Hermes

An open-source “game” library for Processing

→ Supports both traditional and non-traditional conceptions of game.

→ built for the Processing community

→ the process of creating a game is *defining a set of rules.*

General Design Goals

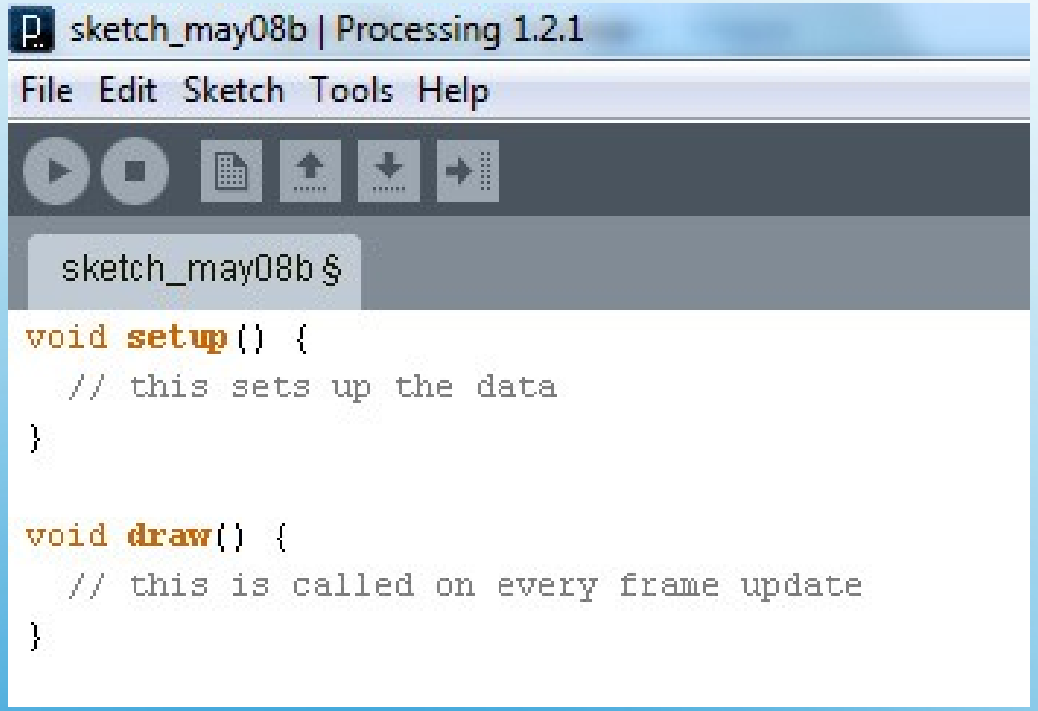
Ease of use – can be used by even a relatively inexperienced programmer.

Robustness – will still be relevant and sufficiently powerful for a more advanced programmer.

Modularity – utilities such as animation, the post office, and physics can be used independently of the core framework.

Self-contained – doesn't require knowledge of any other library except Processing and Java.

This is a Processing sketch:



→ we've added a second loop: the World update loop

The Core

World – automates the game update loop. Runs on its own thread, so drawing and game logic are parallel.

Being – a spatially located object, such as a character or platform. Has abstract draw() and update() methods.

Interactor – interface for the “rules” by which beings interact. Has detect() and handle() methods.

Camera – provides a view into the World. Draws registered Beings. Handles coordinate transforms.

The Post Office

→ Handles I/O with the outside world.

→ Objects can subscribe to messages, they can come from the mouse or the keyboard, or from some external program via OSC.

→ The post office queues messages as they come in and dispatches them on command, preventing threading issues.

Physics

Shape system – implementations for axis-aligned bounding box, circle, and arbitrary convex polygon. Collision detection for these shapes, using double dispatch.

MassedBeing – an extension of Being with support for force and impulse accelerations, and multisampling.

Collisions – robust impulse and projection-based collision response. Variable elasticity.

→ why write our own?

Animations

- works hand-in-hand with Processing's PImage functionality
- basically a beat slicer/arpeggiator for images
- variable playback speed, can play forwards or backwards
- can slice tile maps, or load images from frame sequences

Demos!!!