



Ficções Interativas em Perl

Nelson Corrêa de Toledo Ferraz

<nferraz@gnubis.com.br>



O que é “Ficção Interativa”?

"Interactive fiction" is a catch-all name for many forms of story-telling. Most forms are text-based (but see below) and feature some degree of reader, or player, participation beyond the act of, say, turning the page of a book to read the next one.

In the context of rec.arts.int-fiction the name is most commonly used to refer to just one type: computer-based text adventures. These games involve the player entering textual commands in response to the game's output. In turn, this output is influenced by the player's input.

-- rec.arts.int-fiction FAQ



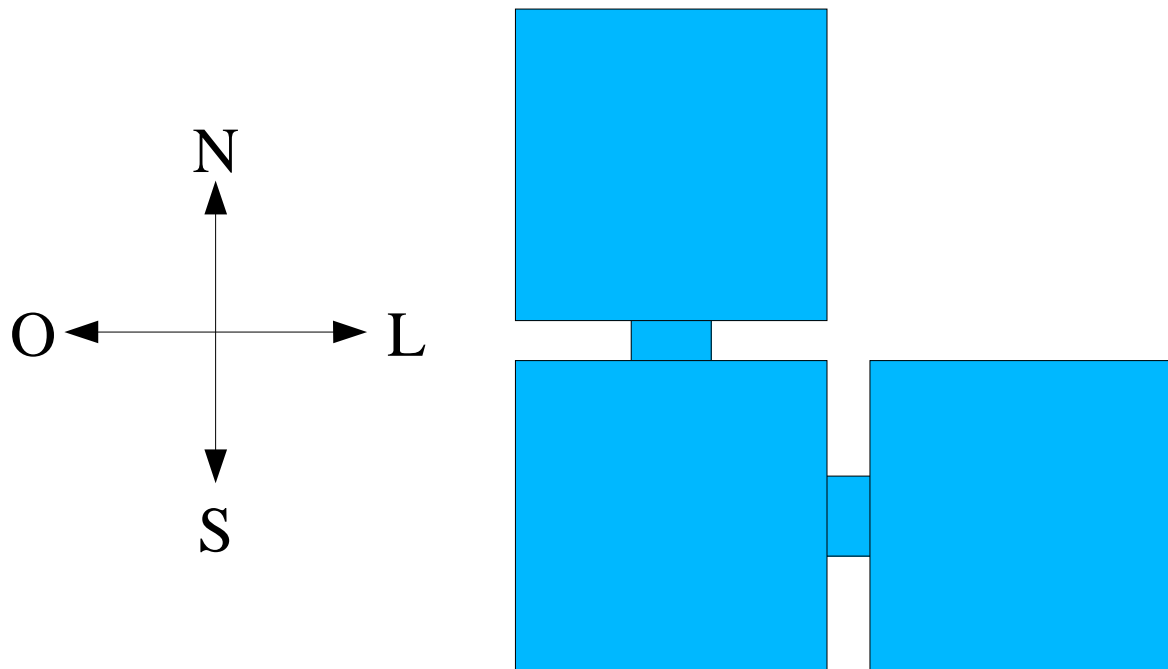
Planejando uma Ficção Interativa



- Enredo
- Mapa
- Objetos
- Desafios



Mapa





Locais

```
# Locais

my %loc = (
    'sala' => {
        descr => 'a sala',
        norte => 'quarto'
    },
    'quarto' => {
        descr => 'o quarto',
        sul => 'sala',
    },
);
```



Objetos

```
# Objetos

my %obj = (
    'lanterna' => {
        descr => 'uma lanterna',
        local => 'quarto'
    },
    'chave' => {
        descr => 'uma chave',
        local => 'quarto'
    },
    'mesa' => {
        descr => 'uma mesa',
        local => 'sala'
    },
);
```



O Jogador

```
my %jogador = (  
    'local' => 'sala',      # Posicao inicial  
);
```



Loop Principal

```
while (1) {  
    # ...  
}
```




Loop Principal

```
while (1) {  
    # Imprime a posicao atual do jogador e pede um comando  
    my $local = $jogador{local};  
    my $descr = $loc{$local}->{descr};  
    print "[[ $descr ]]\n";  
    print "> ";  
    my $frase = <>;  
    chomp $frase;  
  
    # ...  
}
```



Loop Principal

```
while (1) {
  # Imprime a posicao atual do jogador e pede um comando
  print "[[ $loc{$jogador{local}}->{descr} ]]\n";
  print "> ";
  my $frase = <>;
  chomp $frase;

  # Divide comando em verbo e objetos
  my ($vrb,@obj) = split (/\\s+/, $frase);

  # Verifica se o verbo existe
  if (defined $vrb{$vrb}) {
    eval($vrb{$vrb});
    print "$@" if defined $@;
  } else {
    print "Não sei como fazer isso!\n";
  }
}
```



Verbos e Ações

```
# Verbos

my %vrb = (
    'pegue' => '&pegar(@obj)',
    'solte' => '&soltar(@obj)',
    'examine' => '&examinar(@obj)',
    'norte' => '&ir("norte")',
    'sul' => '&ir("sul")',
    'leste' => '&ir("leste")',
    'oeste' => '&ir("oeste")',
    'fim' => '&fim',
    'quit' => '&fim',
    'exit' => '&fim',
);
```



Ação: “fim”

```
sub fim {  
    # Termina o jogo  
    exit;  
}
```



Ação: “ir <direção>”

```
sub ir {
  # Move o jogador em determinada direcao (se possivel)
  my $direcao = shift;

  if (defined $loc{$jogador{local}}->{$direcao}) {
    $jogador{local} = $loc{$jogador{local}}->{$direcao};
  } else {
    print "Não posso seguir por aí!\n";
  }
}
```



Ação: “examinar <objeto>”

```
sub examinar {
  # Examina um objeto
  my $obj = shift;

  if ($obj{$obj}->{local} eq $jogador{local} or
      $obj{$obj}->{local} eq 'jogador') {
    print "É apenas $obj{$obj}->{descr} comum.\n";
  } else {
    print "Não vejo $obj{$obj}->{descr} por aqui.\n";
  }
}
```



Ação: “pegar <objeto>”

```
sub pegar {
  # Pega um objeto
  my $obj = shift;

  if ($obj{$obj}->{local} eq $jogador{local}) {
    $obj{$obj}->{local} = 'jogador';
    print "Ok\n";
  } elsif ($obj{$obj}->{local} eq 'jogador') {
    print "Você já tem $obj{$obj}->{descr}!\n";
  } else {
    print "Não vejo $obj{$obj}->{descr} por aqui.\n";
  }
}
```



Ação: “soltar <objeto>”

```
sub soltar {
  # Solta um objeto
  my $obj = shift;

  if ($obj{$obj}->{local} eq 'jogador') {
    $obj{$obj}->{local} = $jogador{local};
    print "Ok\n";
  } else {
    print "Você não tem $obj{$obj}->{descr}.\n";
  }
}
```




Subs Auxiliares

```
sub listar_objetos {  
    # Lista os objetos em um determinado local  
    my $local = shift;  
  
    my @lista;  
    foreach (keys %obj) {  
        if ($obj{$_}->{local} eq $local) {  
            push @lista, $_;  
        }  
    }  
  
    return @lista;  
}
```



Exercício

- Criar um novo verbo, “temos”, e a sub correspondente
- Quando o jogador digitar “temos”, deverá obter uma lista de objetos que estão sendo carregados pelo jogador (local = “jogador”)



Possível Solução

```
my %vrb = (  
    ...  
    'temos' => '&temos',  
);
```

```
sub temos {  
    # Lista objetos sendo carregados pelo jogador  
    my @lista = listar_objetos_em("jogador");  
  
    print "Você está carregando: ";  
    if (@lista) {  
        print join(", ", @lista), "\n";  
    } else {  
        print "nada de interessante.\n";  
    }  
}
```



Exercício

- Modifique o loop principal, para que o jogador obtenha uma lista de objetos presentes em cada sala, automaticamente. Exemplo:

```
[[ a sala ]]  
  
Eu vejo aqui: uma mesa.  
> _
```



Possível Solução

```
while (1) {
  # Imprime a posicao atual do jogador e pede um comando
  print "[[ $loc{$jogador{local}}->{descr} ]]\n";
  my @lista = listar_objetos_em($jogador{local});
  if (@lista) {
    print "Eu vejo aqui: ", join(", ", @lista), "\n";
  }
  print "> ";
  my $frase = <>;
  chomp $frase;

  # ...
}
```



Exercício

- Um novo objeto: “porta”
- Um novo verbo: “abrir” (“abra <objeto>”)
 - Atenção: a porta só poderá ser aberta se o jogador estiver carregando a chave!



Possível Solução

```
sub abrir {
  # Abre um objeto
  my $obj = shift;

  if ($obj ne 'porta') {
    print "Não posso abrir isso!\n";
    return;
  }

  if ($obj{$obj}->{local} eq $jogador{local}) {
    if ($obj{chave}->{local} eq 'jogador') {
      print "Ok\n";
      $obj{$obj}->{aberta} = 1;
    } else {
      print "A porta está trancada!\n";
    }
  } else {
    print "Onde você está vendo $obj{$obj}->{descr}?\n";
  }
}
```