Adventures in Perl6

Nelson Ferraz <nctf@log.pt>

Lisbon Perl Mongers
Perl Tech Meeting
2007-07-10

Introduction

Adventures

- Text-based

- Interactive

- Story telling

What we'll see

- Data structures

- Flow control

- Text manipulation

- Objects and Roles

- ...in Perl6

Perl6 now

- Pugs

- v6

- mp6

- kp6

Installing v6.pm

From the command line (as root):

```
# cpan v6
```

Hello, v6!

```perl
#!/usr/bin/perl

use v6;

say "Hello, world!"
```

Reading user input

In Perl 5: the "diamond operator"

<>

Reading user input

In Perl6: the "fish operator"

```
=<>
```

Reading user input

```
while ( 1 ) {
    print "What do you want to do? ";

    my $input = =<>;

    say "I don't know how to do that!"
}
```

Parsing commands

In Perl 5: regexps

```
# clean up articles, prepositions, etc.
$input =~ s/\b(a|an|the|in|on|at)\b/ /g;
$input =~ s/\s+/ /;

# extract verb and object from input
my ( $verb, $object ) = split( $input );
```

Parsing commands

In Per6: grammars!

```
grammar Adventure {
 token command  {
      | <verb> <ws> <object>
      | <verb>
 }

 token verb    { look|take|drop|use }
 token object  { sign|coin }
}
```

Parsing commands

It's easy to extend a grammar...

```
grammar Adventure {
 token command  {
      | <verb> <ws> <article> <ws> <object>
      | <verb> <ws> <object>
      | <verb>
 }

 token verb     { look|take|drop|use }
 token object  { sign|coin }
 token article { a|an|the|at|in|on|to }
}
```

Parsing commands

Really easy...

```
grammar Adventure {
 token command  {
     | <verb> <ws> <article> <ws> <object>
     | <verb> <ws> <object>
     | <verb>
     | <direction>
 }


 token verb    { look|take|drop|use }
 token object  { sign|coin }
 token article { a|an|the|at|in|on|to }
 token direction { north | south | east | west }
}
```

Planning the Adventure

Bill of Player's Rights

- ❖ Not to be killed without warning

- ❖ Not to be given horribly unclear hints

- ❖ To be able to win without experience of past lives

- ❖ To be able to win without knowledge of future events

- ❖ Not to have the game closed off without warning

Bill of Player's Rights (2)

- Not to need to do unlikely things

- Not to need to do boring things for the sake of it

- Not to have to type exactly the right verb

- To be allowed reasonable synonyms

- To have a decent parser

- To have reasonable freedom of action

Bill of Player's Rights (3)

- Not to depend much on luck

- To be able to understand a problem once it is solved

- Not to be given too many red herrings

- To have a good reason why something is impossible

- To know how the game is getting on
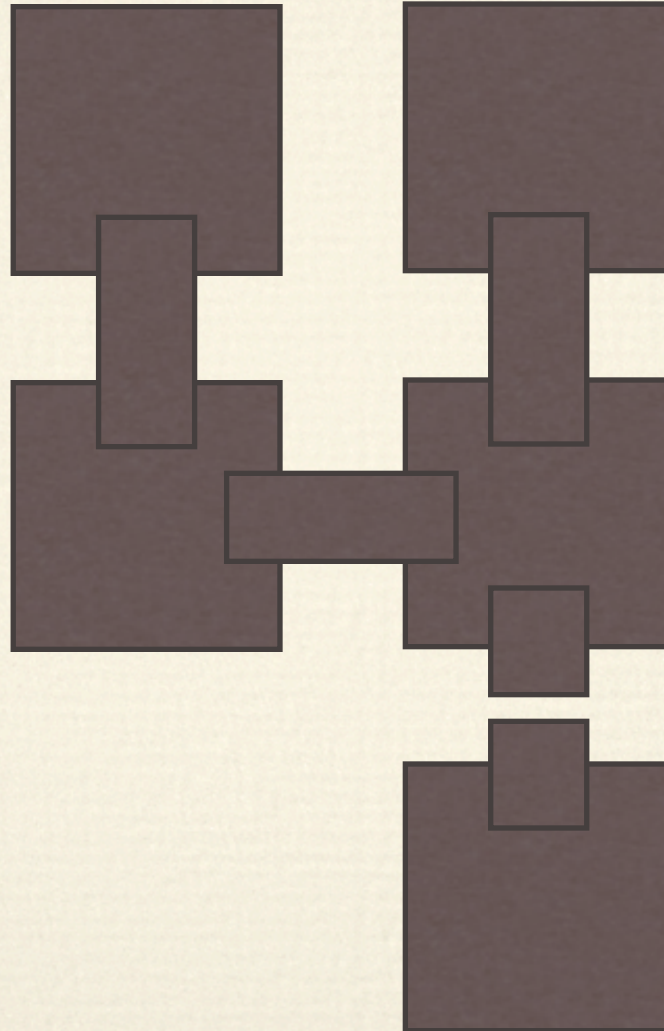
Planning

- Plot

- Scenes

- Puzzles

Useful techniques

- Synopsis

- Scripts

- Maps

The Map

Writing the Adventure

The map

```perl
my %map = (
  chamber => {
    north => 'throne room',
    east  => 'dungeon',
  },
  'throne room' => { south => 'chamber', },
  dungeon => {
    west  => 'chamber',
    north => 'crypt',
    south => '', # blocked by the door
  },
  crypt => { south => 'dungeon', },
  cell => { north => 'dungeon', },
);
```

The objects

```perl
my %object = (
    sign => {
        place => "chamber",
        description => "...",
    },
    coin => { place => "cell", },
    key => { place => "crypt", },
    vampire => { place => 'crypt', },
    cross => { place => 'throne room', }
    door => {
        place => "dungeon",
        description => "The door is closed",
    },
);
```

The player

```
my %player = (
    place => "chamber",
    score => 0,
);
```

The game loop

```
while (1) {
    print "> ";
    my $input = =<>;

    my $response = Adventure.command( $input )<command>;

    if $response{'direction'} {
        # ...
    }
    elsif $response{'verb'} {
        # ...
    }
}
```

Playing the Adventure

Resources

* Nelson, Graham. Craft of Adventure.

* Taylor, Mike. Games::ScottAdams::Tutorial.

Questions?

Nelson Ferraz <nctf@log.pt>