Week four

Help

Lists — Lists

- Lists can be constructed as a sequence of objects inside square brackets; e.g; `[2,4,6]`. The `list` function also converts other types of sequence data such as strings into lists.
- Sequence operations such as indexing `l[i]`, concatenation `+`, length `len(l)` and slicing apply to lists.
- As opposed to strings, an element of a list can be *mutated* via assignment `l[i] = x`.
- Lecture examples - None
- More examples - Structure, Tuples, True-False Quiz, Silly Words, Rainbow Canvas, Digital Numbers

Points and vectors — Motion

- A point in 2D is represented by a pair of Cartesian coordinates.
- A vector in 2D is represented by a pair of numbers (its horizontal and vertical components).
- Taking the difference of two points (componentwise) yields a vector.
- Vectors can be added and scaled (componentwise). Points should not be added or scaled.
- Adding a point and a vector (componentwise) yields a new point. This operation can be used to animate moving points.
- Lecture examples - Timer Control, Formula Control
- More examples - Drawing Vectors

Distance computations — Collisions and reflections

- The distance between two points `p0` and `p1` is $\sqrt{(p0[0]-p1[0])^2+(p0[1]-p1[1])^2}$.
- The distance between a point and a circle and the distance between two circles follows from this formula.
- The set of points `[x,y]` that satisfy the equation `a*x + b*y + c == 0` is a line. The distance from this line to a point `p` is $(a*p[0] + b*p[1] + c)/\sqrt{a^2 + b^2}$
- The distance from a circle to a line is the distance from the center of the circle to the line minus the radius.
- Lecture examples - Timer Control
- More examples - Drawing Vectors

Reflections — Collisions and reflections

- The direction of reflection for a ball bouncing off of a wall depends on the incoming velocity vector `v` and the normal vector to the wall at the point of contact.
- The incoming velocity vector can be decomposed into $v = v_p + v_n$ where v_n is the component of v orthogonal to wall and v_p is the component parallel to to the wall.
- In this model, the reflected vector is $v = v_p - v_n$.
- This model simplifies for horizontal and vertical walls. In particular, reflection simply negates one component of the velocity vector `v`.
- Lecture examples - Timer Control
- More examples - None

Keyboard events — Keyboard input

- SimpleGUI suppports two event handlers for keyboard events.
- The key down event handler is registered via `set_keydown_handler`.

- The key up event handler is registered via `set_keyup_handler`.
- The variable passed to each of these handlers is a number that can be compared at the constants in `KEY_MAP` to determine which key has been pressed.
- Lecture examples - Echo
- More examples - Shape Selection

Positional control — Keyboard input

- We can control the position of a point `p` directly using key board events.
- The horizontal and vertical components of a point `p`'s position can be increment/decrement via `p[i] += c` in response to key presses.
- Lecture examples - Ball Position
- More examples - None

Velocity control — Velocity control

- Basic physics relates the position of a point `p` and its velocity `v` via the equation `p += dt * v` where `dt` is a small time step.
- In this model, we can control the motion of `p` via keyup/keydown events that increment/decrement the two components of the velocity vector `v` via `v[i] += c`.
- Lecture examples - Velocity Control
- More examples - Ball Track

Mutable vs. immutable data — Programming tips #4

- Numbers, Booleans and strings are *immutable* and can not be modified. Only new copies of these kinds of data can be made.
- On the other hand, parts of a list can be *mutated* via assignment to individual elements of the list.
- Assignment of an entire list to a variable generates a *reference* that refers to the list. Subsequent assignment may generate multiple references to the same list.
- Mutating a list with multiple references modifies all references to the list. This capability is very useful, but can generate subtle errors.
- Lecture examples - Global, Tuples
- More examples - List Structure

Created Wed 5 Sep 2012 10:44 PM CEST

Last Modified Sun 3 Nov 2013 4:41 PM CET