# **Evolutionary Fractal Art**

New Mexico Supercomputing Challenge

Final Report April 2, 2014

Team 140

Saturday Science and Math Academy

Team Member: Rachel Washington

Project Mentor: Wayne Witzel

Teacher: Debra Johns

Executive Summarypage 3
Introductionpage 4
Descriptionpage 5
Gamificationpage 6
Flame Fractalspage 7
Sexual Reproductionpage 8
Asexual Reproductionpage 9
Toolspage 10
Python Programpage11-16
JavaScript and HTML Programpage 17-25
Website Designpage 26-27
Processpage 28
Similar Researchpage 29
Data Analysispage 30- 31
Resultspage 32
Conclusionpage 33
Resourcespage 34
Acknowledgementspage 35

### **Executive Summary**

Flame fractals are iterated functions that create beautiful images. Flame fractals can create new fractals by asexual and sexual reproduction. By producing new fractals from either one or two parents flame fractals can create more beautiful offspring by selecting their parent's best traits. In result flame fractals evolve by inheriting desirable traits each generation. The Evolutionary Fractal Art project utilizes flame fractals reproductive properties as well as human preference to evolve fractal art.

Flame fractals are evolved through an interactive website called the Fractal Breeding Game. Users log in to the website to create flocks of fractals, enter contests, and barter on the market place. The website uses gamification or the utilization of game techniques to obtain larger quantities of data. As users continue to create and expand their fractal flocks. Fractals will become more pleasing with each generation evolving over time.

### Introduction

Evolution is the process that drives all life forward. In the project Evolutionary Fractal Art human preference is the only force driving evolution forward. Unlike many other similar research project where there are other factors driving evolution forward. An example of this is Darwin Tunes. Darwin Tunes is a research project where users log in to a website and listen to music clips. Users rate these music clips from 1 to 5, 1 being the lowest rating, while 5 is the highest. Music clips reproduce by sexual reproduction but do not allow the users to pick the parents. Music clips continued to evolve over time until there was a plateau where no improvement was found. The project Evolutionary Fractal Art attempts to eclipse this plateau by allowing the user to be the only force driving evolution forward.

The project Evolutionary Fractal Art utilizes two types of reproduction, human preference, and gamification to evolve fractal art. Fractal art is evolved through an interactive website called the Fractal Breeding Game. Users will breed fractals together by sexual and asexual reproduction, evolving a fractal flock. Evolution is quantified by the amount of the contest votes and the virtual money sold on the market place. By utilizing human preference, gamification, and two types of reproduction fractal should evolve over time.

#### Description

Evolution is the process that drives all life forward. In the project Evolutionary Fractal Art human preference is the only force driving evolution forward. Unlike many other similar research project where there are other factors driving evolution forward. An example of this is Darwin Tunes. Darwin Tunes is a research project where users log in to a website and listen to music clips. Users rate these music clips from 1 to 5, 1 being the lowest rating while 5 is the highest. Music clips reproduce by sexual reproduction but do not allow the users to pick the parents. The music clips continued to evolve over time until there was a plateau where no improvement was found. The project Evolutionary Fractal Art plans to eclipse this by allowing the human preference to be the only force driving evolution forward.

The project Evolutionary Fractal Art collects data from an interactive website called the Fractal Breeding Game. The Fractal Breeding Game allows users to customize their very own fractal flock. Users log into the interactive website and will breed, clone (mutate), and create fractals. The data from the game is analyzed by the fractals ancestors. The ancestors of the fractals are each counted individually, regardless of reproduction type. Each fractal starts with 1 ancestor and fractals obtain 1 ancestor from asexual reproduction and 2 from sexual reproduction. Because human preference is the only force driving evolution forward there should be a consistent relationship between the number of ancestors and the votes. The more votes a fractal receives the more ancestors the fractal has.

### Gamification

Gamification is the utilization of game techniques such as gaining prestige or earning virtual game money to obtain larger quantities of data. The Fractal Breeding Game uses gamification in order to obtain larger amounts of data from users. Gamification gives incentive to spend their time because they are having fun.

The driving force of incentive in The Fractal Breeding Game is the virtual game currency called gold. Each users starts out with only 700 gold to start their fractal flock. With the game currency users can create new random fractals, breed two fractals together, or clone (mutate) a fractal each for 100 gold a piece. Players can earn more gold by winning contests, selling owned fractals on the market place, and judging contests. Users earn 200 gold for winning a contest and 25 gold for judging a contest. With more gold players can add more fractals to their flock and can take their flock to new heights.

### Flame Fractals

Flame fractals are a set of iterated functions. Each flame fractal is defined by a set of 'flames'. Each 'flame' follows an algorithm and is defined by one of the 14 different flame functions. Flam fractals were invented in 1992 by Scot Draves. The flam fractals in the project are generated by flam3 a command line prompt.

$$F_0(x, y) = \left(\frac{x}{2}, \frac{y}{2}\right)$$
  $F_1(x, y) = \left(\frac{x+1}{2}, \frac{y}{2}\right)$   $F_2(x, y) = \left(\frac{x}{2}, \frac{y+1}{2}\right)$ 

Algorithm: Start at a random point and map to a new point according to one of the flame functions chosen at random. The color and intensity is determined according to how often a point is visited and what function brought it there.

The 14 Different Flame Functions



## Sexual Reproduction

There are two parents for each offspring. The users are allowed to pick both parents and be the only force driving evolution forward.

Fractal Family Tree



## Asexual Reproduction

Fractals are reproduced by only 1 parent. Even though new fractals are created from 1 parent they do not turn out the same in appearance. Instead the fractals are variations of each other. Fractal Family Tree



## Tools

Web2py: To program the website through Python, JavaScript, and HTML

Flam3: A command line-prompt to generate the flame fractals. This includes mutating the fractals and breeding the fractals together.

### Python Program

The website's main brain is programmed through python. Part of the python program is the flock function or the main page where most of the user activity occurs.

def flock():

#try:

```
mayCompleteRegisterLater()
```

anonLogin = session.anonymouslyLoggedIn

#to ensure the contest and the market close

currentContest(db)

currentMarket(db)

#except NameError:

# # don't worry of session variables for anonymous login weren't defined

```
# anonLogin = False
```

```
if len(db(db.gold.user==auth.user_id).select()) == 0:
```

# initialize gold

```
db.gold.insert(amount=700, debt=600)
```

db.commit()

goldRecord = db(db.gold.user==auth.user\_id).select()[0]

```
fractalsforselect = []
```

msg=None

for v in reversed(list(request.vars)):

if v=='kill':

fractalsforselect = request.vars['selected'].split(',')

fractalstokill = request.vars[v].split(',')

newGoldAmount = goldRecord.amount

for idtokill in fractalstokill:

fractal = getMyFractal(db, idtokill, auth)

if fractal != None:

fractal.update\_record(alive=False)

newGoldAmount= newGoldAmount + 100

fractalsforselect = list(set(fractalsforselect) - set(fractalstokill))

goldRecord.update\_record(amount=newGoldAmount)

break

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, request, fractal)

renderFlame(db, request, fractal)

goldRecord.update\_record(amount=goldRecord.amount-100)

break

if v=='breed' and goldRecord.amount>99:

(idA,idB) = request.vars[v].split(',')

parentA = getMyFractal(db, idA, auth)

parentB = getMyFractal(db, idB, auth)

if parentA != None and parentB != None:

newFractalId = db.fractal.insert(parentA=int(idA), parentB=int(idB))

fractal = getMyFractal(db, newFractalId, auth)

genomeMakeFlame(db, request, fractal)

renderFlame(db, request, fractal)

goldRecord.update\_record(amount=goldRecord.amount-100)

fractal.update\_record(ancestorCount=fractal.ancestorCount+2)

fractalsforselect=(idA, idB, newFractalId)

break

```
if v=='mutate' and goldRecord.amount>99:
```

(idA) = request.vars[v]

parentA = getMyFractal(db, idA, auth)

if parentA != None:

newFractalId = db.fractal.insert(parentA=int(idA), parentB=None)

fractal = getMyFractal(db, newFractalId, auth)

genomeMakeFlame(db, request, fractal)

renderFlame(db, request, fractal)

goldRecord.update\_record(amount=goldRecord.amount-100)

fractal.update\_record(ancestorCount=fractal.ancestorCount+1)

fractalsforselect=(idA, newFractalId)

break

if v=='enterContest':

fractalid = request.vars[v]

fractal = getMyFractal(db, fractalid, auth)

if anonLogin:

msg="You can not enter a contest if you are anonymous."

elif fractal != None:

contest = nextContest(db)

# Delete any previous entry of the user

contestants = getContestants(db, contest)

for contestant in contestants:

if contestant.fractal.owner == auth.user\_id:

db(db.contestant.fractal==contestant.fractal.id).delete()

# Insert this new entry

db.contestant.insert(contest=contest, fractal=fractalid)

db.commit()

redirect("entries")

break

```
if v=='gotomarket':
```

fractalid = request.vars[v]

fractal = getMyFractal(db, fractalid, auth)

if anonLogin:

msg="You can not sell a fractal on the market if you are anonymous."

elif fractal != None:

gotomarket = nextMarket(db)

fractalsonsale=getFractalsonsale(db, gotomarket)

*#* for fractalonsale in fractalsonsale:

# if fractalonsale.fractal.owner == auth.user\_id:

# db(db.fractalonsale.fractal==fractalonsale.fractal.id).delete()

initialbid = request.vars["initialbid"]

goldRecord.update\_record(amount=goldRecord.amount-25)

fractal.update\_record(owner=None)

db.fractalonsale.insert(market=gotomarket, fractal=fractal.id, initialbid=int(initialbid),

```
seller= auth.user_id, bid=int(0))
```

db.commit()

break

db.commit()

query = (db.fractal.owner==auth.user\_id) & (db.fractal.alive==True) #&

(db.fractal.sold==False)

fractals = list(db(query).select())

goldRecord = db(db.gold.user==auth.user\_id).select()[0]

username = ((auth.user.first\_name) +" "+ (auth.user.last\_name))

newMessages = db((db.message.received==False) &

(db.message.receiver==auth.user\_id)).select()

hasnewMessages = (len(newMessages) > 0)

#checks for new users, then makes them old users

aNewUser = len(db(db.newUsers.user==auth.user\_id).select()) > 0

db(db.newUsers.user==auth.user\_id).delete()

db.commit()

return dict(fractals = fractals, gold = goldRecord.amount, debt = goldRecord.debt, anonLogin

= anonLogin, username = username, fractalsforselect = fractalsforselect, msg=msg, hasnewMessages=hasnewMessages, newMessages=newMessages, session=session, aNewUser=aNewUser)

### HTML/JavaScript Program

The appearance of the website is programmed through HTML and JavaScript. This is a small part of the corresponding code to the Python that is the image of the website.

HTML

```
<title>Fractal Breeding Game</title></head>
```

<body>

```
<form id="form">
```

```
{{=username}}'s Flock
```

```
{{if session.whichToRegister != None:}}
```

```
WhichToRegister{{=session.whichToRegister}}
```

 $\{\{pass\}\}$ 

```
<img onclick="openMessages()" width="80" height="80"
```

```
src="../static/images/message.png"/>
```

```
<img onclick="openQuestion()" width="40" height="60"
```

```
src="../static/images/question.png"/>
```

```
<a href="user/logout">logout</a>
```

```
{{if anonLogin:}}
```

<a href="registerFromAnon">register</a>

```
{{else:}}
```

```
<a href="user/profile">change profile</a>
```

{{pass}}

```
Your Gold = {{=gold}}
```

```
{{if anonLogin:}}
```

 $\{\{else:\}\}$ 

```
<a href="user/change_password">change password</a>
```

 $\{\{pass\}\}$ 

```
<div style ="height:225px; border:1px solid black; overflow:auto; white-space: nowrap;
```

```
width:100%;">
```

```
{ { for fractal in fractals: } }
```

```
<span id="{{=fractal.id}}"><img src="{{=URL('download', args=fractal.image)}}"
```

```
width="200px" onClick="select({{=fractal.id}})"/></span>
```

 $\{\{pass\}\}$ 

</div>

<a href="contest"><img width="130"

src="../static/images/beautycontest.jpg"/></a>

```
<a href="market"><img width="180"
```

```
src="../static/images/market.jpg"/></a>
```

<input type="submit" value="create for 100 gold" id="generate" onClick="createNew()"/>

<input type="submit" value="delete to earn 100 gold" id="button" onClick="return kill()"/>

<input type="submit" value="sell on market" id="button" onClick="return gotomarket()"/>

<input type="submit" value="enter contest" id="button" onClick="return enterContest()"/>

{{for fractal in fractals:}}

 <t.---Placeholder for fractal image---> <t.---Placeholder for scrap button--->

```
JavaScript
function breed()
{
  if(selectedFractals.length!==2)
  {
       alert("You must select two fractals to breed")
       return false
  }
  else
  {
       form = document.getElementById("form")
       setHidden(form, 'breed', selectedFractals.join())
       if({=gold})<100)
       {
              alert("You do not have enough gold!")
       }
       form.submit()
  }
}
function mutate()
{
  if(selectedFractals.length!==1)
  {
```

alert("You must select one fractal before you can make a sibling")

```
return false
  }
  else
  {
       form = document.getElementById("form")
       setHidden(form, 'mutate', selectedFractals.join())
       if({{=gold}}<100)
       {
              alert("You do not have enough gold!")
       }
       form.submit()
  }
}
function enterContest()
{
       if(selectedFractals.length !== 1)
  {
       alert("You must select 1 fractal to enter in the contest")
       return false
  }
       form = document.getElementById("form")
       setHidden(form, 'enterContest', selectedFractals.join())
       form.submit()
}
```

```
function gotomarket()
{
       if(selectedFractals.length !== 1)
  {
       alert("You must select 1 fractal to sell on the market")
       return false
  }
  form = document.getElementById("form")
  if ({{=gold}}<25)
  {
     window.alert("You do not have enough gold!")
     return false
  }
  else
  {
    var d=confirm("Do you want to enter this into the market? This will cost 25 gold initially")
    if (d==true)
    {
      var initialbid=window.prompt("What do you want to set the initial bid price for?","100")
      setHidden(form, 'initialbid', initialbid);
      setHidden(form, 'gotomarket', selectedFractals.join())
      form.submit()
     }
```

```
}
}
function kill()
{
       form = document.getElementById("form")
       if(selectedFractals.length == 0)
  {
     //when the user doesnt select a fractal
       window.alert("You must select a fractal to delete")
       return false
  }
  else if(selectedFractals.length == 1)
  {
    //when the user only selects 1 fractal
       var d=confirm("Do you want to delete this fractal to earn 100 gold")
  }
  else
  {
    //when the user selects multiple fractals
     var d=confirm("Do you want to delete these fractals to earn
"+(selectedFractals.length*100).toString()+" gold")
  }
  if (d==true)
```

{

```
setHidden(form, 'kill', selectedFractals.join())
       setHidden(form, 'selected', selectedFractals.join())
               form.submit()
  }
}
function scrap(id)
{
       form = document.getElementById("form")
     //if you want to delete the fractal
       var d=confirm("Do you want to delete this fractal to earn 100 gold")
  if (d==true)
  {
       setHidden(form, 'kill', id)
       setHidden(form, 'selected', selectedFractals.join())
               form.submit()
  }
}
function createNew()
```

```
{
```

```
form = document.getElementById("form")
setHidden(form, 'createNew', ")
```

```
if ({{=gold}}<100)
```

{

window.alert("You do not have enough gold!")

25

## }

}

form.submit()

## Website Design

### Visit the website at fractalbreedinggame.org







delete





### Process

The Evolutionary Fractal Art project started over a year ago. The concept was simple to have users rate log into a website and rate fractals. Users would first generate their own fractals and other users would then judge the fractals. The first website did not implement gamification. It was very similar to Darwin Tunes and users would just rate fractals from 1 to 5. The first website compared 3 different types of reproduction, sexual reproduction with user specified pairs, asexual reproduction with non-user specified pairs, and asexual reproduction. After the first evaluations the judges' feedback was incredibly helpful. Realizing that users had no incentive to provide large quantities and honest feedback the project immediately changed course. During the first challenge the website was not fully completed and no data was collected. But main infrastructure and programming was done and only included one type of reproduction, sexual reproduction with user specified pairs.

Later on the website main programming had been completed enough to be released an to collect data. The website still needs more users to collect larger quantities of data. New users log into the website every day. The website still gets continuous upgrades to improve the user experience and to increase user participation. Over the past 2 years the project has grown from a concept to being close to reality.

### Similar Research

Darwin Tunes: A music research project where sounds evolve and become better and better using human preference. Darwin Tunes uses sexual reproduction with random pairs. Eventually the music does not improve but instead hits a plateau where there is no evolution to be found. Apophysis7x: A fractal flame editor, with a mutation function that uses asexual reproduction to make new fractals.

Electric Sheep: A distributed computing project that mutates and crosses-over different fractal flame animations to make new ones, based on human preference.

What makes Evolutionary Fractal Art Different: Human preference is the only force driving evolution forward.

### Data Analysis

The Evolutionary Fractal Art project collects data from the Fractal Breeding Game contests. To see if fractals are evolving, the number of unique ancestors categorizes each fractal. For each category, the probability is found of winning a contest by counting the number fractals in the category that have won a contest and dividing by the total number of fractals in that category. Ancestors are in groups of 20, except for ancestor 1 or the fractal with no parents. Currently there has been a trend in data that fractals with more ancestors have a higher chance of winning a contest. On a whole fractal with 1 ancestor have a 0.19% chance of winning while fractals with multiple ancestors have 0.337% chance of winning a contest. With the current amount of data there is not enough to validate if the fractals are evolving overtime. Each contest was won by only a few votes and the data is subject to the biases of the breeders in choosing which fractal to enter into each contest. A future plan is to collect many votes for a super-contest among all the past winning fractals and see whether or not fractals with higher ancestor counts get more votes.





### Results

From the last year in Super Computing Challenge the Evolutionary Fractal Art project has come a long way. The website has improved greatly. These improvements include a mobile devices being able to run the website in a browser and an improved user interface. The website also obtained a small database of users that is continually increasing.

There has been data collected where there is a trend observed that fractals with more ancestors have a higher chance of winning a contest. There is not enough information to truly analyze the data; this includes comparing reproduction types and individually comparing ancestor counts. There is also not enough data to make a concrete conclusion that the fractals are evolving over time.

## Conclusion

The project greatly improved from last year. Unfortunately there is not a large enough database of users and data to come to a solid conclusion. From participating in SC Challenge I have learned computer programming skills, presentation skills, and how to write a report. Next year I plan to continue the Evolutionary Fractal Art project and collect enough data to form a solid conclusion.

### Resources

1)"Fractal Evolution." *Fractal Evolution*. N.p., n.d. Web. 9 Dec. 2012. <a href="http://www.fractal.org/Bewustzijns-Besturings-Model/Fractal-Evolution.htm">http://www.fractal.org/Bewustzijns-Besturings-Model/Fractal-Evolution.htm</a>>.

2)"About | Electric Sheep." *About | Electric Sheep.* N.p., n.d. Web. 9 Dec. 2012. <a href="http://www.electricsheep.org/">http://www.electricsheep.org/</a>>.

3)"DarwinTunes." *PNAS Paper out Now.* N.p., n.d. Web. 7 Dec. 2012. <a href="http://darwintunes.org/pnas-paper">http://darwintunes.org/pnas-paper</a>>.

4)"The Image Breeder Project." *ImageBreeder*. N.p., n.d. Web. 8 Dec. 2012. <a href="http://www.imagebreeder.com/>.</a>

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

## Acknowledgments

Albuquerque ACTSO Chapter

New Mexico Super Computing Challenge

Saturday Science and Math Academy

Shavonne Betts

Jaci Hernandez

Debra Johns

Family and friends