Module 3a: How to Design Worlds

CPSC 110

Peyton Seigo

2018-09-19

Module 3a: How to Design Worlds

- Be able to explain the inherent structure of interactive graphical programs.
- Be able to use the How to Design Worlds (HtDW) recipe to design interactive programs with atomic world state.
- Be able to read and write big-bang expressions.

How to Design Worlds (HtDW) Recipe

World program design is divided into two phases, each of which has sub-parts:

1. Domain analysis (use a piece of paper!)
    1. Sketch program scenarios
    2. Identify constant information
    3. Identify changing information
    4. Identify big-bang options
2. Build the actual program
    1. Constants (based on 1.2 above)
    2. Data definitions using HtDD (based on 1.3 above)
    3. Functions using HtDF
        1. main first (based on 1.3, 1.4 and 2.2 above)
        2. wish list entries for big-bang handlers
    4. Work through wish list until done

Working through the recipe

- `empty-scene` is a primitive that allows you to create a background
- HtDF
    - Use the world constants in `check-expect`s
        * Clarity + correctness if constants change
    - Using named constants provides a "single point of control"
    - **Large enumeration rule** (i.e. `KeyEvent`)
        * only include `cond` cases the function cares about
        * other cases are handled by an **`else`**
- HtDD
    - It is important to state units in the interpretation
- HtDW
    - Work on this process until the flow from one recipe to the next is SECOND NATURE!
    - **Wish list entry**: a big-bang handler's signature, purpose, `!!!`, and stub

  - The domain analysis is a **model** of the program
    * Improve program by marking up analysis

Notes

Interactive behaviour is generally defined as:

- changing state
- changing display
- keyboard and/or mouse affects behaviour

Terminology

- `big-bang` is **polymorphic**: it can work for any type of world state
  - an interface that works for many different types of data