

LOVE2D Image Editor

CPS 499 (SP 17)
Luc Talatinian

The LOVE2D Engine

- Written in C++
- Game development through Lua scripting
 - Callbacks: `love.load()`, `love.update()`, `love.draw()`...
 - I/O packages
 - Graphics support
 - Filesystem support, etc.

The “Main-Game Loop”

- General approach for single-threaded game design
- Three major steps for every frame of game
 - Accept input
 - Update game state
 - Draw new game state for user
- These steps are accessed through LOVE2D callbacks

Basics of Image Editing

- Program loads encoded file (.png, .jpg, etc.)
- File is decoded into pixel matrix (RGB, grayscale...)
- Image is displayed to user
- User selects editing tools (pen, cut/paste, crop...)
- User changes image
 - User applies tools to image
 - Image is updated to reflect changes in real-time

Image Editing as a Main-Game Loop

- Can extend main-game loop to any interactive program
- Image editing as “game loop”
 - Get user input: tool selection, image ops
 - Update state: change pixel values
 - Draw state: refresh image with changed pixels

Image Editing in LOVE2D

- On start (`love.load()`): load image file
- Begin interactive loop
 - Get input
 - `love.keypressed()`: change editing tool or properties
 - `love.mousepressed()`: set start of region for some tools
 - Update state
 - `love.update()`: apply pixel changes for drawing tools
 - `love.mousereleased()`: apply regional changes
 - Draw
 - `love.draw()`: refresh pixel data and draw to screen
- On close (`love.quit()`): re-encode pixels into file

Storing Pixel Data

- LOVE2D Image object (remember that object=table)
 - Represents file loaded from memory
 - Drawable supertype: can be drawn to framebuffer in LOVE2D
 - Contains ImageData object that holds pixel matrix
- ImageData
 - Stores array (matrix) of RGBA pixels
 - Edit pixels through member functions
 - `setPixel()`
 - `mapPixel()`

Canvas Layer

- Can't draw directly to ImageData
- Need intermediate framebuffer for certain ops
- Canvas holds shapes drawn by some tools
- On draw application:
 - Shape is drawn to transparent canvas
 - Canvas ImageData is alpha-blended over existing ImageData
 - Canvas is cleared for next draw operation

Filesystem

- Encoding/Decoding
 - Image is loaded from main game directory
 - Saved to save data directory (usually in os user appdata)
- Resultant image encoded as .png (uncompressed)

Questions
