

// readme.txt file :: project overview

CLASS: CPSC-335 - Algorithms | Section-02

PROJECT #1: Cella-Ant-x15

TEAM NAME: BurritoUI

TEAM MEMBERS:

- Mason Godfrey
- Jared Dyreson
- Brian Lucero

INTRODUCTION:

Cella-Ant-x15

A cellular automaton (Cella) variant of Langton's Ant written in Javascript and P5JS for CS 335.

Christopher Langton, an American computer scientist, conceptualized the initial ant logic.

Visually, an animation of an "ant" starts out on a 2 dimensional grid containing black and white cells, and it follows a set of rules which determine the ant's path.

1. If the ant is on a black square, it turns right degrees and moves forward one unit.
2. If the ant is on a white square, it turns left degrees and moves forward one unit.
3. When the ant leaves a square, it inverts the color.

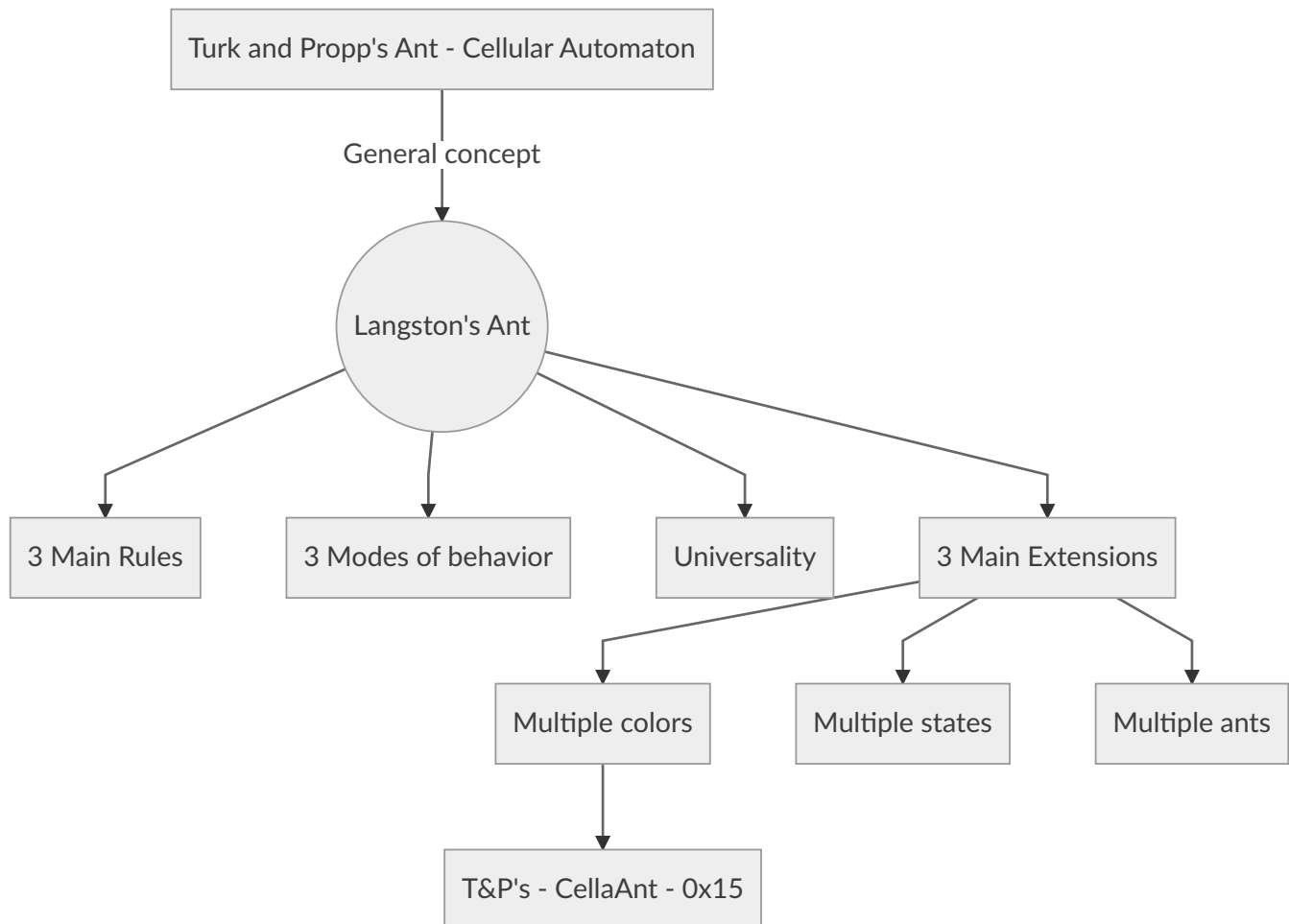
Despite having such a simple instruction set, the navigation pattern becomes more complex and emergent behavior is observed. Even more so when Turk and Propp thought to involve

more colors than just black and white, thus described as a *Cellular Automaton*, “Cella-Ant-x15”.

The concepts and topics shown below, form the compositional model of Langton's Ant, the foundational idea.

The text in this file will also produce a neat diagram of the concept inner-relationships, if the text in this readme file is pasted into a markdown format extension that supports it. I used the following extension:

Written with [StackEdit](#).



In this project:

The cella will be shown in a 2D grid of black, red, yellow, blue, and green cells; initially all are black. Each color is associated with a change of direction: to the Left or to the Right of the Ant's current heading, based on the Ant's number and the index order of that color in the color sequence.

For Ant #x15, the Ant's number is 15 hexadecimal = 10101 binary (= 21 decimal) ==> cell turning states of LRLRL, where a binary 1 bit means turn Left, and a 0 means turn Right.

Our team, translated these logic instructions into a **general algorithm** enabling the ant to

```
setMovementDirection(cellState)
```

The cellState value, or the color, will be deduced to a single binary value, 0 or 1:

```
var antDir = (this.hex % (32 >> cellState)) >> (4 - cellState);
```

```
setMovementDirection(cellState) {  
var antDir = (this.hex % (32 >> cellState)) >> (4 - cellState);  
  
if(antDir == turnDirection.LEFT)  
    this.facing--;  
else  
    this.facing++;  
  
if(this.facing < direction.NORTH)  
    this.facing = direction.WEST;  
else if(this.facing > direction.WEST)  
    this.facing = direction.NORTH;  
}
```

ZIP CONTENTS:

File **documentation/exported/Cella_Ant_Big_O_Analysis.pdf**

This file contains the 1-page paper describing the analysis of the Big-O running time of our algorithm

File **documentation/readme.txt**

This file

File **documentation/readme.pdf**

This file but in a PDF rendition

File **runtime/Display.js**

This file contains 2 functions that initially build and maintain the environment.
All of the other files organize the object classes which sum up these two functions

- **setup()**: to be run when a web page initially runs
- **draw()**: updates the program every frame.

File **runtime/Grid.js**

This file provides the instantiation of cells, a matrix containing said cells and logic to change the color of cells at arbitrary positions.

Contains the implementations for 3 classes, needed for functions in Display.js file:

Cell class: Defines the dimensions, state, color, and ant properties for each cell object within our grid. Which are then used by methods to display the cell grid, and ant triangle animation.

Point class: Model after the ordered pair type of object which includes x & y coordinates. This is useful for determining (x,y) position within the 2-dimensional grid.

grid class: Initializes and maintains the dimension properties of a grid. Methods within the class use to display real time updates to a cell color, ant triangle animation, or if it needs to be redrawn.

File **runtime/RedAnt.js**

This file defines the Position-Direction-Turn relationship properties later used to implement the RedAnt class (RedAnt class handles the cellState or color portion of the logic and its navigation instruction logic. Cell color determines turn, further influencing direction, thus eventually establishing a simple navigation pattern, due to value dependence.

File **runtime/p5.js**

This is the P5 package. It is loaded inside the html.

File **runtime/driver.html**

This file will start the program, and run it within a webpage browser tab

File **assets/pix-js-cellaAnt-10s.jpg**

A snapshot of the example webpage after the program has run for 10 seconds

File **assets/pix-js-cellaAnt-15m.jpg**

A snapshot of the example webpage after the program has run for 15 minutes

File **assets/pix-js-cellaAnt-2h.jpg**

A snapshot of the example webpage after the program has run for 2 hours

INSTALLATION AND RUNNING:

1. Extract the .zip file into a folder.
2. Navigate to the driver.html file within the folder. Right-click on the file and select "Open".
The p5 program should start immediately. You should see what looks to be a 41x41 grid (white lines on black background) with and some of the grid cells colored. See the picture **assets/pix-js-cellaAnt10s.jpg** for reference.

KNOWN BUGS:

Currently, no known bugs exist nor undesired program outcomes.

WARNINGS

The team did not experience any warnings, and has run error free.

TESTING

Following installation instruction, above, watched it run. Appears ready to ship and downloadable by anyone

CREDITS

The team referenced professor Charles Siska's example program of the same project, and modeled the overall program design and file management.

"Some code was borrowed and modified from Stuart's book.
Introducing JavaScript Game Development: Build a 2D Game from the
Ground Up, by Graeme Stuart, 2018, 209 pages.
And, of course, thanks to the HTML and P5.js developers."
– Siska

Mason

<https://p5js.org/reference/>

referenced code snippets and general syntax for p5js functionalities

Jared

https://en.wikipedia.org/wiki/The_C_Programming_Language

referenced Chapter 6, Section 2 of the text.

Brian

https://en.wikipedia.org/wiki/Langton's_ant

referenced various explanations of the concepts and logic
to accurately produce the project documentation

<https://mathworld.wolfram.com/LangtonsAnt.html>

more referencing for concepts, logic, etc.

