que diversas vezes o grupo se deparou com problemas diferentes dos propostos em sala de aula e, ao buscar soluções, compreendeu os problemas cotidianos de um profissional da área de engenharia.

No início do projeto houve bastante dificuldade com relação à divisão e execução de tarefas em áreas com pouco conhecimento ou fora do escopo proposto pelo curso. Ao concluir o projeto e verificar que seu funcionamento atendeu as expectativas foi gratificante observar a superação das adversidades encontradas inicialmente.

8 REFERÊNCIAS

TOCCI, Ronald J.; Widmer, Neal S. Sistemas Digitais: Princípios e Aplicações. Editora LTC. 7ª edição, 2000.

HALLIDAY, David, RESNIZKY, Silvia, WALKER, Jearl, FUNDAMENTOS DE FISICA, V.3 ELETROMAGNETISMO, Editora: LTC, 4 Edição 1996

Site: <http://electronicpiece.blogspot.com.br/2012/02/buzzer-arduino.html>.
Acesso em: 11 de Setembro de 2012

Site: <http://www.audionasigrejas.org/Apostila/frequencia.htm>. Acesso em: 11 de Setembro de 2012

9 GLOSSÁRIO

L298N/L293D - CI de ponte-h dupla utilizado para controlar até dois motores.

Buzzer - Beeper. Dispositivo de sinalização sonora comumente utilizado em eletrônica.

LED - Diodo emissor de luz, ou do inglês Light Emitting Diode. Utilizado em eletrônica para sinalização visual.

Circuito - Circuito elétrico. Ligação de elementos elétricos de tal forma que forneça pelo menos um caminho fechado para a corrente elétrica.

Board - Nome dado ao desenho a ser impresso na placa de fenolite.

Resistor de pull Down - Resistor utilizado em circuitos elétricos para garantir o estado lógico "baixo".

Capacitor - Dispositivo utilizado para armazenar energia temporariamente.

VCC - Voltagem Corrente Contínua. Sigla utilizada para denominar o ponto de maior potencial do circuito.

GND - Ground (do inglês). Terra. Sigla utilizada para denominar o ponto de menor potencial do circuito.

Fim-de-curso - Sensor de toque. Dispositivo mecânico responsável por indicar ao controlador a interrupção do movimento.

Arduino - Plataforma de prototipagem eletrônica de hardware livre, projetada com um microcontrolador ATmega328. É programado com uma linguagem de programação própria baseada na linguagem C.

PCI/PCB - Placa de circuito impresso ou, do inglês, Printed circuit board. Placa contendo o circuito impresso em uma superfície de cobre.

EagleCAD Soft - Software utilizado para o desenho de circuitos a serem impressos.

CI - Circuito Integrado. Circuito eletrônico miniaturizado.

Ponte-h - Circuito utilizado para controle de motores DC.

Motor DC - Motor alimentado com corrente contínua.

Linguagem de Programação - Método padronizado para comunicar instruções para um computador. Conjunto de regras sintáticas e semânticas usadas para definir um programa de computador.

Função - Desvio de código que agrupa comandos que são utilizados diversas vezes durante o programa