Package: r.oguelike (via r-universe)

November 2, 2024

Title Play a Roguelike 'Game' in the Console

Version 0.1.0

Description Play with a simple ASCII-only tile-based toy in the console, inspired heavily by roguelike games like Rogue (1980). Proof of concept.

License MIT + file LICENSE

URL https://github.com/matt-dray/r.oguelike, https://matt-dray.github.io/r.oguelike/, https://www.rostrum.blog/tags/r.oguelike/

BugReports https://github.com/matt-dray/r.oguelike/issues

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Imports crayon, keypress, sonify

Suggests covr, testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://trevorld.r-universe.dev

RemoteUrl https://github.com/matt-dray/r.oguelike

RemoteRef HEAD

RemoteSha 7d964f0975fc493090f6e0f11ef7f128b6e43a8b

Contents

.move_enemy *Move The Enemy To The Player*

Description

Move The Enemy To The Player

Usage

```
.move_enemy(room, dist, enemy_loc)
```

Arguments

room	Matrix. 2D room layout.
dist	Matrix. Tile distance to player.
enemy_loc	Numeric. Matrix index of the tile occupied by the enemy character.

generate_dungeon *Generate A Dungeon Map*

Description

Procedurally generate a tile-based ASCII-character dungeon map. Creates a tile-based map of user-specified size; places randomly a user-specified number of rooms; connects them with a continuous corridor; and iteratively expands the interior space by sampling from adjacent tiles. Note: much of this function has been absorbed into the start_game function, but generate_dungeon will continue to be exported for now.

Usage

```
generate_dungeon(
  iterations = 5,
  n_row = 30,
  n_col = 50,
  n_rooms = 5,
  is_snake = FALSE,
  is_organic = TRUE,
  seed = NULL,
  colour = TRUE
)
```

Arguments

iterations	Numeric. How many times to 'grow' iteratively the dungeon rooms, where tiles adjacent to current floor tiles (.) have a random chance of becoming floor tiles themselves with each iteration.
n_row	Numeric. Number of row tiles in the dungeon, i.e. its height.
n_col	Numeric. Number of column tiles in the dungeon, i.e. its width.
n_rooms	Numeric. Number of rooms to place randomly on the map as starting points for iterative growth.
is_snake	Logical. Should the room start points be connected randomly (FALSE, the default) or from left to right across the room matrix (TRUE)? See details.
is_organic	Logical. Join the room start points with corridors before iterative growth steps (TRUE, the default), or after (FALSE)? See details.
seed	Numeric. Seed to reproduce a dungeon.
colour	Logical. Should the characters be coloured using crayon (TRUE, the default)?

Details

- You'll have to experiment to find the 'best' argument values for your needs. Typically, a larger dungeon should have a higher n_rooms value and can be grown through more iterations. Use is_snake and is_organic to play with dungeon appearance and connectedness.
- For argument is_snake, TRUE will create a single continuous, winding cavern from left to right, while FALSE will create a more maze-like cavern.
- For argument is_organic, TRUE will generally create what looks like a natural cavern, since the room start points and corridors are subject to iterative growth. When FALSE and is_snake is FALSE, the dungeon's caverns are more square, or more 'artificial' looking. When FALSE and is_snake is TRUE, the result is more likely to be a series of discrete roundish rooms connected by a narrow (one tile-width) corridor.

Value

A matrix, invisibly. Output via cat to the console.

Examples

```
## Not run:

# A 'natural' cavern with default arguments
generate_dungeon(seed = 23456)

# Rooms connected sequentially by narrow corridors
generate_dungeon(
  iterations = 8,      # iterate room growth 8 times
  n_rooms = 4,         # randomly place 4 room tiles to start
  is_snake = TRUE,     # connect rooms from left- to right-most
  is_organic = FALSE,  # add single tile-width corridors after growth
  seed = 2
)
```

```
## End(Not run)
```

`start_game` *Play A Roguelike Game*

Description

Clears the console and starts a game of 'r.oguelike' by printing a map with an inventory and status message. The user inputs a keypress to move the character and explore the map, fighting enemies and collecting objects. This is a toy; a proof-of-concept.

Usage

```
start_game(
  max_turns = Inf,
  n_row = 20L,
  n_col = 30L,
  n_rooms = 5L,
  iterations = 4L,
  is_snake = FALSE,
  is_organic = TRUE,
  has_colour = TRUE,
  has_sfx = TRUE
)
```

Arguments

`max_turns`	Integer. How many turns? Default is `Inf`(inite).
`n_row`	Integer Number of row tiles in the dungeon, i.e. its height.
`n_col`	Integer. Number of column tiles in the dungeon, i.e. its width.
`n_rooms`	Integer Number of rooms to place randomly on the map as starting points for iterative growth.
`iterations`	Integer. How many times to 'grow' iteratively the dungeon rooms, where tiles adjacent to current floor tiles (`.`) have a random chance of becoming floor tiles themselves with each iteration.
`is_snake`	Logical. Should the room start points be connected randomly (`FALSE`, the default) or from left to right across the room matrix (`TRUE`)? See details.
`is_organic`	Logical. Join the room start points with corridors before iterative growth steps (`TRUE`, the default), or after (`FALSE`)? See details.
`has_colour`	Logical. Should the characters in the output be coloured using `crayon` (`TRUE`, the default)?
`has_sfx`	Logical. Should sound effects be used? Defaults to `TRUE`.

Details

Use the WASD keys to move up, left, down and right. Use the '1' key to eat an apple from your inventory. Use the '0' to quit the game. If your terminal supports the 'keypress' package, then you can use a single keypress as input (e.g. the up arrow key), otherwise you will have to type at the prompt and then press 'Enter'.Use `has_keypress_support` to see if 'keypress' is supported.

Symbols used in the game are as follows:

- `.` floor tile
- # wall
- `@` player (10 HP max, -1 HP attack damage)
- $ gold (+1 to +3 G)
- `E` enemy (3 HP max, -1 HP attack damage)
- `a` apple (+1 HP)

When `TRUE`, `is_snake` will tend to create one continuous cavern; `is_organic` will tend to create more 'natural' looking caves.

Arguments that take integer values are coerced to integer if provided as numeric values.

Value

Nothing. Clears the console and prints to it with `cat`.

Examples

```
## Not run: start_game()
```

Index