Evolutionary Fractal Art

New Mexico

Supercomputing Challenge

Team 99

Sat. Science Math Academy

Team Members

Shavonne Betts

Rachel Washington

Teacher

Debra Johns

Project Mentor

Wayne Witzel

Table of Contents

Executive Summary

Our project uses human preference and gamification to evolve fractal art. The problem that our project is trying to solve is how human preference and gamification can evolve fractal art through sexual reproduction. In our project users log into an interactive website that allows users to evolve fractals while playing the fractal breeding game. For users to breed fractals as effectively as possible we plan to use one method of reproduction to evolve fractals. The method we plan to use is sexual reproduction with user specified pairs. Through the use of gamification and human preference we plan to see continued evolution in the fractals.

Introduction

Fractals are beautiful mathematical patterns. In the project Evolutionary fractal art, we are utilizing fractals, gamification, and human preference to evolve fractals. Utilizing fractal art and gamification in our project keeps user interest and makes it possible to collect data from multiple users. Our project is modeled off of similar research such as Darwin Tunes and Electric Sheep. Darwin Tunes and Electric sheep are evolutionary art research projects. Electric Sheep breeds together fractals with sexual reproduction and asexual reproduction. Darwin Tunes breeds together music clips using sexual reproduction and user preference. Utilizing sexual reproduction with user specified pairs we plan to see continued evolution.

Description

Using human preference and gamification we plan to see evolution in fractals. Users have an interactive website which they log into. Once they log into the interactive website they automatically take out a loan of 300 game currency (or gold and two random fractals. With this loan they can create new random fractals for 100 gold a piece, send a fractal to the meat grinder (trash) for 100 gold, and breed two fractals together for 100 gold. The user can also breed the two random fractals together. Also with game currency users can enter contests with other users. Users enter their best and most visually appealing fractals into the contest to be judged by other users. The user with the fractal with the most votes wins the contest and is awarded with game gold and is able to increase their flock. Utilizing the beauty contest and the game gold we are able to see the evolution in different fractal flocks and the fractals themselves.

Gamification

Gamification is the use of programing a game or using game techniques to solve a task or obtain useful information. Our project utilizes gamification to obtain information from users. With the use of gamification we plan to collect more data from more users. Gamification also encourages users to give more useful information to the project.

The objective of the Fractal Breeders game is to create a large flock of fractals by breeding appealing fractals and winning contests. For users to gain prestige users must earn game currency or “gold” to increase their fractal flock, win contests, and continuously breed more gorgeous fractals.

Similar Research

An art research project that we modeled our project after is Electric Sheep. Electric Sheep is an art research project where fractals evolve based on human preference of multiple users. Electric Sheep works as a screensaver on users' desktops. Fractals breed in two different ways mutation or asexually and sexual reproduction. Our project unlike Electric Sheep has the users fractal flock evolve as a single user and breed fractals with sexual reproduction with user specified pairs.

Darwin Tunes another art research project we are modeling our project after. Darwin Tunes evolves music sounds based on human preference. Users rate the sounds 1 through 5. The sounds users prefer are the sounds that are bred together utilizing sexual reproduction. The music than evolves continually becoming more pleasing. Darwin Tunes does reach a plateau where there is no improvement in evolution. In our project we plan to break this plateau by utilizing sexual reproduction with user specified pairs allowing users to pick the parents.

Tools

To build the basic functionality of our website we are using the web tool Web2py. Web2py uses python to build a website and HTML. Web2py also uses the flam-3 app to breed fractals together and create new fractals.

In our project we also utilize the flam-3 tool. The flam-3 tool breeds fractals together and randomly generates new fractals as well. Web2py uses the flam-3 tool to breed and generate the fractals and to add them to the users flock.

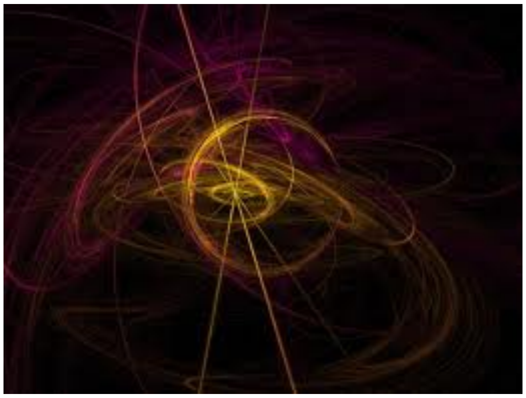
We also utilize the tool Rapid Weaver to mock up the web design, and the appearance of the web pages.

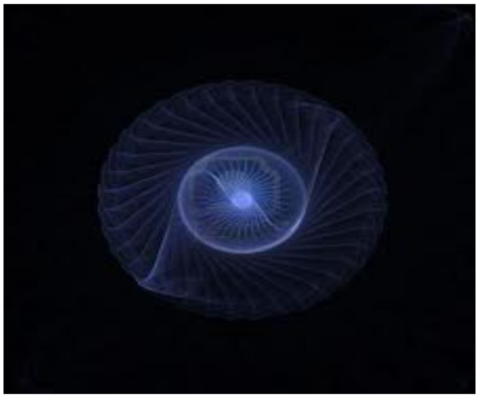
Sexual Reproduction with User Specified Pairs

The way users can breed together fractals is sexual reproduction with user specified pairs. Each fractal is generated at random before the users can breed two of their fractals together. Fractals that are generated at random do not have a parents but are given each an id. When a user picks two fractals from there flock to breed together this fractal is also given an id but is also identified by the id of both parents. Utilizing sexual reproduction with user specified pairs we plan to see evolution in fractals.

Flame Fractals

Flame fractals are a set of an iterated function system class. In our project we are using flame fractals and breeding them together through Flam-3. Flame fractals breed and mutate well. Utilizing flame fractals we plan to obtain and collect data from many users, while keeping users involved in the Fractal Breeders game. With flame fractals we hope to see continued evolution.





Python Based Program

To program the basic functionality of our program we used Web2py a python based website tool. The main function we used in our program was the flock method. The flock method is used to program the game.

```
from flam3 import *

@auth.requires_login()
def flock():
    goldRecord = db(db.gold.user==auth.user_id).select()[0]
    for v in request.vars:
        if v[0]=='x':
            id = int(v[1:])
            fractal = db(db.fractal.id==id).select().first()
            fractal.update_record(x=request.vars[v])
        if v[0]=='y':
            id = int(v[1:])
            fractal = db(db.fractal.id==id).select().first()
            fractal.update_record(y=request.vars[v])
        if v=='kill':
            fractal = db(db.fractal.id==request.vars[v]).select().first()
            fractal.update_record(alive=False)
            goldRecord.update_record(amount=goldRecord.amount+100)
        if v=='createNew' and goldRecord.amount>99:
```

```
        newFractalId = db.fractal.insert(parentA=None, parentB=None)
        fractal = db(db.fractal.id==newFractalId).select().first()
        genomeMakeFlame(db, fractal)
        renderFlame(db, fractal)
        goldRecord.update_record(amount=goldRecord.amount-100)
    if v=='breed' and goldRecord.amount>99:
        (idA,idB) = request.vars[v].split(',')
        newFractalId = db.fractal.insert(parentA=int(idA), parentB=int(idB))
        fractal = db(db.fractal.id==newFractalId).select().first()
        genomeMakeFlame(db, fractal)
        renderFlame(db, fractal)
        goldRecord.update_record(amount=goldRecord.amount-100)
db.commit()
query = (db.fractal.owner==auth.user_id) & (db.fractal.alive==True)
fractals = list(db(query).select())
if len(db(db.gold.user==auth.user_id).select()) == 0:
    # initialize gold
    db.gold.insert(amount=300, debt=300)
goldRecord = db(db.gold.user==auth.user_id).select()[0]
return dict(fractals = fractals, gold = goldRecord.amount, debt = goldRecord.debt)
```

HTML Based Program

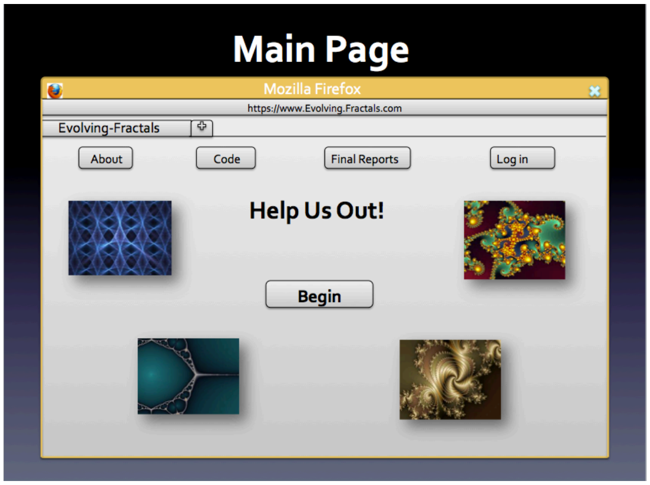
We also used the Web2py tool to program the web interface of our program in HTML. There are many basic functions that we programmed mainly in HTML, including the drag and drop function. The drag and drop function is programmed in java script inside of HTML. This function allows users to drag fractals on top of each other and breed them and drag fractals to the trash.

We particularly use the On Mouse Up function to drag and drop objects and to identify where the mouse is at a particular moment.

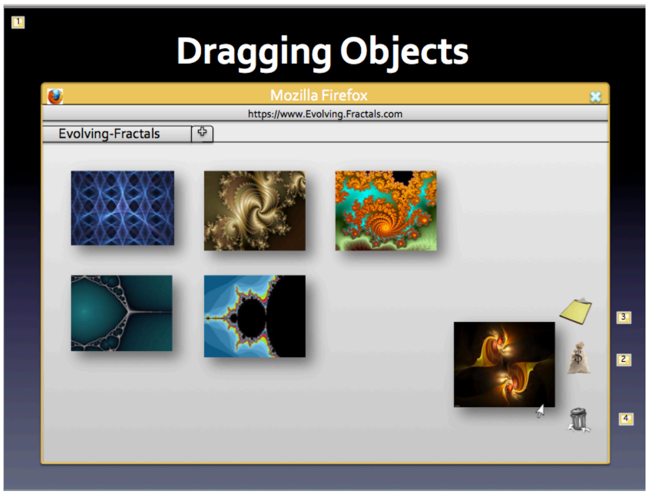
```
function OnMouseUp(e)
{
 // IE doesn't pass the event object
 if (e == null)
  e = window.event;
    if (_dragElement != null)
    {
        _dragElement.style.zIndex = _oldZIndex;
        form = document.getElementById("form");
        setHidden(form, 'x' + _dragElement.id, (_offsetX + e.clientX - _startX));
        setHidden(form, 'y' + _dragElement.id, (_offsetY + e.clientY - _startY));

        if (inside("MeatGrinder", e.clientX, e.clientY))
        {
            setHidden(form, 'kill', _dragElement.id);
        }
        else
        {
           var otherId = insideAnyBesides('drag', _dragElement.id , e.clientX , e.clientY);

            if (otherId != -1)
            {
                setHidden(form, 'breed', _dragElement.id + ',' + otherId);
            }
        }
        // we're done with these events until the next OnMouseDown
        document.onmousemove = null;
        document.onselectstart = null;
        _dragElement.ondragstart = null;

        // this is how we know we're not dragging
        _dragElement = null;

        _debug.innerHTML = 'mouse up';
        form.submit();
    }
}
```

Website Design

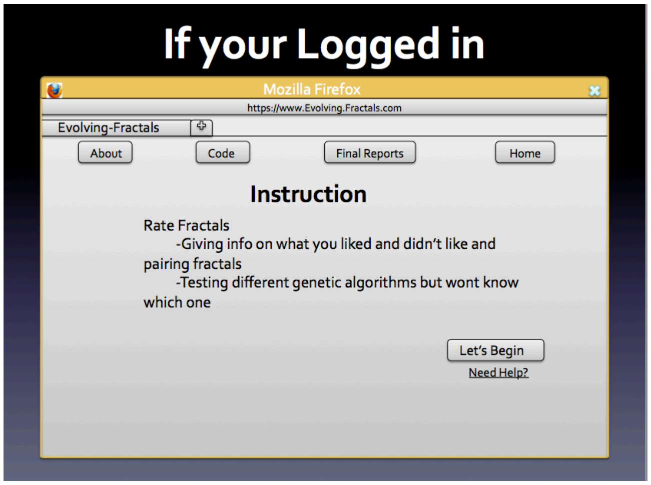
These are some clips of how we are planning for our game to look. We wanted the main page to look very simple so the users can start using the game right away and aren't trying to figure out how to work everything.



When dragging the objects the users are able to drag their fractal to the competition icon, bank icon, or the meat grinder icon. The user just simply clips then drags the fractal.



This clip shows what the page will look like if you are logged in, there are instructions that tell you how to start using the game. We are debating if we will keep the logging in part in the website, because we know that some people might not want to waste their time logging in, or them might not want to put any of their information on there.

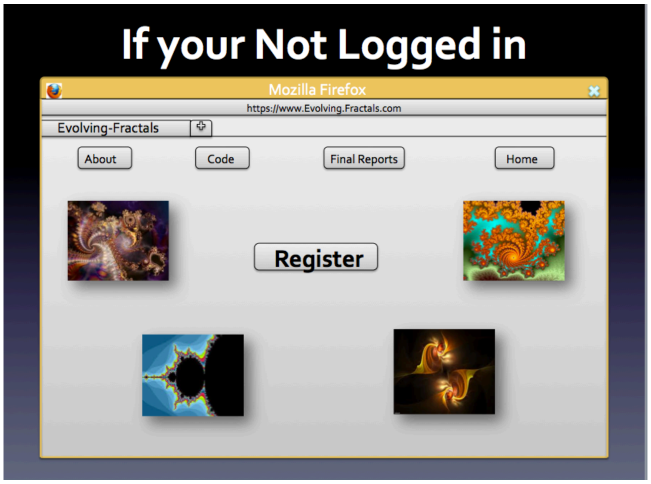


If your Logged in

Instruction

Rate Fractals
-Giving info on what you liked and didn't like and pairing fractals
-Testing different genetic algorithms but wont know which one

If you are not logged in we will have a simple register button that will lead you through logging in and the process of beginning the game



If your Not Logged in

Results

The project Evolutionary Fractal Art is not a website yet, but the basic functionality of the game is working. Even though the project does not have any collected data from users. The initial idea of the project was for users to log in and rate many fractals realizing that this was not an effective way to obtain users we changed the course of the project to gamification to be more successful at obtaining and keeping users.

We learned how to program in new languages such as Python, HTML, and java script. We also were able to work together as a team. By next year we hope to put this game online to collect data from many users. As we put it online we hope to show that fractal will evolve and produce more visually appealing fractals.

Conclusion

We have programmed the basic functionality of the game. The project Evolutionary Fractals Art and the Fractal Breeders game still needs to be polished before it can be put online. By next year we hope to have the Fractal Breeders game online with many users playing and enjoying the game, while giving us data on how fractals evolve.

Resources

"Fractal Evolution." *Fractal Evolution*. N.p., n.d. Web. 9 Dec. 2012.

<http://www.fractal.org/Bewustzijns-Besturings-Model/Fractal-Evolution.htm>.

"About | Electric Sheep." *About* | *Electric Sheep*. N.p., n.d. Web. 9 Dec. 2012.

<http://www.electricsheep.org/>.

"DarwinTunes." *PNAS Paper out Now*. N.p., n.d. Web. 7 Dec. 2012.

<http://darwintunes.org/pnas-paper>.

"The Image Breeder Project." *ImageBreeder*. N.p., n.d. Web. 8 Dec. 2012.

<http://www.imagebreeder.com/>.

"Breed." *Breed*. N.p., n.d. Web. 9 Dec. 2012. <http://notnot.home.xs4all.nl/breed/Breed.html>

Acknowledgements

Debra Johns

Wayne Witzel

Rebecca Koskela

Katie Richardson

Mark Fleharty