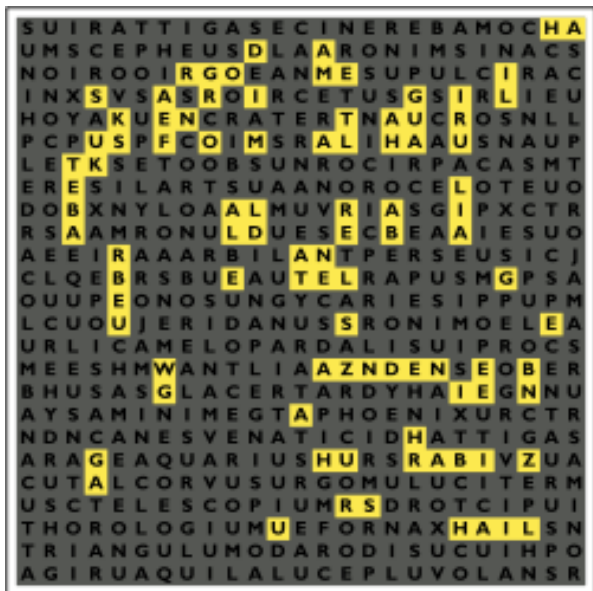


Caça Palavras



Aldebaran • Betelgeuse • Deneb • Fomalhaut • Gienah Ghurab • Gomeisa
• Hadar • Izar • Kelb Alrai • Rigil Kentaurus • Suhail • Wazn

Caçar uma palavra em um monte de letras embaralhadas pode ser um passatempo divertido para muitas pessoas. Além de estimular o raciocínio lógico, serve como forma de auxiliar na educação de crianças que estão aprendendo a ler até aquelas que já sabem ler, como forma de melhorar o português, ajudando os adultos, também, nessa questão.

Originária no antigo Egito foi modernizada pelo jornalista Arthur Wynne da New York Times em 1913, chegando ao Brasil em 1925 onde ganhou mais força em 1948 quando a Ediouro Publicações lançou sua primeira revista de palavras cruzadas. Hoje, com a tecnologia em alta, pode-se encontrar vários aplicativos de caça-palavras gratuitos para smartphones e em várias línguas.

O jogo de **caça-palavras**, ou **sopa de letras**, é um *passatempo* que consiste de letras arranjadas aparentemente aleatórias em uma grade quadrada ou retangular. O objetivo do jogo é encontrar e circundar as palavras escondidas na grade tão rapidamente quanto possível. As palavras podem estar escondidas verticalmente, horizontalmente ou diagonalmente dentro da grade. As palavras são arranjadas normalmente de modo que possam ser lidas da esquerda para a direita ou de cima para baixo, sendo que em passatempos de maior dificuldade também pode ocorrer o oposto. Algumas vezes uma lista de palavras escondidas é fornecida, mas os passatempos mais desafiadores podem fazer com que o jogador as descubra. A maioria dos passatempos de caça-palavras tem também um tema comum a qual todas as palavras escondidas estão relacionadas.

Os caça-palavras são normalmente encontrados em **jornais diários e revistas de passatempo**. Alguns professores usam-nos como ferramentas educacionais para crianças, o benefício que é que as mentes jovens podem aprender palavras novas e a soletrá-las através da busca intensa, procurando-as letra por letra no passatempo, além de entreter, desenvolve o raciocínio lógico e a memória, ajudando a evitar mal de Alzheimer.

Estratégias

A melhor estratégia para se encontrar todas as palavras é fazer uma varredura no passatempo da esquerda para a direita (ou vice-versa) e procurar a primeira letra da palavra (se uma lista de palavras é fornecida). Depois que se encontrou a letra, deve olhar as oito letras circunvizinhas para ver se a letra seguinte da palavra está lá e então continuar até que a palavra inteira seja encontrada. Uma outra estratégia é procurar letras que se destacam dentro da palavra que se está procurando (se uma lista de palavras é fornecida). Uma vez que a maioria dos caça-

palavras usam letras maiúsculas, é mais fácil encontrar as letras que se destacam entre as outras. Estas letras são: Q, O, U, X, e Z.

Por último, a estratégia de procurar letras dobradas na palavra que você está procurando (se uma lista da palavra é fornecida) pode ajudar, porque é mais fácil de se identificar duas letras idênticas lado a lado dentro de uma grande grade de letras aleatórias.

Se uma lista da palavra não for fornecida, uma boa maneira de se encontrar rapidamente palavras é ir linha por linha. Primeiramente, todas as linhas horizontais devem ser lidas de frente para trás e vice-versa, e então deve se ler as colunas vertical, etc.

Mensagem secreta em caça-palavras

Um caça-palavras pode conter uma mensagem secreta: outras palavras ou frases que não constem da lista, colocadas propositadamente para a pessoa as encontrar.

Motivação

Este jogo foi escolhido para ser implementado pelo fato de estar a muito tempo presente na vida das pessoas independentemente de raça, cor, classe financeira ou social, atraindo qualquer faixa etária. Tendo como requisito mínimo necessário que o usuário tenha a capacidade de ler para que consiga identificar as palavras escondidas no quadro. Além disso é um jogo consolidado como passatempo e método de ensino, mesmo com as mudanças e influencias tecnológicas dos últimos anos. É um ótimo jogo de interação com o usuário, pois além de prender a sua atenção, permite estimular o raciocínio lógico como foi dito anteriormente sobre a utilidade do mesmo.

Categoria

A categoria que este jogo está inserido reúne os jogos baseados em palavras.

Características da Implementação

OBJETIVO

Encontrar o máximo de palavras antes que o tempo se esgote.

- 4 letras = 1 ponto;
- 5 letras = 3 pontos;
- 6 ou mais letras = 5 pontos!

REGRAS

- As palavras têm que ter pelo menos 3 letras.
- Forme palavras utilizando letras que estejam juntas nas direções: Vertical ou horizontal.
- Desconsidere acentos ao formar palavras. (Um "Ç" pode ser "C" ou um "Ã" pode ser "A".)
- Forme qualquer palavra no plural ou singular
- Não valem nomes próprios ou palavras estrangeiras.

TECNOLOGIA UTILIZADA – SOCKETS TCP/IP

Diversas aplicações que utilizamos no dia-a-dia fazem uso de sockets para se comunicar. Nosso navegador web utiliza sockets para requisitar páginas. Quando um sistema se integra com um banco de dados ele abre um socket. Quando fazemos um SSH (**Secure Shell**) em um servidor estamos abrindo e utilizando um socket. Os sockets foram criados na forma de uma API que possibilita aplicações/processos se comunicarem.

Abaixo listamos alguns das principais funções utilizadas ao criar um programa utilizando sockets:

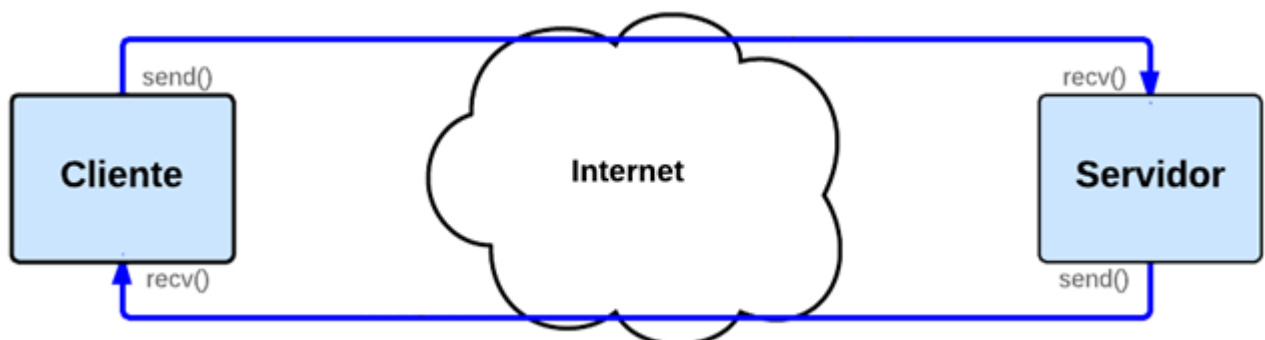
```
/**
 * Principais funções para escrever programas com sockets
 */

getaddrinfo() // Traduz nomes para endereços sockets
socket()       // Cria um socket e retorna o descritor de arquivo
bind()         // Associa o socket a um endereço socket e uma porta
connect()      // Tenta estabelecer uma conexão com um socket
listen()       // Coloca o socket para aguardar conexões
accept()       // Aceita uma nova conexão e cria um socket
send()         // caso conectado, transmite mensagens ao socket
recv()         // recebe as mensagens através do socket
close()        // desaloca o descritor de arquivo
shutdown()     // desabilita a comunicação do socket
```

Utilizamos a seguinte estrutura para implementar o jogo usando a tecnologia Socket-TCP/IP:

CLIENTE

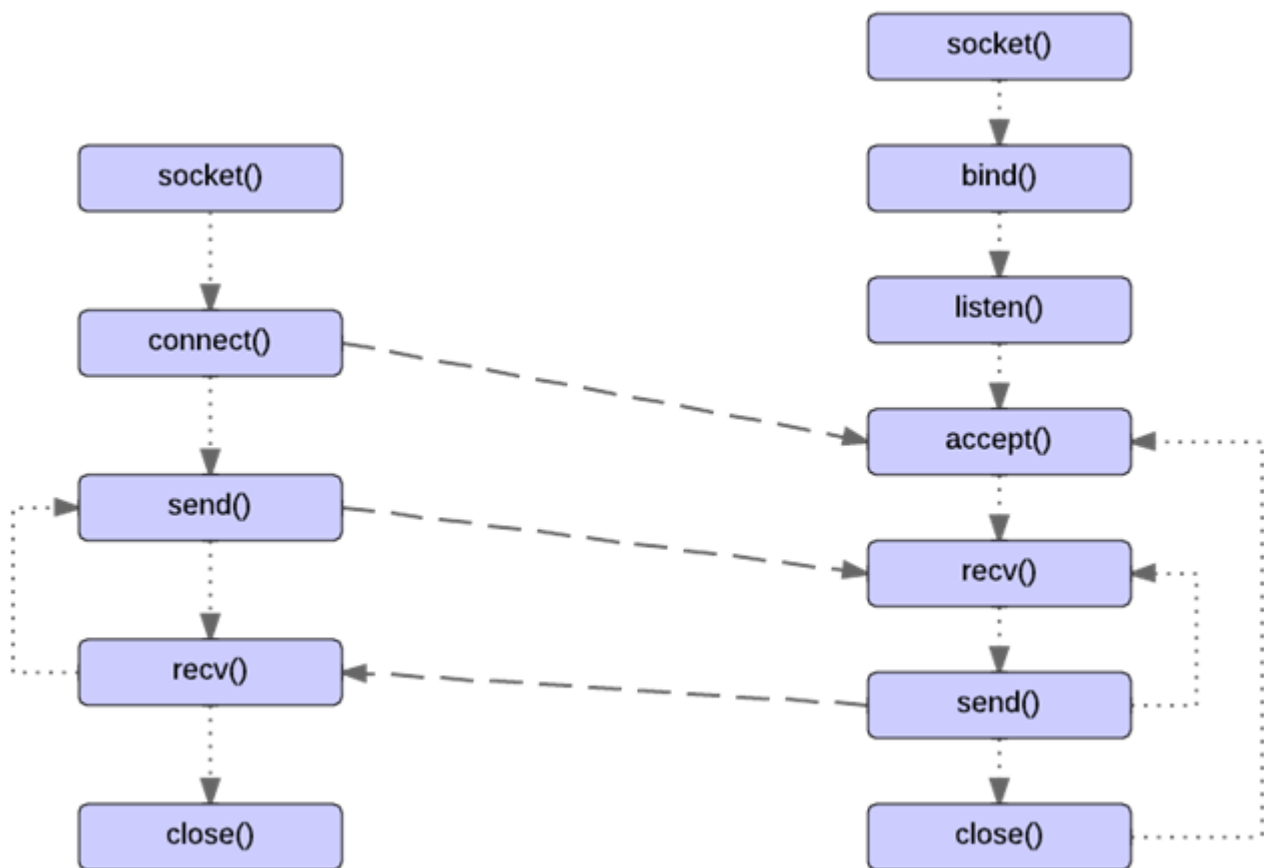
O programa cliente primeiro cria um socket através da função **socket()**. Em seguida ele se conecta ao servidor através da função **connect()** e inicia um loop (laço) que fica fazendo **send()** (envio) e **recv()** (recebimento) com as mensagens específicas da aplicação. É no par **send, recv** que temos a comunicação lógica. Quando alguma mensagem da aplicação diz que é o momento de terminar a conexão, o programa chama a função **close()** para finalizar o socket.



SERVIDOR

O programa servidor também utiliza a mesma API de sockets. Ou seja, inicialmente ele também cria um socket. No entanto, diferentemente do cliente, o servidor precisa fazer um **bind()**, que associa o socket a uma porta do sistema operacional, e depois utilizar o **listen()** para escutar novas conexões de clientes nessa porta.

Quando um novo cliente faz uma nova conexão, a chamada **accept()** é utilizada para começar a se comunicar. Da mesma forma que no cliente, o servidor fica em um loop (laço) recebendo e enviando mensagens através do par de funções **send()** e **recv()**. Quando a comunicação com o cliente termina, o servidor volta a aguardar novas conexões de clientes.



Os sockets do tipo TCP são orientados a conexão e tem um canal exclusivo de comunicação entre cliente e servidor. Eles garantem a ordem dos pacotes, são considerados confiáveis e sem perda. No entanto, quando se trata de se recuperar de falhas e perda de pacotes ele é mais burocrático e lento.

Utilizamos o **JAVA** como linguagem de implementação dos sockets e criamos as seguintes classes: **Servidor**, **Cliente** e **CacaPalavra**. Para permitir que o jogo fosse multiusuário utilizamos fluxos de execução que permitem um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas concorrentemente (**Threads**). Segue abaixo a estrutura básica cliente-servidor que implementa essa abstração.

28 lines (22 sloc) | 660 Bytes

Raw

Blame

History



```
1 package negocio;
2
3 import java.io.PrintWriter;
4 import java.net.Socket;
5 import java.util.Scanner;
6
7 public class Client {
8     public static void main(String[] args) {
9         try {
10             Socket client = new Socket("", 1234);
11             Scanner entrada = new Scanner(client.getInputStream());
12             Scanner entradaUsuario = new Scanner(System.in);
13             PrintWriter out = new PrintWriter(client.getOutputStream(), true);
14             System.out.println(entrada.nextLine());
15
16             while (true) {
17                 System.out.println("Digite uma palavra: ");
18                 out.println(entradaUsuario.nextLine());
19                 System.out.println(entrada.nextLine());
20             }
21
22         } catch (Exception e) {
23             e.printStackTrace();
24         }
25     }
26 }
27 }
```

24 lines (19 sloc) | 550 Bytes

Raw

Blame

History



```
1 package negocio;
2
3 import java.net.ServerSocket;
4 import java.net.Socket;
5
6 public class Server {
7     public static void main(String[] args) {
8         try {
9             ServerSocket server = new ServerSocket(1234);
10             System.out.println("Servidor em rede");
11             while (true) {
12                 Socket cliente = server.accept();
13                 System.out.println("Cliente [" + cliente.getInetAddress() + "] conectado ");
14                 Thread thread = new Thread(new CacaPalavra(cliente));
15                 thread.start();
16             }
17
18         } catch (Exception e) {
19             e.printStackTrace();
20         }
21     }
22 }
23 }
```

Referências bibliográficas

<https://www.portaleducacao.com.br/conteudo/artigos/estetica/caca-palavras/49252>

<https://pt.wikipedia.org/wiki/Ca%C3%A7a-palavras>

https://pt.wikipedia.org/wiki/Categoria:Jogos_de_palavras