Poker Controlado por Voz e Gestos

Miguel Ferreira, Nº 98599
Raquel Lopes, Nº 97500

Interação Multimodal

Objetivos

Aplicação

Free Texas Holdem Poker Game

Interação por Gestos:

Microsoft Kinect

Interação por Voz:

speechModality

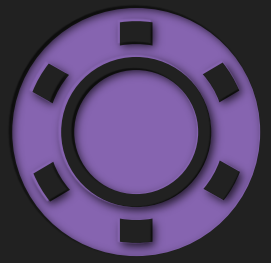
Controlo:

Selenium integrado no Google Chrome

Cenário

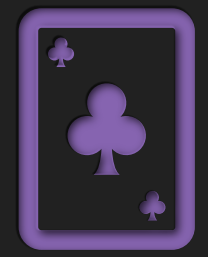
Objetivo do Utilizador	Problema	Solução
O Utilizador pretende jogar uma partida de Poker.	O Utilizador pretende utilizar a voz e gestos para tal.	Desenvolvimento de um Sistema que responda a voz e gestos de forma adequada.

Cenário Final



O Utilizador pretende jogar uma partida de Poker utilizando voz ou gestos, colocando-se em frente a um computador.

O Utilizador levanta o polegar para o computador para iniciar um jogo ou diz “Iniciar Partida!” e o computador inicia uma partida.



O Utilizador realiza uma seleção de gestos e ações por voz para progredir no jogo conforme entender, com resposta adequada do sistema.

Arquitetura

Aplicação de Poker

- Chamar os inputs detetados;
- Aplicar o input à sua função;
- Realizar a função dentro da aplicação, com resposta adequada.

Single Actions

Controlo do Sistema

- Iniciar Jogo
- Reiniciar Partida
- Terminar Partida
- Continuar a Jogar

Apoio ao Jogador

- Ver Cartas na Mão
- Ver Cartas na Mesa
- Valor Total Apostado
- Saldo Atual
- Definir Limite

```
fg.Single(Speech.END, Output.END);
fg.Single(Speech.RESTART, Output.RESTART);
fg.Single(Speech.CONTINUE, Output.CONTINUE);
fg.Single(Speech.ALLIN, Output.ALLIN);
fg.Single(Speech.TABLE, Output.TABLE);
fg.Single(Speech.HAND, Output.HAND);
fg.Single(Speech.POT, Output.POT);
fg.Single(Speech.LIMIT, Output.LIMIT);
fg.Single(Speech.CASH, Output.CASH);
fg.Single(Speech.YES, Output.YES);
```

Redudancy Actions

Iniciar Jogo

Fold (Desistir)

Check (Passar)

Raise (Aumentar)

Call (Igualar)

```
fg.Redundancy(Speech.START, SecondMod.START, Output.START);
fg.Redundancy(Speech.CHECK, SecondMod.CHECK, Output.CHECK);
fg.Redundancy(Speech.CALL, SecondMod.CALL, Output.CALL);
fg.Redundancy(Speech.RAISE, SecondMod.RAISE, Output.RAISE);
fg.Redundancy(Speech.FOLD, SecondMod.FOLD, Output.FOLD);
```

Single Actions

Ver Cartas na Mesa

```
private void cardsInTable()
{
    List<String> cards_in_table = new List<String>();
    cards_in_table.Add(driver.FindElement(By.Id("flop1")).GetAttribute("style").Split('/')[2].Split('.')[0].ToString());
    cards_in_table.Add(driver.FindElement(By.Id("flop2")).GetAttribute("style").Split('/')[2].Split('.')[0].ToString());
    cards_in_table.Add(driver.FindElement(By.Id("flop3")).GetAttribute("style").Split('/')[2].Split('.')[0].ToString());
    cards_in_table.Add(driver.FindElement(By.Id("turn")).GetAttribute("style").Split('/')[2].Split('.')[0].ToString());
    cards_in_table.Add(driver.FindElement(By.Id("river")).GetAttribute("style").Split('/')[2].Split('.')[0].ToString());
    List<String> final_cards = new List<String>();
    for (int i = 0; i < cards_in_table.Count; i++)
    {
        if (cards.ContainsKey(cards_in_table[i]))
        {
            final_cards.Add(cards_in_table[i]);
        }
    }
    String cardsString = "";
    for(int i = 0; i < final_cards.Count; i++)
    {
        cardsString = cardsString + ", " + cards[final_cards[i]];
    }
    if(cardsString == "")
    {
        System.Diagnostics.Debug.WriteLine("A mesa está vazia.");
        call_tts("A mesa está vazia.");
    }
    else
    {
        System.Diagnostics.Debug.WriteLine("A mesa contém " + cardsString);
        call_tts("A mesa contém " + cardsString);
    }
}
```

Ver cartas na Mão

```
private void cardsInHand()
{
    var textC1 = driver.FindElement(By.XPath("//*[@id=\"seat0\"]/div[1]/div[1]")).GetAttribute("style");
    var card1 = textC1.Split('/')[2].Split('.')[0].ToString();

    var textC2 = driver.FindElement(By.XPath("//*[@id=\"seat0\"]/div[1]/div[2]")).GetAttribute("style");
    var card2 = textC2.Split('/')[2].Split('.')[0].ToString();

    var textCardC1 = cards[card1];
    var textCardC2 = cards[card2];

    System.Diagnostics.Debug.WriteLine("A sua mão contem " + textCardC1 + " e " + textCardC2);
    call_tts("A sua mão contem " + textCardC1 + " e " + textCardC2);
}
```

Single Actions

Valor Total Apostado

```
case "POT":
    if (driver.FindElements(By.Id("total-pot")).Count() > 0)
    {
        String pot_total = driver.FindElement(By.Id("total-pot")).Text;
        System.Diagnostics.Debug.WriteLine("O valor total apostado atualmente é " + pot_total.Split('$')[1] + "dólares");
        call_tts("O valor total apostado atualmente é " + pot_total.Split('$')[1] + "dólares");
    }
    //procurar valor no id="total-pot"
    break;
```

Saldo Atual

```
private int get_Current_Cash()
{
    if (driver.FindElements(By.XPath("//*[@id=\"seat0\"]/div[2]/div[2]")).Count() > 0)
    {
        return int.Parse(driver.FindElement(By.XPath("//*[@id=\"seat0\"]/div[2]/div[2]")).Text.Split('$')[1]);
    }
    return -1;
}
```

Definir Limite

```
case "LIMIT":
    if (limit_flag == true)
    {
        if (json.recognized[0].ToString().Contains("NUMBERS"))
        {
            var numero = (json.recognized[0].ToString().Split('S')[1]);
            System.Diagnostics.Debug.WriteLine("Limite de " + numero + " dólares definido.");
            limit_flag = false;
            this.bet_limit = int.Parse(numero);

        }
    }
    else
    {
        System.Diagnostics.Debug.WriteLine("Qual é o valor que pretende definir como limite?");
        limit_flag = true;
    }
    break;
private void verify_Limit()
{
    if ((this.currentCash != -1) && this.bet_limit != -1)
    {
        if ((500 - this.currentCash) > this.bet_limit)
        {
            System.Diagnostics.Debug.WriteLine("Atenção! Já atingiu o seu limite de apostas.");
        }
    }
}
```

Exemplo de Redudancy

Iniciar Jogo

```
<rule id="Start">
    <example> Quero começar o jogo! </example>
    <example> Pretendo iniciar o jogo! </example>
    <example> Inicia um novo jogo! </example>
    <example> Começa a jogar. </example>
    <example> Gostava de começar a jogar. </example>

    <item repeat="0-1">Quero </item>
    <item repeat="0-1">Pretendo</item>
    <item repeat="0-1">Gostava de</item>
    <one-of>
        <item> começar</item>
        <item> iniciar</item>
        <item> começa</item>
        <item> inicia</item>
    </one-of>
    <item repeat="0-1"> jogo </item>
    <item repeat="0-1"> novo jogo </item>
    <item repeat="0-1"> jogar</item>
</rule>
```

Fold (Desistir)

```
<rule id="Fold">
    <example>quero desistir</example>
    <example>quero sair</example>
    <example>desisto</example>

    <item repeat="0-1">Quero </item>
    <item repeat="0-1">Pretendo</item>
    <item repeat="0-1">Gostava de</item>
    <one-of>
        <item>desistir</item>
        <item>sair</item>
        <item>desisto</item>
    </one-of>
</rule>
```

Exemplo de Redudancy

Check (Passar)

```
<rule id="Check">
    <example> Quero passar a aposta! </example>
    <example> Passa ao próximo jogador! </example>

    <item repeat="0-1">Quero </item>
    <item repeat="0-1">Pretendo</item>
    <item repeat="0-1">Gostava de</item>
    <one-of>
        <item> passar</item>
        <item> passa</item>

    </one-of>
    <item repeat="0-1"> aposta</item>
    <item repeat="0-1"> próximo</item>
    <item repeat="0-1"> jogador</item>
</rule>
```

Call (Igualar)

```
<rule id="Call">
    <example> Igualo! </example>
    <example> Quero igualar a aposta! </example>

    <item repeat="0-1">Quero </item>
    <item repeat="0-1">Pretendo</item>
    <item repeat="0-1">Gostava de</item>
    <one-of>
        <item> igualar</item>
        <item> igualo</item>
        <item> pago</item>

    </one-of>
    <item repeat="0-1"> aposta</item>
</rule>
```

Exemplo de Redudancy

Raise (Aumentar)

O utilizador pode dizer "Aumentar aposta", sendo questionado pelo programa qual o valor que deseja apostar. O utilizador procede, então, a dizer o valor para o qual pretende aumentar ("cinquenta", por exemplo) e a aposta é colocada.

```
<rule id="Raise">
    <example> Aumentar a aposta! </example>
    <example> Apostar mais! </example>
    <example> Subir a aposta! </example>

    <item repeat="0-1">Quero </item>
    <item repeat="0-1">Pretendo</item>
    <item repeat="0-1">Gostava de</item>
    <one-of>
        <item> aumentar</item>
        <item> subir</item>
        <item> apostar</item>

    </one-of>
    <item repeat="0-1"> aposta</item>
    <item repeat="0-1"> mais</item>
</rule>
```

Exemplo de Gesto

Check (Passar)

Passar a mão horizontalmente: gesto indicativo de que o Utilizador pretende passar (dar check) no decorrer do jogo.

Demonstração
Demo

Miguel Ferreira, Nº 98599
Raquel Lopes, Nº 97500

Interação Multimodal