Pungas
GLSL Mutator

Introduction

Pungas GLSL Mutator is a tool that PVM uses for experimentation with GLSL shaders. Some of them are used in our demos, some others are a part of other projects or future intros.

It is intended for:

- Programmers who wants to create shaders and check mutations interactively. In this case, first we introduce glsl shader code and break it into functions for later mutation.

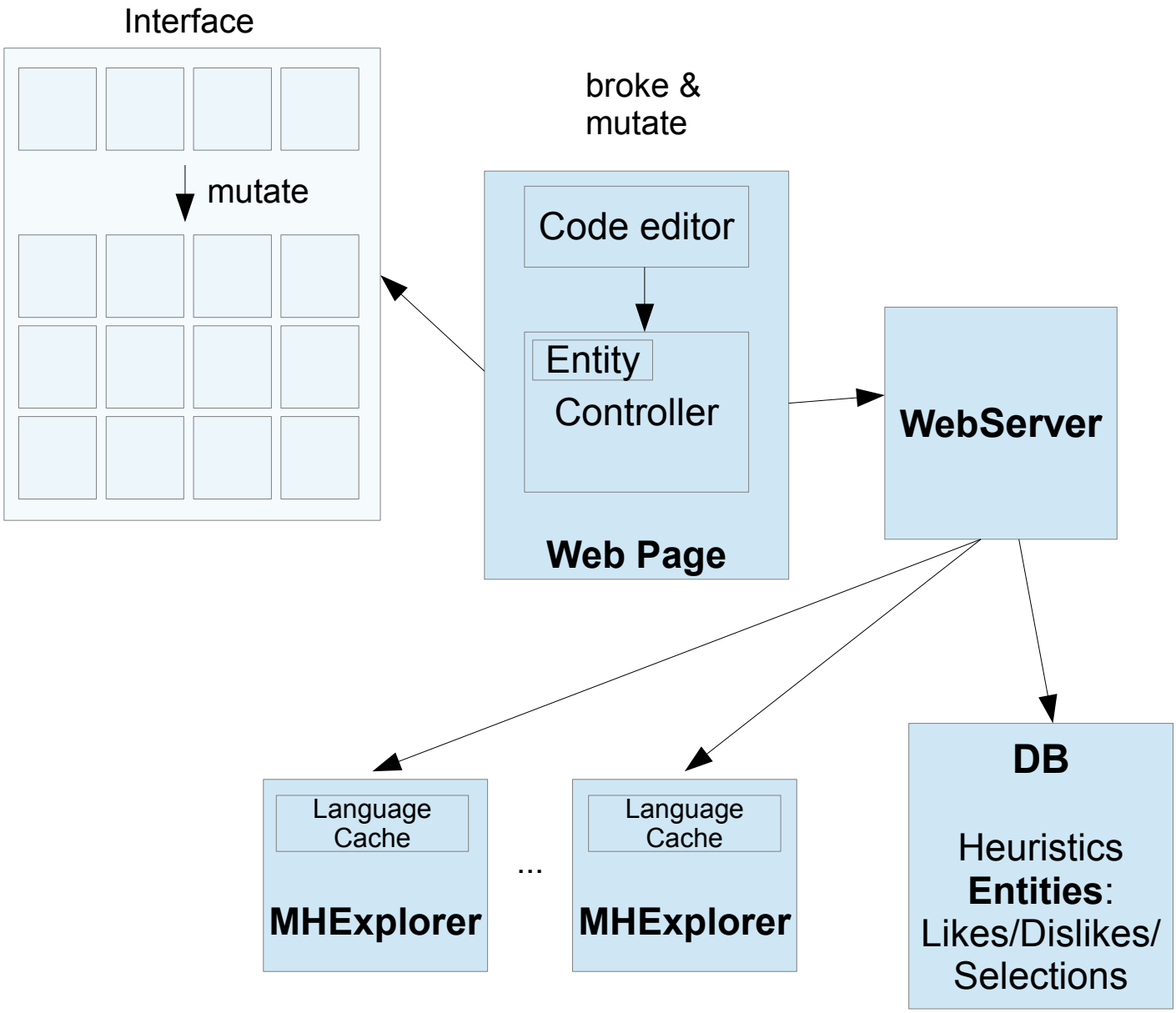
- Non programmers who want to help and create new objects by just clicking.

Online implementation with low resources can be found (when online) at http://evolvingshaders.ddns.net:9999/editor-museum.html.

It is strongly recommended the use of a touch screen hardware helper.

General Scheme

Following graphic is a general scheme of our drawing system. It is composed of some interfaces, a web server, a simple non relational database and some executables to parse and mutate glsl code.



The recommended system settings is Phenom II 6 with cores at 3.2 ghz, 8GB RAM and GLSL shaders 3.0 compliant GPU (we tested it with NVIDIA GTX 970 and ATI R9 280X) or better.

Module: MHExplorer

Module for expression processing and mutation.

Main responsibilities

- Mutate expressions using buildt in operators and basic search tasks.
- In future releases, the use of evolutionary strategies / heuristics objects.

Auxiliar responsibilities

- Parse formulas on a DSL of GLSL language buildt on server.

Implementation

- Written in common lisp, actually compiled using LispWorks 5.1.
- Due to single core nature of LispWorks 5.1 compiler, we have to run several instances (unfortunatelly it is a heavy process, 30mb executable and 200 mb in memory aproximatelly).

It is convenient to execute several instances of this module, in order to improve system responsiveness.

More info at: https://github.com/sebafonte/MetaHeuristicExplorer

Module: Web Server

Main responsibilities

- Parse GLSL code and build a DSL expression.
- Select nodes to replace with functions depending on heuristics.
- Deparse final tree with replacements.
- Manage database (likes, dislikes).
- Serve page files.

Auxiliar responsibilities

- Transform lisp-like code into js code.
- Group common operations with MHExplorer to reduce http calls.

Implementation

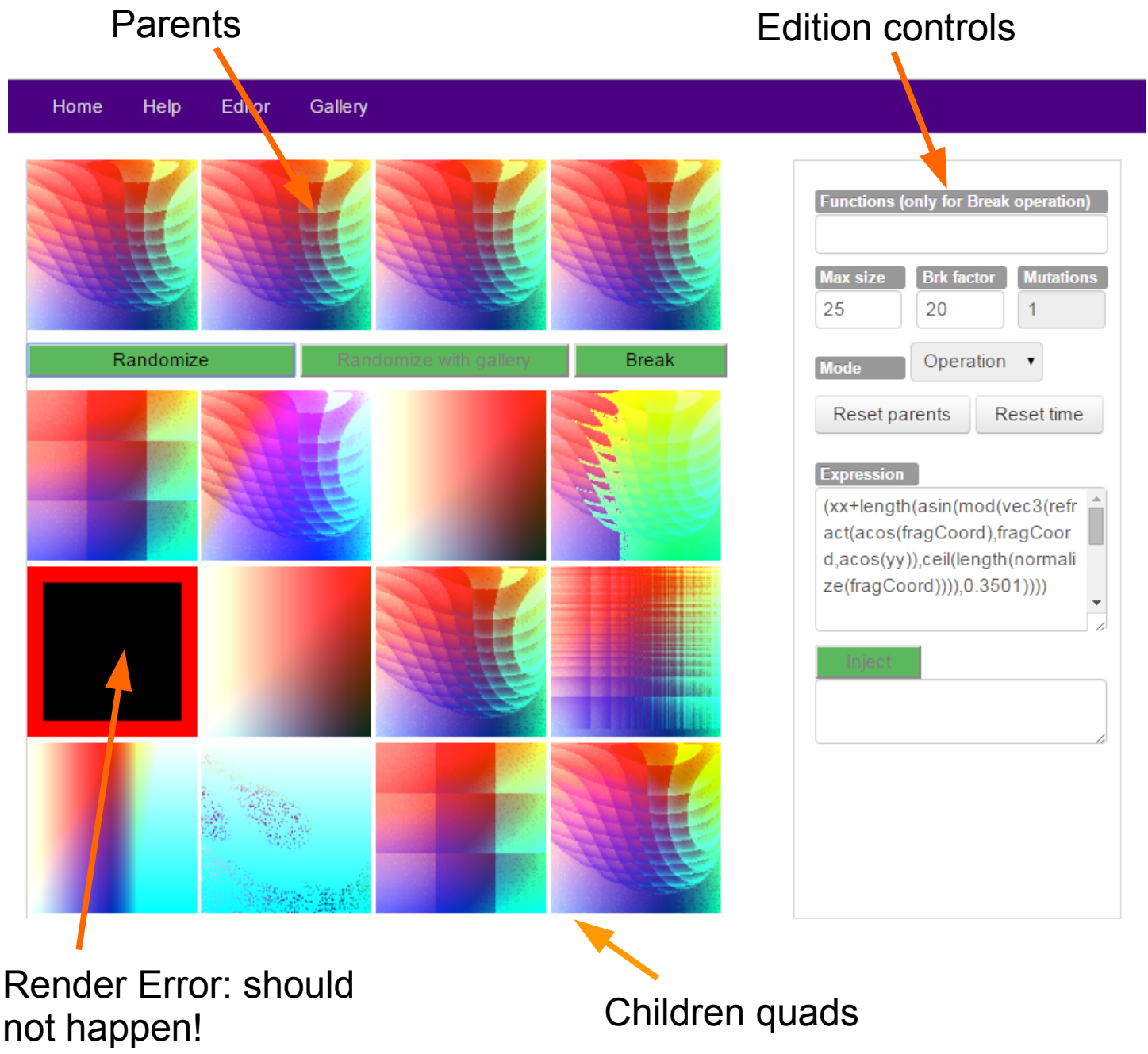
- Node.js with swig as view engine.
- glsl-man for parsing and deparsing.

More info at:
https://github.com/pungasdevillamartelli/evolvingshaders.ddns.net

Editor Interface

This interface is intended to combine properties from similar objects.

- Parents or base objects are at the top
- Clicking a parent makes random in the “children” quads
- Clicking “randomize” will do a mix of mutation and crossover between parents to fill the children quads.
- Clicking a child will mutate it into the other children quads.



Editor Interface

Main behavior

- Click parent to mutate directly.
- Drag and drop between individuals.
- Contextual menu with like, edit, view.
  - Edit:
    - Shadertoy similar editor but without multiple layers.
  - Like / Dislike:
    - Store your preferenced genes.
    - Contribute to our heuristic building process.

Buttons

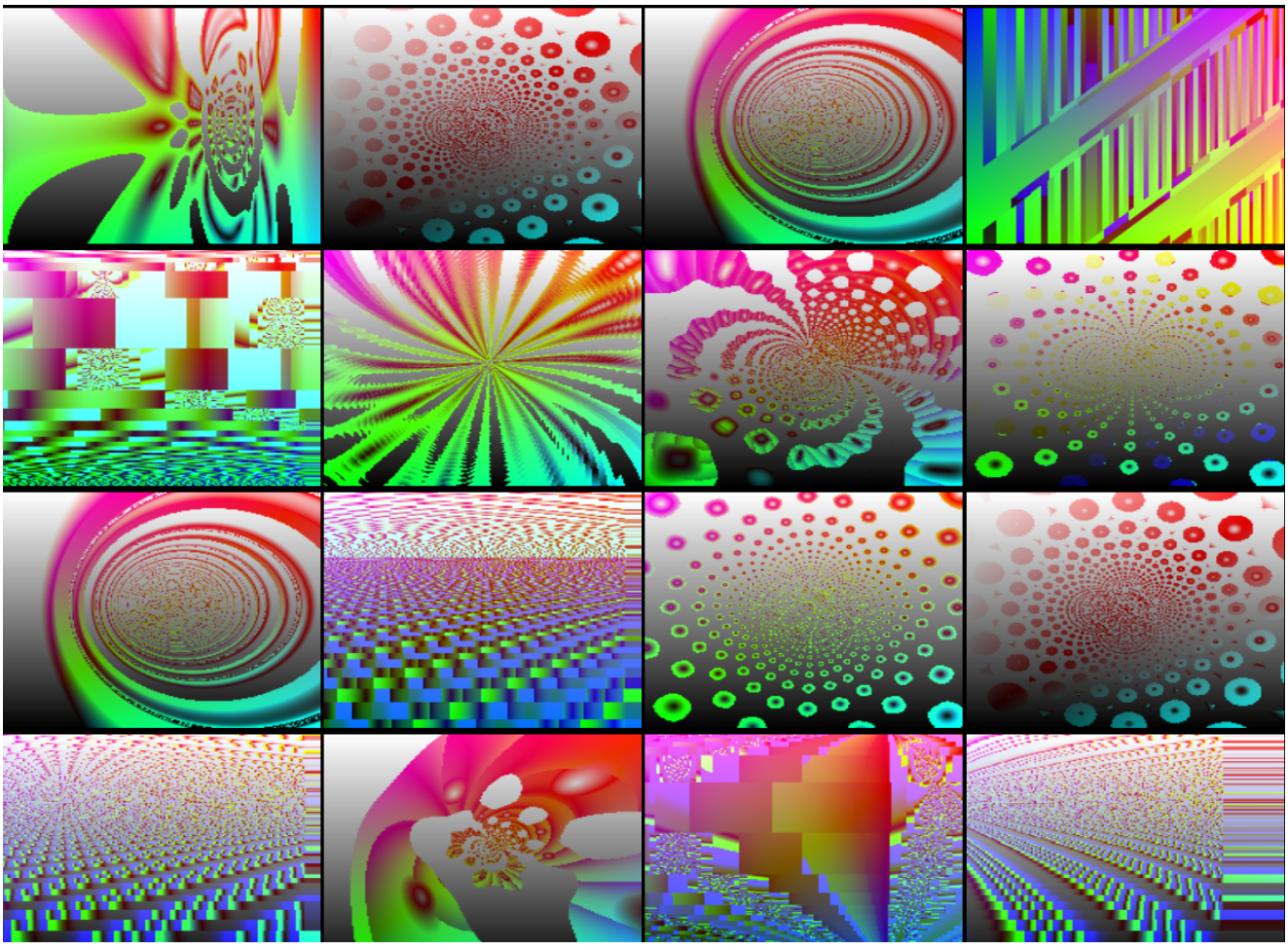
- Break:
  - Break parents into pieces.
- Randomize:
  - 50% probability of mutation.
  - 50% probability of directed crossover.
- Randomize with gallery:
  - 50% probability of mutation.
  - 50% probability of directed crossover between a parent and an object into library.
- Reset parents:
  - Bring parents to original state.
- Reset time:
  - Bring time to 0 (float time; shader variable).
- Max size
  - Max tree size for each formula component.

Editor Touch Interface

Simple interface for mutating objects just clicking.

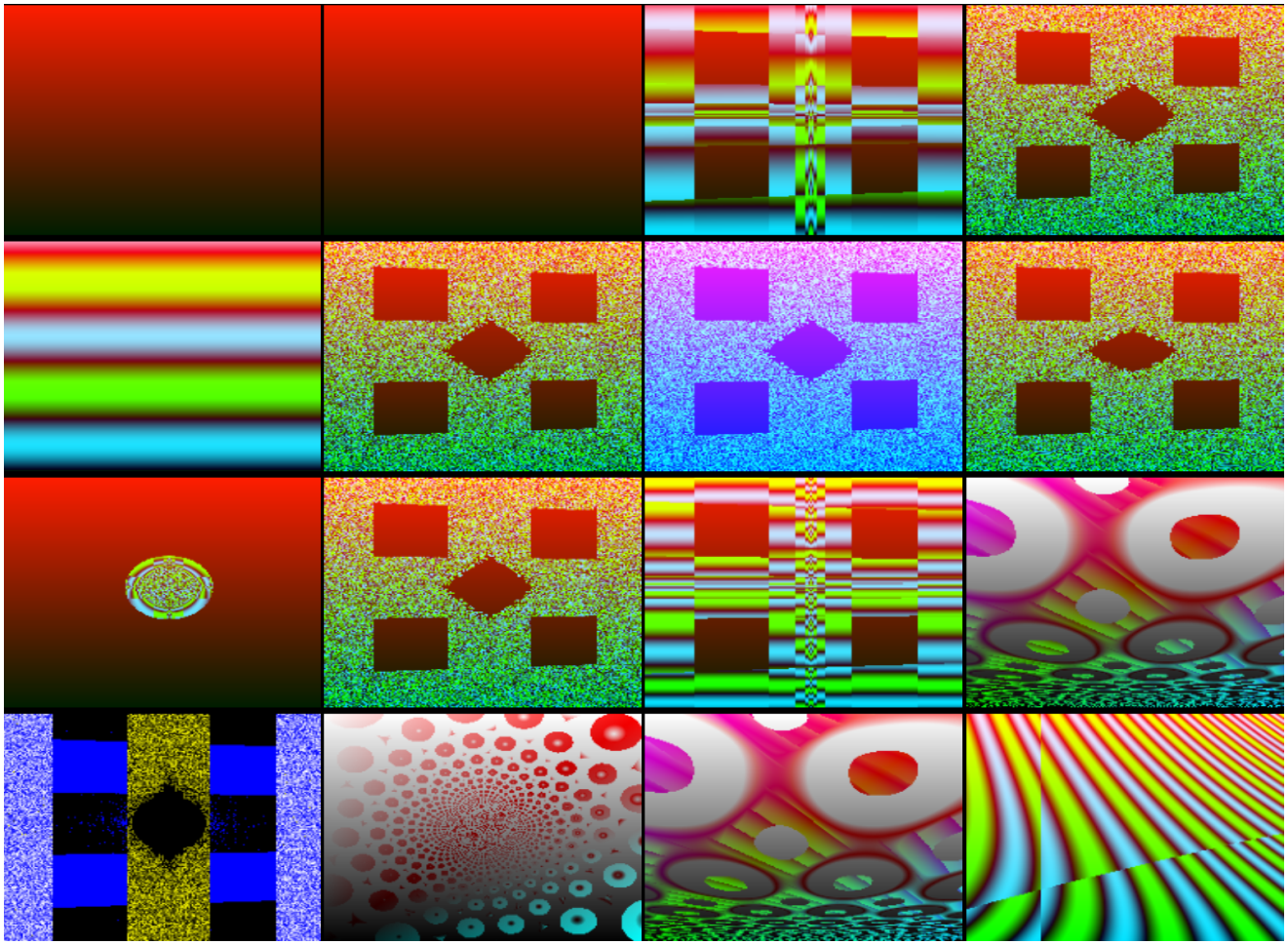
- Clicking an image makes the other be filled with a new entity which is usually a mutation of the one you clicked.
- Bottom right quads have more probability to be from the library, as a method to get off when getting stuck in a ugly generation.
- Designed to be simple to be used with a touch screen.

Example of startup screen (or refresh):



Editor Touch Interface

Example of a more specific instance of the interface:



Main behavior

- Click parent to mutate shaders into the quads directly.
- No need to drag and drop.
- No contextual menu (just prepared for single touch).

Configuration

- Max size: Max tree size for formulas is setted to the fixed value of 40.
- Mutation level: **Coming soon**.
- Libray percentaje: **Coming soon**.

Credits

Programming

- Carlos Sebastian Fonte

Support and configuration

- Leonardo Salvador Rocco

Acknowledgments

Thanks to all the PVM team for testing and supporting the installation and to Karl Sims for the initial idea.

Carlos Sebastián Fonte,
sebafonte@gmail.com
1/6/2017