**Functional Requirements**

* Input a non-negative integer for the state # to approximate.
* Compute the next state based on the Cella Rule-150 of the graph of filled/empty cells.
* Output the next state to the screen.
* Repeat those tasks until the right state #.

**Use Case Diagram**

User

Input State #

Compute nth State

Clear Graph

**Sequence Diagram**

User

Webpage

ALT

[valid] : Show Result

[invalid] : Show Error Message

Compute n-th state

Give starting state and final state

**Class Diagram**

Canvas

Animation

p5 *<<package>>*

Cella150.js

+ displayGrid()

+ generateNthLevel(n : int)

+ updateTiles()

**Data Flow Diagram (Level 0)**

Input Data

1

`

Cella Rule Computation

User

Result Graph

**Data Flow Diagram (Level 1)**

Input Data

User

1.1

Client-Side Input Acceptor

1.2

Generate  
Nth Level

N, Graph

1.3

Draw Graph

New Graph

Result Graph

**Website Design**

Cella Rule 150 Program

Enter the Nth State To Generate:

Generate

401

401

**Pseudocode**

grid = int[401][401]

n = 0

maxLevel = 0

Function displayGrid( n : Integer )

For each column c in grid:

If grid[r][c] == White

Set display grid at (r, c) to White

Else

Set display grid at (r, c) to Black

Function clearGrid( )

For each row r in grid:

For each column c in grid

If r == 0

Grid[r][c] = 1

Else

Grid[r][c] = 0

Grid[0][200] = 0

Function generateNthLevel( n : Integer )

If n is null OR n is not a number OR n < 0:

Print ‘Error: Invalid nth state, retry again’

Else

For each column c in grid:

grid[n][c] = calculateNextState(n, c)

displayGrid( )

Function calculateNextState( r : Integer, c : Integer )

maxcols = num columns in grid

count = 0

If c >= 1 AND grid[r-1][c-1] == 1

count = count + 1

If grid[r-1][c] == 1

count = count + 1

If grid <= maxcols – 1 AND grid[r-1][c+1] == 1

count = count + 1

If count == 1 OR count == 3

Return Black

Else

Return White

Function setMaxLevel( level : Integer )

If n is null OR n is not a number OR n < 0 OR n > 401:

Print ‘Error: Invalid nth state, retry again’

Else

maxLevel = level

Function draw( )

If maxLevel >= 0 AND n == 0

clearGrid()

For each row r in grid

displayGrid(r)

if maxLevel > 0

n = n + 1

Else

g\_frame\_cnt = g\_frame\_cnt + 1

if g\_frame\_cnt % g\_frame\_mod == 0 AND n < maxLevel

generateNthLevel(n)

n = n + 1

if n == maxLevel

maxLevel = 0

n = 0

**Activity Diagram**

Input n

No

No

No

Yes

Yes

Yes

Show nth Graph

Determine Cell Value

Is Through  
all cells

Is n valid?

Show Error Message

Is Through n iterations?

Show nth Graph