libchar Quick Reference

github.com/LieutenantPorky/libchar

Setup

`libchar.setup(width, height)` – create a screen with given width and height, and setup the environment

`libchar.set_background(path)` – set the image at the given path as the background

Agents

`my_agent = libchar.Agent(path)` – create a new agent using the image at the given path

`my_character = libchar.Character(path)` – create a new character using the image at the given path. Characters work just like agents, but won't get garbage-collected

`my_agent.move(position)` – move an agent to a new position. Position is calculated based on an agent's center

`my_agent.velocity` – an agent's velocity: has an x and y component, and can be read and set

`my_agent.set_rotation(angle)` – set the rotation of an agent

`my_agent.rotate(angle)` – apply an additional rotation to an agent

`my_agent.get_height()` – get the height of an agent

`my_agent.get_width()` – get the width of an agent

`my_agent.kill()` – delete an agent

Collisions

`my_agent.add_tag(tagname)` – add a tag to an agent

`my_agent.remove_tag(tagname)` – remove a tag from an agent

`my_agent.is_colliding(other_agent)` – returns true if *other_agent* is colliding with the agent

`my_agent.is_colliding_tag(tagname)` – returns the first agent with tag *tagname* that is colliding with the agent. If there are no colliding objects, returns None

Constants

`libchar.dt` – returns the length of time taken up by a frame

`libchar.frame` – returns the number of frames since launch

`libchar.living` – returns a list of all living agents and characters

`libchar.screen` – returns the pygame screen object of the application

libchar.gc_int – interval in frames between garbage collection passes. GC involves deleting all agents (not characters) that are far from the screen area. Setting *gc_int* to 0 disables garbage collection.