CicloExergame

Jogos Sérios e Elementos de Jogos na Promoção de Engajamento em Contextos de Telerreabilitação de Pacientes





Assuntos Abordados

- 4 Introdução
- ► 13 Arquitetura do Exergame
- 24 Adaptação do Cicloergômetro



Membros

- Cristhinny Sarah Pires Araujo 201801484
- Júlio César Freitas Bueno de Moraes 201801511
- Lucas de Sousa Neves 201801516



Introdução

Cristhinny Sarah



CicloExergame

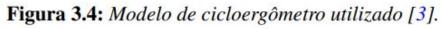
- Os jogos trazem muitas possibilidades para aplicações.
- Em contextos de reabilitação, as abordagens que fazem uso de jogos são utilizadas por serem capazes de despertar o engajamento e a motivação do paciente, trazendo uma dimensão de entretenimento para as atividades que seriam cansativas e repetitivas.
- Nesse caso, estamos considerando cenários onde a reabilitação acontece de forma <u>remota</u> (chamada Telerreabilitação).



CicloExergame

- Para a realização das sessões de Telereabilitação, é utilizado o aparelho cicloergômetro. Este aparelho é uma bicicleta de cabeceira, por meio da qual o paciente pode exercitar seus membros superiores e inferiores.
- Ele é utilizado em casos de disfunções musculares, causadas por <u>inatividade física</u> (em cenários de pós-operatórios ou de recuperação em Unidades de Terapia Intensiva - como é o caso dos pacientes pós-COVID-19), pela ocorrência de um <u>Acidente Vascular Cerebral (AVC)</u> ou ainda <u>problemas</u> <u>cardiorrespiratórios</u>.

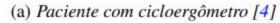








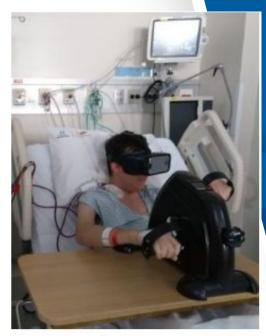






(b) Fisioterapeuta, paciente e cicloergômetro [4]





(c) Paciente, cicloergômetro e óculos de realidade virtual [2]



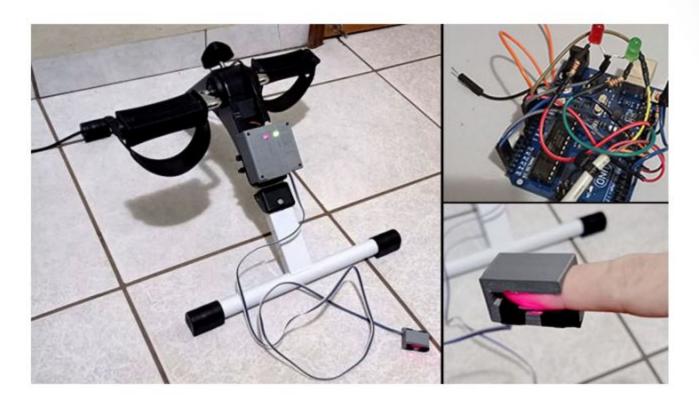


Fig. 1. Cicloergômetro adaptado para o projeto.



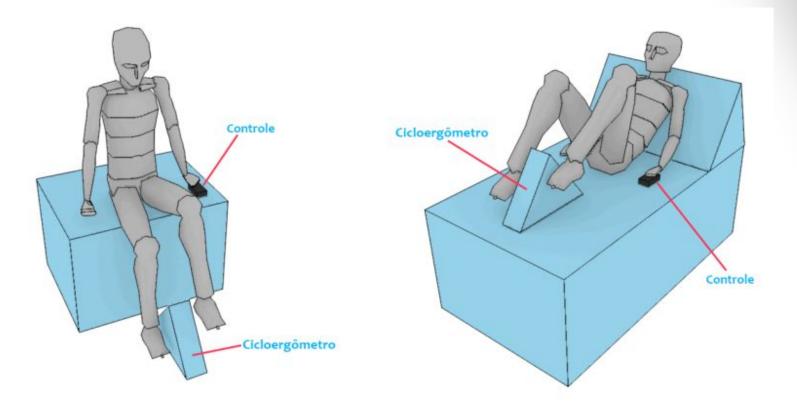
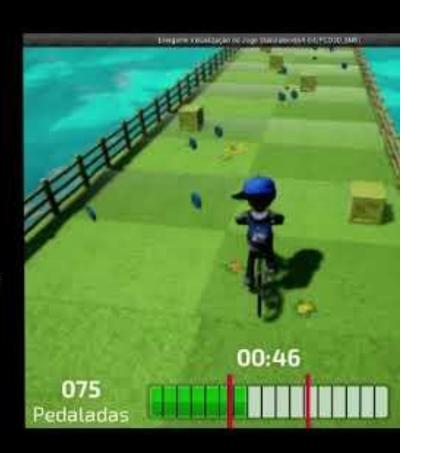


Fig. 5. Posições ergonômicas para a interação com o exergame.

Ele deverá pedalar dentro da velocidade estipulada pelas marcações vermelhas.

Deverá recolher os objetos coletáveis (atingindo a meta de coleta por sessão) e desviar dos obstáculos!



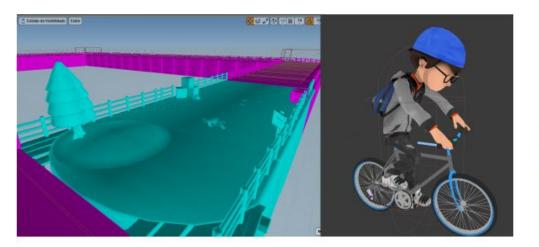


Fig. 4. Cenário e personagem desenvolvidos para o exergame.

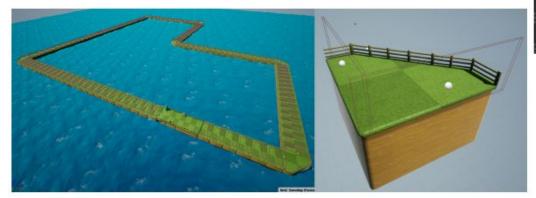


Fig. 6. Estrada cíclica e sistema para curvas.

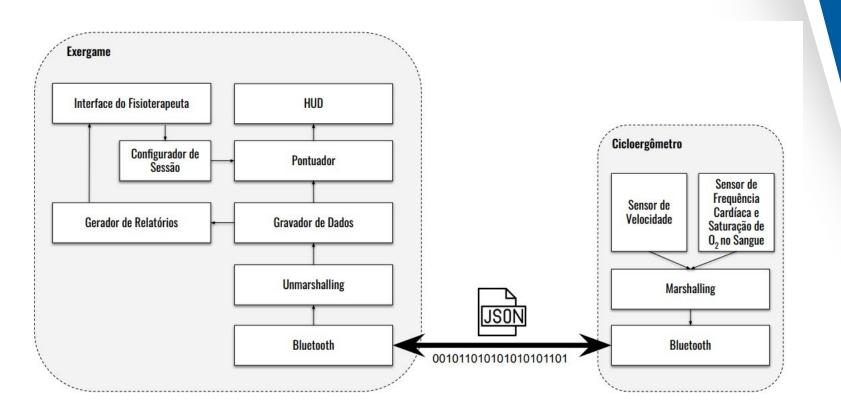




Fig. 7. Obstáculos e objetos coletáveis.



Júlio César









- Sensor de Velocidade: para determinar a velocidade de pedaladas do paciente, o cicloergômetro deve possuir um sensor que contabilize cada giro do pedal. O velocímetro no Exergame é incrementado apenas se houver rotações no pedal. Os dados gerados por esse sensor são enviados ao módulo Marshaling.
- Sensor de Frequência Cardíaca e Saturação de Oxigênio no Sangue: o paciente deve manter a frequência cardíaca e a saturação de oxigênio dentro de um limite, ou seja, não pode ultrapassar determinado valor, sob o risco de ter uma intercorrência médica. Esses dados também são importantes para que o fisioterapeuta possa ver graficamente a evolução durante o exercício.



- Marshaling: uma vez gerados os dados pelos sensores, o módulo Marshaling reúne os dados para envio no formato JSON. O resultado gerado por esse módulo é um JSON que é repassado ao módulo Bluetooth.
- Bluetooth: esse módulo é capaz de realizar a comunicação de dados do Exergame para o cicloergômetro e vice-e-versa.
 Seguindo o fluxo no cicloergômetro, ele recebe o arquivo JSON, serializa e envia ao Exergame. No Exergame, esse dado é de-serializado e transformado em arquivo texto no formato JSON.



- Unmarshaling: Ao receber o texto em formato JSON esse módulo remove os dados de velocidade e batimento cardíaco do JSON e repassa para o gravador de dados.
- Gravador de Dados: nesse módulo os dados são persistidos.
 Essa persistência é necessária já que os dados devem estar disponíveis após o término da sessão de jogo terminar, para que o módulo gerador de relatório possa gerar um histórico de dados durante a sessão.



 Pontuador: verifica a velocidade vinda do cicloergômetro. Os pontos são gerados somente se a velocidade estiver entre os limites configurados pelo fisioterapeuta. O jogador pontua mais se conseguir manter uma velocidade constante dentro dos limites. Essa regra faz parte de um dos requisitos gerados durante a interação do desenvolvedor de jogos e os fisioterapeutas.



- HUD: nos videogames, o HUD (do inglês head-up display) ou a barra de status é o método pelo qual as informações são visualmente transmitidas ao jogador como parte da interface do usuário do jogo. Esse módulo apresenta visualmente a <u>barra de velocidade</u>, o <u>tempo</u> que resta para o fim do exercício, a <u>frequência cardíaca</u> e a <u>saturação de oxigênio</u> no sangue.
- Gerador de Relatórios: esse módulo é responsável por ler os dados persistidos pelo módulo Gravador de Dados, gerar um relatório e enviar para o módulo Interface do Fisioterapeuta para que seja acessado pelo profissional.



Relatório













- Interface do Fisioterapeuta: para configurar e personalizar a sessão de cada jogador o fisioterapeuta tem acesso a uma interface com botões e campos de texto que devem ser preenchidos.
- Configurador de Sessão: gerado pela Interface do Fisioterapeuta o Configurador de Sessão é o módulo capaz de configurar a duração do exercício, velocidade e limites dos dados fisiológicos. A arquitetura auxilia na organização e implementação do Exergame.



Menu de Configuração para o Fisioterapeuta



Figura 3.11: Menu de configuração para o Fisioterapeuta.



Adaptação do Cicloergômetro

Lucas de Sousa







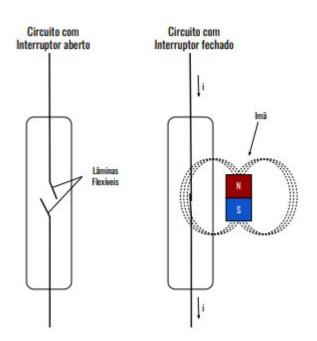


Contador de pedaladas

- A contagem das pedaladas no cicloergômetro é realizada por um Interruptor Magnético ou Reed Switch. Dentro do interruptor há lâminas ferromagnéticas flexíveis que no estado normal estão distantes ou desencostadas. Com a aproximação de um imã essas lâminas se encostam e a corrente elétrica flui pelo interruptor.
- O pedalar fecha e abre o circuito do interruptor magnético que pode ser interpretado como um sinal pelo Arduino. Ao receber um novo sinal o Arduino incrementa um contador interno que pode ser lido como uma contagem de pedaladas. Ao fazer a medição do tempo entre os sinais, pode-se obter também o tempo gasto em cada pedalada e assim servir como parâmetro de velocidade.



Contador de pedaladas





(b) Modelo comercial do Interruptor Magnético

(a) Funcionamento do Interruptor Magnético



Sensor de Batimento Cardíaco

- Dois modelos de sensores foram testados: sensor de frequência cardíaca da pulsesensor.com e o sensor de frequência cardíaca e oxímetro MAX30100.
- Ambos os sensores são baseados no princípio da fotopletismografia, uma técnica óptica que pode ser usada na detecção das alterações no volume sanguíneo. Ele mede a mudança no volume de sangue através da alteração na intensidade da luz. O volume se altera devido ao pulso de pressão, sendo detectado a partir da iluminação da superfície da pele com um diodo emissor de luz (LED) e, em seguida, mede-se a quantidade de luz refletida para um fotodiodo.



Sensor de Batimento Cardíaco

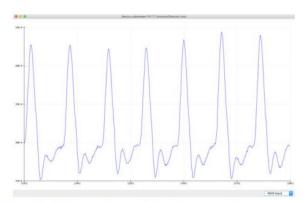


Figura 3.6: Ciclo cardíaco medido pelo sensor de frequência cardíaca.

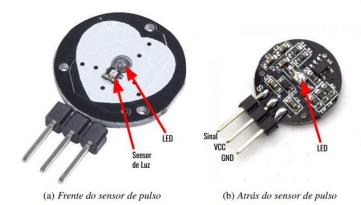


Figura 3.7: Os dois lados do sensor de pulso



Figura 3.8: Resistores removidos do sensor para utilização no Arduino.



Esquema do Arduino

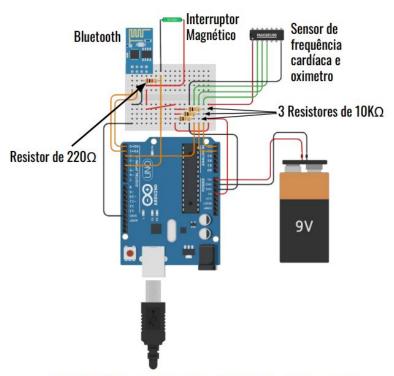


Figura 3.9: Esquema do projeto de adaptação usando o Arduino.



Blocos de Código

Código 1 Configuração inicial do Arduino.

```
#include <Wire.h>
  #include "MAX30100 PulseOximeter.h"
  PulseOximeter Pox;
  int reedSwitch = 4;
  uint32_t tsLastReport = 0;
  void setup()
10
    pinMode (reedSwitch, INPUT);
    Serial.begin(9600);
14
    if (!Pox.begin()){
      Serial.println("FAILED");
    } else{
      Serial.println("SUCCESS");
19
21
```

Código 2 Montagem do pacote JSON e envio ao módulo Bluetooth.

```
#include <ArduinoJson.h>
   #define REPORTING_PERIOD_MS 1000
  void loop()
     Pox.update();
     if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
       heartHeat = Pox.getHeartRate();
10
       spo2value = Pox.getSpO2();
11
       tsLastReport = millis();
12
13
14
     reedDebounce();
15
16
     StaticJsonDocument<256> doc;
17
     JsonObject root = doc.to<JsonObject>();
     root["numCycles"] = numCycles;
     root["cycleTime"] = cycleTime;
20
     root["spo2"] = spo2value;
21
     root["heartRate"] = heartHeat;
     serializeJson(doc, Serial);
23
     Serial.println();
24
25
     delay (50);
26
27
```



Código 3 Função ReedDebounce.

```
void reedDebounce() {
    readingState = digitalRead(reedSwitch);
    if (readingState == HIGH && previousState == LOW
         && millis() - lastTime > debounce) {
      numCycles++;
5
      cvcleTime = millis() - lastTime;
      lastTime = millis();
    previousState = readingState;
10
```

```
Código 4 Leitura do barramento serial pelo Exergame.
class SerialCommunication
private:
        uint8 mPort;
        int32 mBaud;
        void *mhIDComDev;
        OVERLAPPED mOverlappedRead, mOverlappedWrite;
public:
        SerialCommunication(uint8 Port, int32 Baud = 9600);
        ~SerialCommunication();
        bool Open();
        void Close();
        bool Print (FString String);
        bool WriteByte (uint8 Byte);
        bool WriteBytes (TArray < uint8 > Buffer);
        bool ReadByte (uint8 &Byte);
        TArray<uint8> ReadBytes(int32 Limit);
        void Flush():
        bool IsOpened() { return mhIDComDev != NULL; };
```

Código 5 Leitura do barramento serial pelo Exergame.

```
if (mSerial->IsOpened())
      bool bRead = false;
      uint8 Char = 0x0;
      if (mSerial->ReadByte(Char))
           if (Char == '\n')
               auto Convert =
               FUTF8ToTCHAR((ANSICHAR *) mChars.GetData());
               FString JsonString = FString(Convert.Get());
               //Critical section:
              mMutex.Lock();
               mJsonString = JsonString;
               mMutex.Unlock();
               mChars.Empty();
           else
              mChars.Add(Char);
25
```

Obrigado

