

# Modeling Development in *C. elegans*

A Major Qualifying Project Report:  
submitted to the Faculty  
of the  
WORCESTER POLYTECHNIC INSTITUTE  
in partial fulfillment of the requirements for the  
Degree of Bachelor of Science

By

Daniel Brandon BB '13

Josie Cromartie BB '13

William Decker CS '13

***Advisors:***

Professor Elizabeth Ryder, Biology Advisor

Professor Matthew Ward, Computer Science Advisor

# Table of Contents

<b>Abstract</b> .....	<b>5</b>
<b>1 Introduction</b> .....	<b>6</b>
<b>2 Background</b> .....	<b>9</b>
<b>2.1 Previous Simulation Worm Project</b> .....	<b>9</b>
<b>2.2 Embryonic Development of <i>C. elegans</i></b> .....	<b>11</b>
2.2.1 Proliferation .....	11
2.2.2 Organogenesis and Morphogenesis.....	13
2.2.3 Wnt signaling, <i>mig-10</i> , and neuronal development.....	15
2.2.3 Programmed Cell Death.....	17
<b>2.3 Inverted selective plane illumination microscopy (iSPIM)</b> .....	<b>19</b>
<b>2.4 Computer simulations</b> .....	<b>20</b>
2.4.1 Simulation uses .....	20
2.4.2 Types of simulations .....	21
<b>3 Methodology</b> .....	<b>22</b>
<b>3.1 Biological Methodology</b> .....	<b>22</b>
3.1.1 Growing and maintaining worms.....	22
3.1.2 Strains .....	22
3.1.3 Strain Design .....	24
<b>3.2 Simulation Methodology</b> .....	<b>26</b>
3.2.1 Programing Languages and Libraries .....	26
3.2.2 Algorithms.....	27
3.2.2.1 Metaballs .....	28
3.2.2.2 Marching Cubes .....	29
<b>4 Simulation Design</b> .....	<b>31</b>
<b>4.1 Design</b> .....	<b>31</b>
4.1.1 Data Structure.....	31
4.1.2 Simulation .....	33
4.1.3 Rendering.....	33
<b>4.2 User Interaction and User Interface</b> .....	<b>34</b>
4.2.1 Navigation .....	35
4.2.2 Clipping .....	35
4.2.3 Coloring.....	36
4.2.4 Programmed cell death:.....	38
4.2.5 Photo Gallery .....	38
<b>4.3 Online Capability</b> .....	<b>39</b>
<b>4.6 Cell Visualization</b> .....	<b>40</b>
4.6.1 Two Dimensional Model .....	40
4.6.2 Three Dimensional Model.....	41
<b>5 Results and Discussion</b> .....	<b>44</b>
<b>5.1 Previous Model Comparisons</b> .....	<b>44</b>
<b>5.2 Existing Infrastructure</b> .....	<b>46</b>
<b>5.3 Embryonic Marker</b> .....	<b>47</b>
5.3.1 Method 1: egg visualization on slides .....	48

5.3.2 Method 2: Embryo Visualization <i>in utero</i> .....	48
5.3.3 Cross and future recommendations .....	50
<b>5.4 Future Model Implementations.....</b>	<b>50</b>
<b>6 References .....</b>	<b>53</b>
<b>Appendix A: Crosses .....</b>	<b>55</b>
Cross 1: .....	55
Cross 2: .....	56
<b>Appendix B: Supplementary Images.....</b>	<b>57</b>
<b>Appendix C: Presentation Poster.....</b>	<b>58</b>
<b>Appendix D: Procedures .....</b>	<b>59</b>
Maintaining a line: .....	59
Making an N2 mating plate .....	59
Bleaching Worms to Remove Contaminants.....	60
<b>Appendix E: References and sources for programming Languge’s and Libraries .....</b>	<b>61</b>
<b>Appendix F: User’s Guide .....</b>	<b>62</b>
<b>Appendix G: Programmer’s Guide.....</b>	<b>63</b>

## Table of Figures

FIGURE 1: DIAGRAM OUTLINING SIMULATION DEVELOPMENT AND REFINEMENT PROCESS .....	8
FIGURE 2: SCREENSHOT OF PREVIOUS SIMULATION WORM MODEL .....	10
FIGURE 3 FOUNDER CELL LINEAGE TREE.....	12
FIGURE 4: ASYMMETRICAL SPLIT IN EARLY CELL DIVISION .....	13
FIGURE 5: LIFE CYCLE OF <i>C. ELEGANS</i> SHOWING DIFFERENT STAGES OF EMBRYOGENESIS AND LARVAL DEVELOPMENT TO THE POINT OF ADULTHOOD .....	14
FIGURE 6: ANTERIOR AND POSTERIOR DIRECTED NEURONAL MIGRATIONS .....	16
FIGURE 7: PART OF CELL LINEAGE ILLUSTRATING CELL DEATH (SULSTON ET AL., 1983) .....	18
FIGURE 8: APOPTOTIC CELLS LOCATED IN EMBRYO .....	19
FIGURE 9: VISUAL REPRESENTATION OF METABALL ALGORITHM .....	29
FIGURE 10: VISUAL REPRESENTATIONS OF MARCHING CUBES .....	30
FIGURE 11: APPROACH TO DATA STRUCTURE .....	32
FIGURE 12: DENSITY FUNCTION .....	34
FIGURE 13: SCREENSHOT OF NAVIGATION CONTROLS .....	35
FIGURE 14: SCREENSHOT OF CLIPPING CONTROLS .....	36
FIGURE 15: CLIPPING TOOL IN ACTION .....	36
FIGURE 16: CELL COLORING FEATURE .....	37
FIGURE 17: PHOTO GALLERY AS SEEN ON WWW.SIMWORM.HEROKUAPP.COM .....	39
FIGURE 18: SIMULATION AS SEEN ON GOOGLE CHROME AT WWW.SIMWORM.HEROKUAPP.COM/3DFINISH.....	40
FIGURE 19: TWO-DIMENSIONAL SIMULATION OF EARLY DIVISIONS .....	41
FIGURE 20: THREE-DIMENSIONAL MODEL - VERSION ONE.....	42
FIGURE 21: THREE-DIMENSIONAL MODEL - VERSION TWO .....	42
FIGURE 22: LABORATORY MICROGRAPHS <i>C. ELEGANS</i> COMPARED TO SCREENSHOTS OF SIMULATION.....	44
FIGURE 23: CURRENT SIMULATION COMPARED TO PREVIOUS SIMULATION.....	46
FIGURE 24: GRAVID ADULT WORM ANATOMY.....	49
FIGURE 25: COMPARISON OF HOMOZYGOUS AND HETEROZYGOUS <i>PIE-1::H2B::WCHERRY</i> EMBRYONIC MARKER VIEW IN GRAVID HERMAPHRODITE .....	49

## List of Tables

TABLE 1: FLUORESCENT MARKER INDEX.....	23
TABLE 2: WORM STRAIN INDEX .....	23

## List of Equations

EQUATION 1: GOAL OF CROSS .....	24
EQUATION 2: CROSS 1 .....	24
EQUATION 3: CROSS 2 .....	25
EQUATION 4: CROSS 3 .....	25
EQUATION 5: CROSS 4 .....	26



## **Abstract**

This project aimed to model the embryonic development of *Caenorhabditis elegans* from fertilization to hatching to serve as a dynamic research and educational aid that could be added to or modified by future research. Existing knowledge from current literature and mutant strains generated in lab were used to design and refine the processes modeled.

The simulation was rendered using metaballs and marching cubes to provide a three-dimensional computer simulation that can help further understanding of the development process.

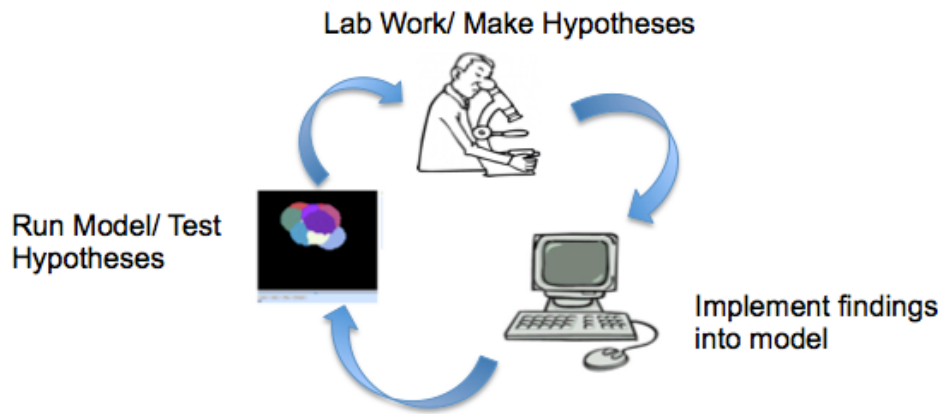
# 1 Introduction

A growing trend in many scientific fields is the use of computational models to better visualize data as well as to simulate different processes. Computer-based models provide many benefits to researchers and allow them to simulate and study many experiments that would be too impractical or expensive to achieve in the real world. Computational models can also improve understanding of many concepts by simplifying data and presenting them visually in a way that would not have worked in a simple figure. Dynamic simulations let observers make changes and test hypotheses in a system with multiple interactions occurring. In developing a model, information comes from scientific literature and experimental data. As the model is developed and tested, it is compared to the real life data and refined until it is more realistic, and can continue to help generate new hypotheses to be tested.

The purpose of this project was to create a computational model of the embryonic development of the model organism *Caenorhabditis elegans* (*C. elegans*). *C. elegans* is an ideal organism to work with because it is a small organism about which much is known. The entire cell lineage of approximately 1000 cells of *C. elegans* is known. Of those 1000 cells, 558 develop during embryogenesis, while the remaining cells develop after hatching (Sulston, Schierenberg, White, Thomas, 1983). The lifespan of *C. elegans* is also relatively short so that it is an easy organism to maintain and conduct experiments with without having to wait too long for generations.

Embryogenesis is approximately 12-14 hours, while the entire lifespan of the organism is 2-3 weeks (Rutgers, 2012). It is important to note that reproductive maturity happens after a span of 3 days. This simplicity and knowledge allows scientists to follow the development of specific pathways that consist of more cells and are more complex in larger organisms.

A previous study worked on producing a model that portrayed the first 100 cells of embryogenesis. The goal of our project was to improve upon this idea and create a model that went through the complete stages of embryogenesis and improved upon the design of the previous project to make realistic cell shapes for the development of *C. elegans*. One of the goals of this project was to develop an agent-based model where each cell in the model could act as an autonomous agent following a programmed set of rules (Bonabeau, 2002). The intercellular communication involved in embryonic development is very complicated and difficult to represent. An agent based model where each cell can be represented as an autonomous agent and respond differently to signaling allows better visualization of the interaction of signals and cells. Agent based models are very advantageous because they allow better visualization of emergent or unanticipated behaviors that would be more difficult to predict or portray with the use of only static figures (Bonabeau, 2002). Another goal of our project was to follow the migration of the CAN neuron throughout embryogenesis. Tracing this migration would allow us to test the accuracy of our model by programming a set of conditions and watching the model's response, then comparing to the experimental migrations of the neuron in the laboratory. If the model does not match the experimental results, the model will be fine-tuned and adjusted, and the hypothesis retested. Figure 1 illustrates this relationship between programming the model, using it to develop new hypotheses, generating new data in lab and using that to refine the model.



**Figure 1: Diagram outlining simulation development and refinement process**

The relationship between the model and the data generated by experimenters is a continuous cycle. The initial model is designed based off of previous research done by a scientist that has been published in the literature. Once the model is created it can be used by researchers to generate new hypothesis and areas of interest to study in lab. The new findings generated by these hypotheses are implemented in the model. As the model is refined new hypotheses are generated and the cycle continues.

The aim of this project was to create and accurately develop a model that represented a basic view of the embryogenesis of *C. elegans*, but that could be able to be used and modified by other researchers. The hope was to have an open source model that would grow more complex and accurate as new experimentation is done and more data added to the model to provide a more accurate representation. It is possible in the future that this model could serve as a basis for other models of later developmental stages of *C. elegans* and other simple organisms.

As well as serving as a research aid, this model could also be used as a classroom tool to help students better understand different aspects of development and cell signaling. Student learn better when exposed to different learning methods and having a visual model to help demonstrate development phases or cell signaling could help them to reinforce and better understand the knowledge they are receiving in the classroom.

## 2 Background

In order to better understand the project and simulation that we worked on, basic knowledge of the model organism and general basics of simulation use, as well as previous work on the model, is important to have. The following section provides a brief overview of the biological and computer science aspects of this project.

### 2.1 Previous Simulation Worm Project

This project is based in part on the work of a previous MQP completed by students, modeling the first 100 cell divisions of *C. elegans*. Their work focused on modeling cellular interactions, divisions, and positions using metaballs, collision detection and momentum equations (Bogdanova, Jajosky, Lloyd & Stolzar, 2004). This version represented models as simple spheres that lacked the flexibility of shape and ability to fit to the egg shape, as seen in Figure 2.

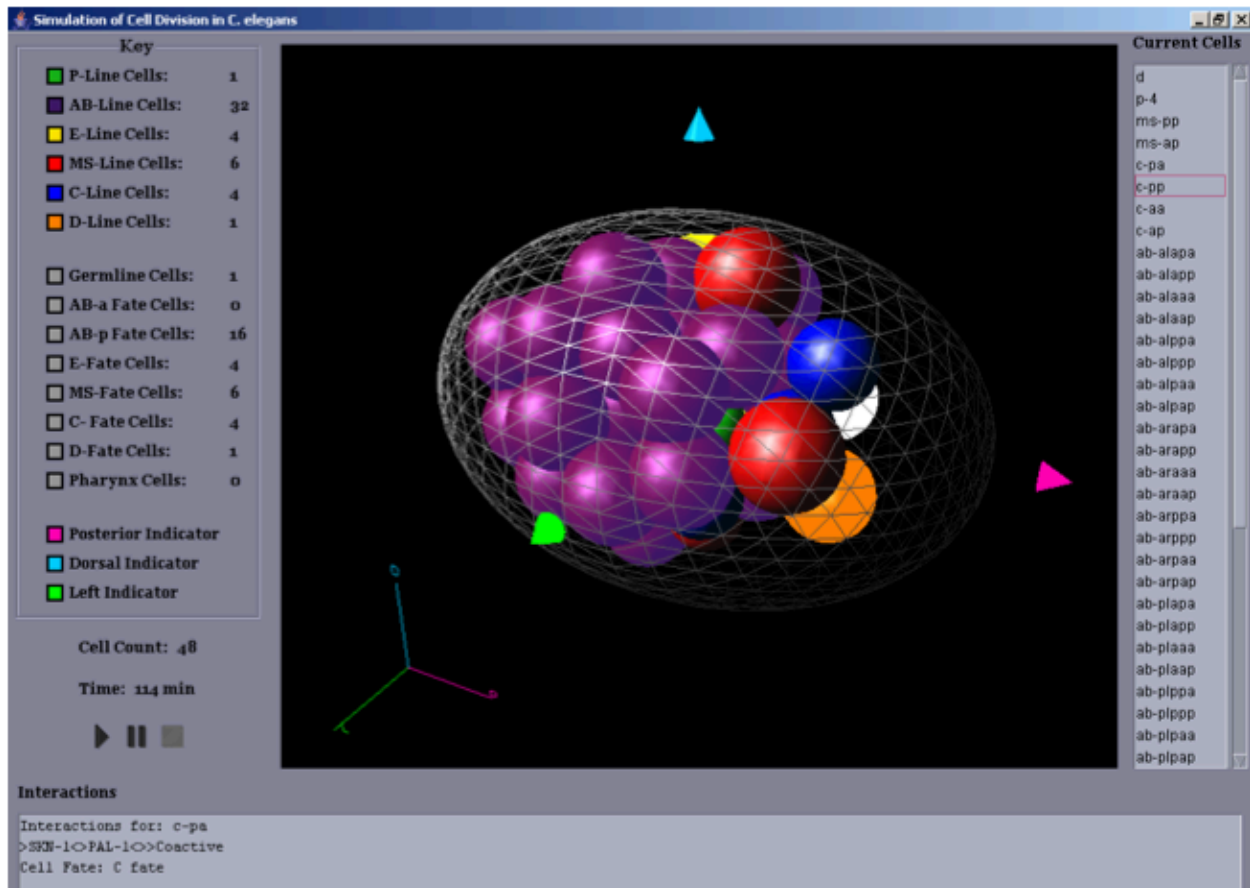


Image taken from (Bogdanova et al., 2004).

Figure 2: Screenshot of previous simulation worm model

The previous incarnation of this simulation progressed through the first 100 cell divisions of *C. elegans* embryogenesis. Cells were represented by simple spheres and kept within a basic eggshell frame. Cells were colored based on the lineages of founder cell, P, AB, E, MS, C, and D. Cell division were based on timing and axis of cell division. Cell interactions were modeled through using a list of internal factors that is passed from parent to daughter cell and added to as cells undergo internal or external signal interactions.

This model, while not a very accurate visual representation of embryogenesis, was able to model the first 100 cell divisions as well as important cell interactions in a non-deterministic fashion.

This model's cell divisions were based on timing cues and the axis of division. Figure 2 shows the simple spheres used to simulate individual cells. The model permitted slight overlaps of cells to simulate the way real cells push against one another, but no two cells were allowed to completely overlap. The cell volumes were determined by the initial volume of the eggshell and the sum of the volume of the all cells remained constant for all cell divisions (Bogdanova et al., 2004).

This version of the simulation included the modeling of both internal and external cellular interactions. Each founder cell was assigned different sets of maternal factors, which are passed on to the daughter cells. As interactions occur, either internal regulatory molecules or external interactions between touching cells, the list of factors is added to or modified in the daughter cells.

The aim of our project was to take the basic concepts of this model, and use the advances in modeling techniques that have been made in the years since this project was completed to make a more accurate and realistic representation of the embryonic development of *C. elegans*.

## **2.2 Embryonic Development of *C. elegans***

The following sections give an overview of the main stages of embryonic development. of *C. elegans*, which was the focus of our model.

### **2.2.1 Proliferation**

The first stage of embryogenesis is proliferation of cells. The entire cell lineage was mapped by John Sulston in 1983, so the origins of all cells are known to come from six key founder cells: AB, MS, E, C, D and P4 (Figure 3). The founder cells are established by asymmetrical cell divisions in the first few divisions (Figure 4). During this phase the embryo changes from a single celled, newly fertilized zygote to a spheroid embryo consisting of 558 undifferentiated cells. This stage occurs from fertilization at minute 0 and lasts approximately 330-350 minutes (~5.5-5.8 hours) (Sulston et al. 1983).

Typically proliferation is viewed in two phases. The first phase takes place inside the uterus of the parent and lasts for approximately 150 minutes, and involves the formation of the zygote to the generation of the founder cells that will go on to produce the main germ layers for the organism. At the end of this phase, the eggs, consisting of approximately 30 cells, are laid outside of the body and gastrulation begins (Wormatlas, 2006).

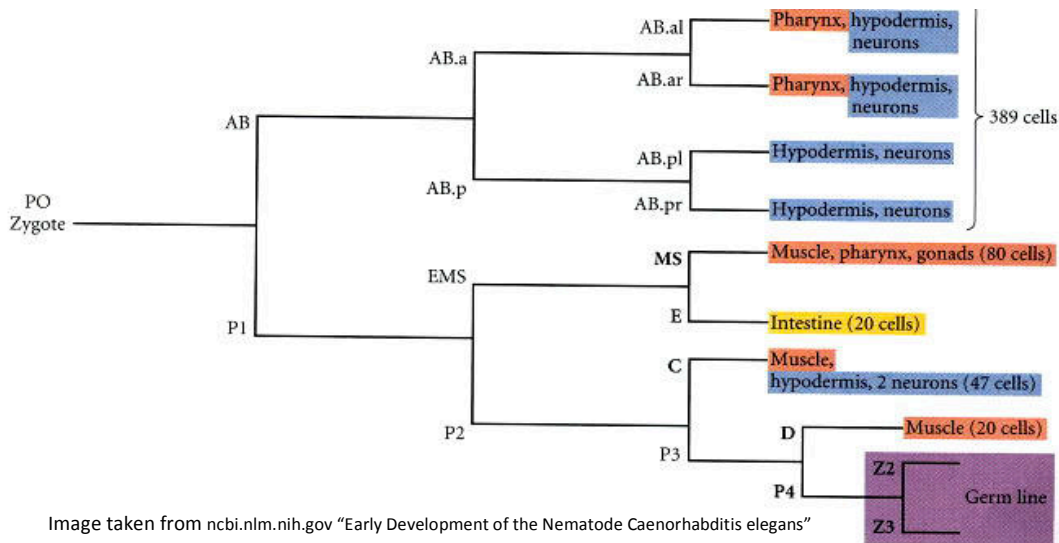


Image taken from ncbi.nlm.nih.gov "Early Development of the Nematode *Caenorhabditis elegans*"

**Figure 3 Founder cell lineage tree**

The *C. elegans* cell lineage has been completely mapped out and every cell is known. The major tissue types can be traced to a few early founder cells. The P cell line gives rise to the germline cells. AB descendants comprise most of the hypodermis, neurons and the pharynx. E cells give rise to the intestines. MS primarily gives rise to muscle cells, but also contributed to the pharynx and gonads. C cells are primarily muscle cells, but some contribute to the hypodermis and 2 neurons. The D line only produces muscle cells. All cell names after the founder cells are based on the axis of division of their parent cell, either anterior/posterior or right/left. AB.p is the posterior daughter cell of AB, while AB.pr is the right daughter of AB.p.



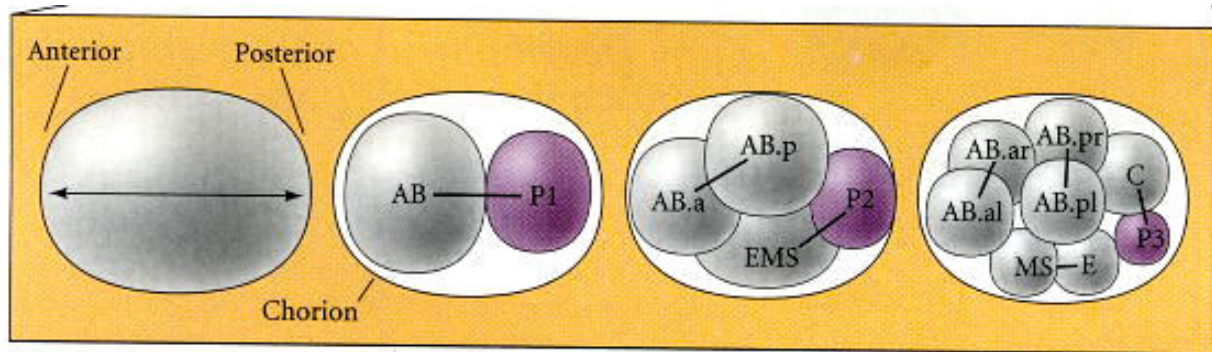


Image taken from ncbi.nlm.nih.gov "Early Development of the Nematode *Caenorhabditis elegans*"

**Figure 4: Asymmetrical split in early cell division**

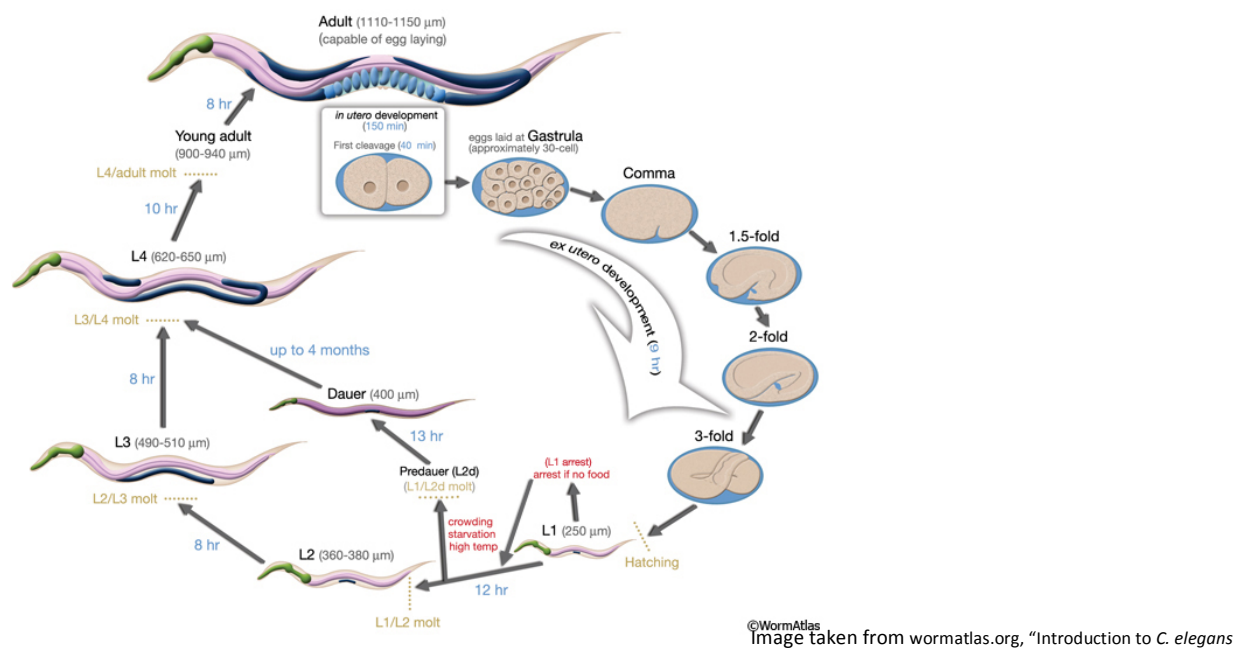
The founder cell lines are generated in the first few cell division of embryogenesis. AB is the first major founder cell to appear and only produces cells from that line. The P cell line only gives rise to germ cells, so it is much smaller than its sister-cells in each subsequent division, which gives rise to a new founder cell. P1 splits in to P2 which is much smaller than its sister EMS which give rise to the two founder lines E and MS. P2 will split into P3 and the C founder cell.

The bulk of cell division occurs during the second phase of proliferation, when the embryo grows from a 30-cell organism to 558 cells from approximately 140-350 minutes post-fertilization. Gastrulation also occurs during this period. During gastrulation, specific cell movements dictate where the precursors to the different germ layers are arranged in the embryo. The endoderm germ layer, which contains the precursor cells to the germline and intestine, migrate into a hollow space in the interior of the embryo known as the blastocoel. The mesoderm germ layer, which gives rise to the pharynx and muscles, follows the endoderm from the ventral surface, leaving the ectoderm, which generates the hypodermis and nerve cells, exposed (Chin-Sang & Chisholm, 2001). At the end of gastrulation, the hypodermal cells of the ectoderm migrate to close the cleft through which the other cells entered the interior to complete the epithelial formation.

**2.2.2 Organogenesis and Morphogenesis**

Organogenesis and morphogenesis takes the embryo from the undifferentiated state at the end of proliferation, as the worm leaves the comma stage, through to the hatching of a fully differentiated worm in the L1 larval phase (Figure 5). This stage lasts from approximately 350-

840 minutes post fertilization and is the end of embryogenesis. This stage occurs without any additional cell division, so the hatched larval worm consists of 558 cells. During this phase, the embryo grows by elongating due to circumferential contraction within the hypodermis. The contraction causes the worm to become three times thinner and approximately four times as long. During elongation, the embryo goes through several stages of development as the embryo begins to fold in on itself to elongate and formulate the worm (WormAtlas, 2006).



**Figure 5: Life cycle of *C. elegans* showing different stages of embryogenesis and larval development to the point of adulthood** The first stages of development occur within the adult worm. Unfertilized cells, or oocytes, are fertilized within the hermaphrodite and go through the first few cell divisions while still inside the adult. The worm lays the egg when is approximately at the 30 cell stage and development of the embryo continues outside of the adult worm. The embryo goes through a series of stages during which it elongates and develops from a cluster of undifferentiated cells into a folded worm larva, at which point it hatches. The worm's development continues after hatching and it goes a four different larval stages until adulthood. *C. elegans* also has a dauer state, in which the worm can remain dormant for up to four months under condition not suitable for normal development.

In Figure 5, the elongation phases are represented from the comma stage to the three-fold (or “pretzel”) stage. As the worm elongates, it folds in on itself to be able to fit inside the egg. At the end of the three-fold stage the main body plan of the embryo that will give rise to the larval and adult stages is fully developed and the L1 larva hatches from the egg.

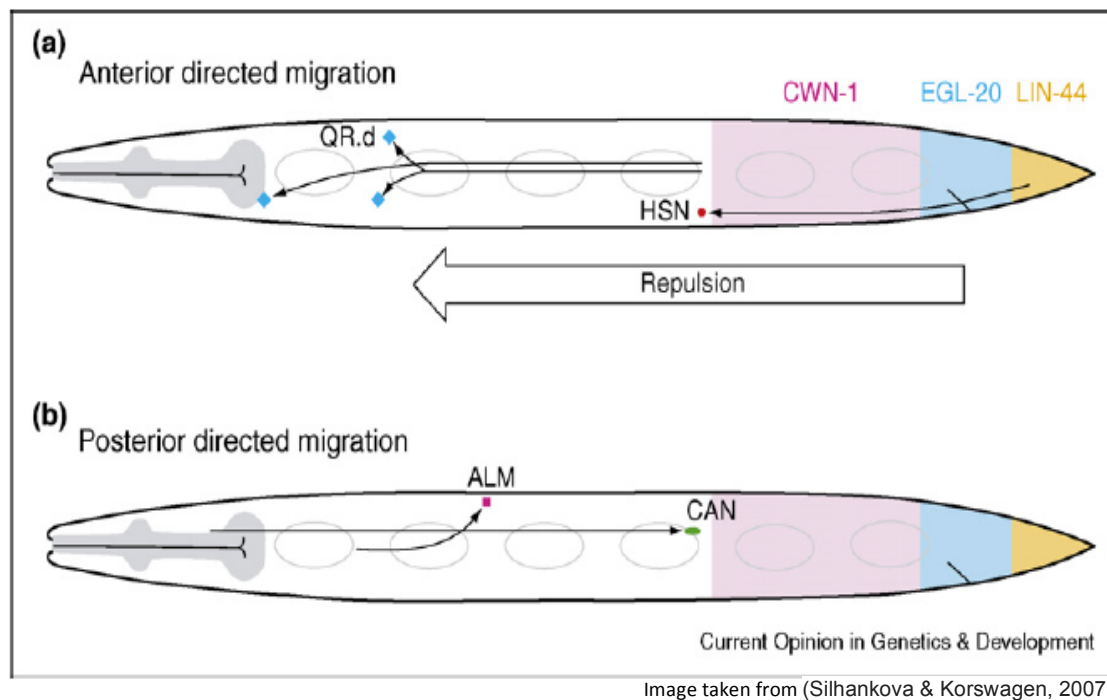
### 2.2.3 Wnt signaling, *mig-10*, and neuronal development

During the development of *C. elegans* certain neurons are known to have significant migrations along the anteroposterior (AP) axis prior to reaching their final location. Out of the 302 neurons found in *C. elegans*, four types of neurons, ALM, CAN and HSN, as well as the Q neuroblast, are known to migrate extensively (Sulston et al., 1983). Both the ALM and CAN neurons begin in the anterior of the body and migrate towards the posterior end. The two Q neuroblasts, one on the left and one on the right lateral sides, begin in the midbody of the worm and migrate in opposite directions: QL migrating posteriorly, while QR migrated anteriorly. HSN begins in the posterior and migrates anteriorly. (Silhankova & Korswagen, 2007).

Wnt proteins are a family of signal protein that play a major role in cell-cell interactions during embryogenesis. Wnts play a particularly important role in directing neuronal migration along the anterior/posterior axis. There are five known wnt genes in *C. elegans*, *lin-44*, *egl-20*, *cwn-1*, *cwn-2*, *mom-2*, and six wnt receptors, four Frizzelds receptors, one ryk receptor and one ror receptor (Zinovyeva, Yamamoto, Sawa, Forrester, 2008). While there is a certain amount of overlap in the expression pattern of these five genes throughout the worm, the differences in the extent of expression along the AP axis as well as the cell type that is expressing the gene are a key area of study for wnt proteins. Wnts interact with certain secreted Frizzled-related proteins (SFRPs), which act as inhibitors to the wnt proteins, and help to form the AP patterning of neuroblasts. (Harterink, Kim, Middlekoop, Doan, van Oudenaarden & Korsawagen, 2011).

Three of the wnt genes, *lin-44*, *egl-20*, and *cwn-1* are localized to the posterior portion of the worm (Harterink et al., 2011). The primary areas of expression for *lin-44*, *egl-20*, and *cwn-1* can be seen in Figure 6 in yellow, blue and pink respectively. The remaining wnt genes, *mom-2* and

*cwn-2*, are expressed along the AP axis. *cwn-2* is more prominently expressed in head neurons as well as muscle and intestinal cells along the anterior portion of the worm, while *mom-2* is expressed in the germ cell precursors (Harterink et al., 2011). The expression of these genes begins early in development, with the patterning being developed by the comma stage of embryogenesis. Transcripts of the *mom-2* gene were found as early as the four-cell stage of the embryo, and the anterior localization of *cwn-2* and the posterior localizations of *cwn-1* were apparent by the 100-cell stage of development (Harterink et al., 2011).



**Figure 6: Anterior and posterior directed neuronal migrations**  
Neurons migrate anteriorly or posteriorly along the axis of the worm. In 6a, the migration of HSN is away from the expression patterns of the wnt genes, *lin-44*, *egl-20* and *cwn-1*. 6b shows the migrations of ALM and CAN towards the wnt expressions.

The AP expression pattern of the wnt signal is affected by the interaction of SFRPs. *Sfrp-1* is a gene encoding for a wnt inhibitor that is expressed primarily in the anterior of the worm.

Mutations in this gene show the final positions of neurons to shift to the anterior of the animal indicating that SFRP-1 interferes with wnt activity stimulating anterior localization (Harterink et

al., 2011). The complexities of the wnt signaling expression patterns and the interactions between the wnts and their inhibitors make them an excellent candidate for modeling in a simulation, as the interactions and expression patterns could be more easily visualized with a three dimensional model. Hypotheses about how cells integrate information from multiple guidance cues could be tested by comparing simulations and experimental data.

The *mig-10* gene is also important in neuronal migration during development. *MIG-10* is a cytoplasmic adaptor protein that transmits guidance cues that function in promoting polarized outgrowth of axons (Quinn & Wadsworth, 2008). In the *mig-10 (ct41)* mutant strain, the migrations of the CAN, HSM and ALM neurons are disrupted. The mutation also results in the truncation of the excretory cell and defective axon guidance (Manser & Wood, 1990).

### **2.2.3 Programmed Cell Death**

Cell death is a very well known and vital part of *C. elegans* development. Over the course of *C. elegans* development, a total of 1090 cells are generated, approximately 1/6 or 131 of which end up dying (Sendoel & Hengartner, 2009). Today much of the cell death process is already known including many of the genes necessary for programmed cell death to occur. Programmed cell death has been mapped to determine specifically which of the cells during development are destined to die (Sulston et al., 1983). Part of the cell lineage map can be seen in Figure 7 outlining which of cells are pre-programmed to die during embryogenesis. Cells that are destined for death are designated with an x marking their end on the cell lineage.

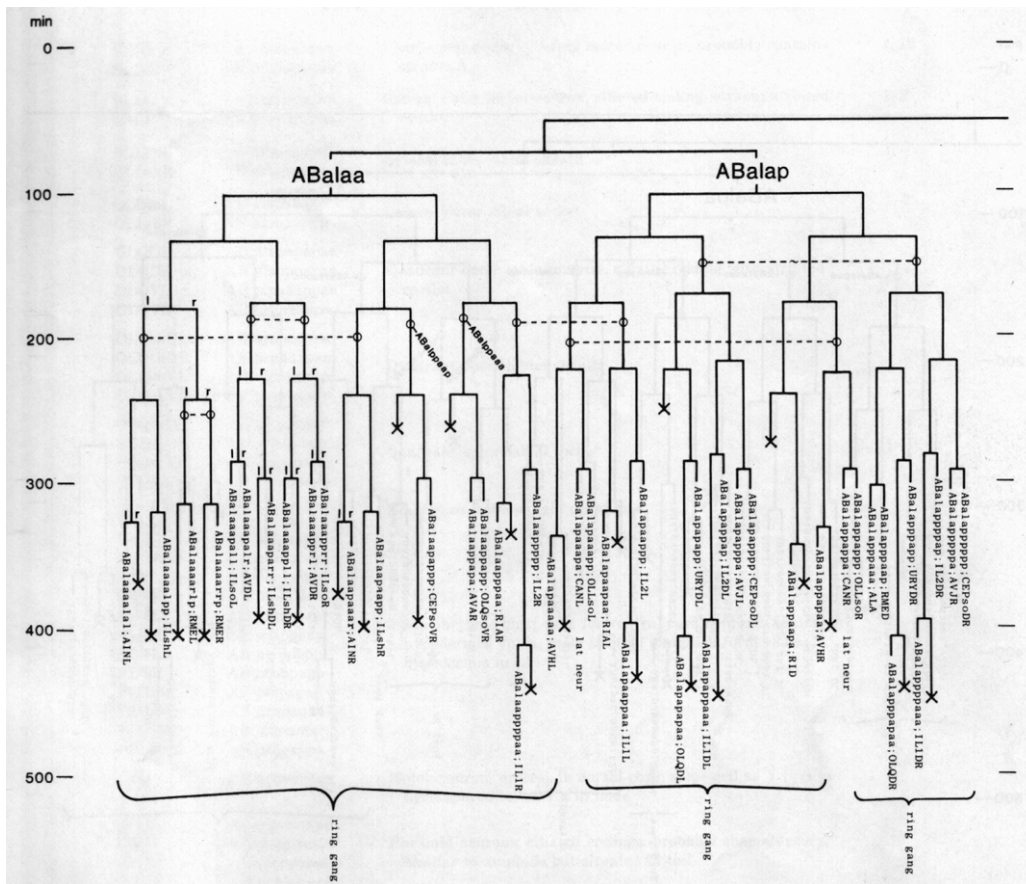


Figure 7: Part of cell lineage illustrating cell death (Sulston et al., 1983)

The entire embryonic lineage of *C. elegans* was mapped out by John Sulston in 1983, where all cell fates were determined and recorded. Individual cells can be followed through the lineage tree to determine their path during embryonic development. Certain cells within the *C. elegans* lineage are known to undergo programmed cell death. John Sulston's lineage is shows which cells undergo this process, marked with an X (Sulston et al., 1983).

Cell death involves three main steps. The first or “specification” phase is when the cells are instructed to undergo programmed cell death through a series of signal pathways. The second or “executions” phase involves the activation of the apoptotic program in the cells, and they are instructed to die. The final or “killing” phase is when the dead cells are broken down and slowly engulfed by neighboring cells. Figure 8 shows cell that are undergoing cell death in a *C. elegans* embryo.

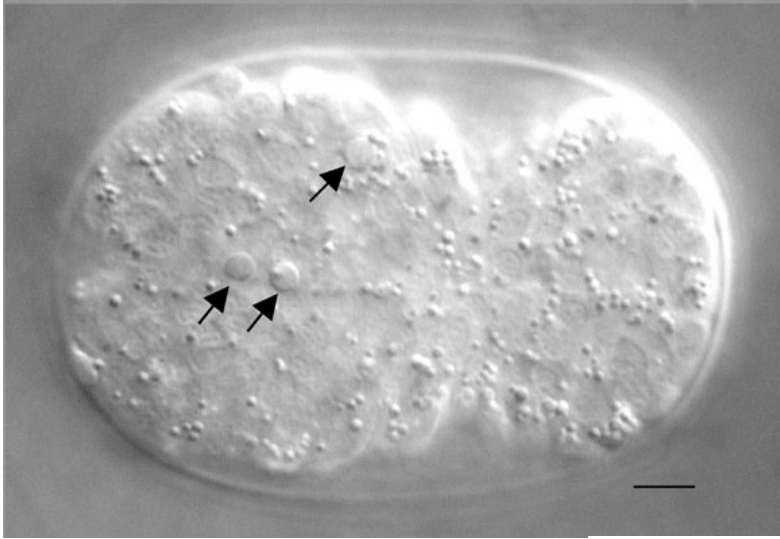


Image taken from (Conradt Xue, 2005)

**Figure 8: Apoptotic cells located in embryo**

As cells undergo apoptosis, their nuclei are broken down as the cells shrink in size becoming small and round until they are engulfed by neighboring cells. These apoptotic cells are marked with arrows.

### 2.3 Inverted selective plane illumination microscopy (iSPIM)

Inverted selective plane illumination microscopy (iSPIM) is a new imaging technique that was developed in response to other imaging techniques being too slow or phototoxic to cells to allow for long term imaging (Wu, Ghitani, Christensen, Santella, Du, Rondeau, Bao, Colón-Ramos, Shroff, 2011). This technique allows the cell lineage of the *C. elegans* embryo to be tracked throughout development, even after the onset of embryo movement, using fluorescent protein markers. Throughout the period of embryogenesis images are taken every two seconds in various cross sections to create a three-dimensional z-value. Automated analysis of these images allows for the measurement of the fluorescent histone markers to track the nuclear positions (Murray & Bao, 2012). The histone fluorescence provides the background scaffold of nuclear positions and allows observation of secondary markers to monitor a particular cell, gene or protein of interest by using a second fluorescent protein that fluoresces on a different wavelength (Murray & Bao,

2012). The nuclei can be semi-automatically annotated using the StarryNite software package (Wu et al., 2011). The package provides x, y, z coordinates of all the nuclei throughout most of development.

## **2.4 Computer simulations**

Computer simulations are used today in a variety of different disciplines and natural systems including physics, biology, chemistry, engineering and even social science. A computer simulation is a computer program that attempts to simulate an abstract model of a specific system of interest. The ability to compute and simulate things that may not be possible due to time or financial restrictions are made possible with computer simulations which is why they are such a valuable tool to research scientists (Simulation Overview, 2013).

### **2.4.1 Simulation uses**

One of the most attractive benefits of developing a computer model is that it can begin with a simple approximation of a process or a sequence of events that can later be refined and modified once the understanding of the progression improves. Computer simulations are very valuable because they can be used when no tractable mathematical model exists for the situation, for example, a simulation that is created to help improve inventory decision-making when the amount of demand from the customers is unknown. Simulations are not only used to understand the current system and why it is performing a certain way, but they can also be used to discover innovative changes that can be made to the system to improve it or even to develop an entirely new approach and solution to a problem (Management Science - Computer Simulation, 2013).



## 2.4.2 Types of simulations

There are two main types of computer simulations that exist, those that detail discrete event systems and those that detail continuous event systems. In a discrete simulation, the time increments are fixed and values change based directly off of advancements in time. In a continuous simulation, the system will change based on the occurrence of new events. The passing of time will have no effect on the model; instead, it is only the presence of a new event that will trigger additional events and allow the simulation to continue. Some simulation incorporates both continuous and event based methods, known as a hybrid simulation (Simulation Overview, 2013).

## 3 Methodology

The following methodology section is an overview of the methods used in the laboratory and the methods used in development of the simulation.

### 3.1 Biological Methodology

#### 3.1.1 Growing and maintaining worms

The various strains used throughout the project, both stock and maintained strains, were maintained on a weekly basis over the year. *C. elegans* grows on Nematode Growth Medium (NGM) agar plates that have been spotted with the mutant *E. coli* strain OP50, which is used as a food source for the worms. Strains were maintained by transferring approximately three to four hermaphrodites in the L4 stage of development, with the exception of the N2M strain. The N2M strain is a male-containing line, which was used for our crosses to insure known genotypes when crossing. In order to maintain this strain, approximately three to four hermaphrodites were mated to approximately four to seven males from that line. Strains currently in use were grown at 20°C, while those not currently in use were grown at 15°C. Strains that had been starved were rescued by chunking to a new plate and allowing them to grow through one or two generations before being used again. Refer to Appendix A for all lab procedures.

#### 3.1.2 Strains

In order to use the iSPIM to track the cell lineages and migrations, we wanted to create a mutant strain of *C. elegans* that contained the necessary fluorescent markers as well as the *mig-10 (ct41)* mutation. The markers necessary for the iSPIM were the two red histone markers, *pie-1 :: H2B ::*

*wcherry* and *his-72 ::HIS-72 :: wcherry*, and the green fluorescent marker *ceh-10 :: GFP*. In addition to these markers, we also used *pgp-12 :: GFP* as a marker that allowed us to select for worms that had the *mig-10 (ct41)* mutation and could be crossed out of the strain once the *mig-10* mutation had been selected for and become homozygous in the strain. All the markers used in our crosses and the cells they affect can be found below in Table 1. The different strains used to cross in the markers and create our mutant strain are listed below in Table 2. The strains RY1220 and RY1221 were generated in our lab using the other four strains from different labs.

Marker	Description
<i>pie-1 :: H2B:: wcherry</i>	Red histone marker for early embryonic cells
<i>His-72 ::HIS-72 wcherry</i>	Red histone marker for larval and adult cells
<i>ceh-10 :: GFP</i>	Green marker for the CAN neuron
<i>pgp-12 :: GFP</i>	Green marker for the excretory cell

Table 1: Fluorescent Marker Index

Strain	Description
N2M	Wild Type mating strains
BV117	<i>ceh-10::GFP; pie-1:: H2B :: wcherry; his-72:: HIS-72 :: wcherry</i>
BW315	<i>mig-10 (ct41)</i>
BC06288	<i>mig-10 (ct41); pgp-12::GFP</i>
RY1220	<i>mig-10 (ct41); ceh-10 :: GFP; pgp-12::GFP</i>
RY1221	<i>mig-10 (ct41); ceh-10 :: GFP; his-72 :: HIS-72 :: wcherry; pgp-12::GFP</i>

Table 2: Worm Strain Index

### 3.1.3 Strain Design

Each new strain design began with N2M males to introduce males into the line. The first cross we attempted was to create the A-term strain containing the *mig-10 (ct41)* mutation; *ceh-10::GFP*; and the *pgp-12::GFP* identifying marker (Equations 1-4). The following crosses were designed and implemented in the same manner and are available in Appendix A.

#### Creating: RY1220 Strain

$$\frac{mig-10 (ct41)}{mig-10 (ct41)} \cdot \frac{ceh-10::GFP}{ceh-10::GFP} \cdot \frac{pgp-12::GFP}{pgp-12::GFP}$$

##### Equation 1: Goal of Cross

The goal of the cross was to create a strain that had that contained the *mig-10* mutation and the two GFP markers.

The first step of the cross was to cross the BV117 hermaphrodites, which is homozygous for the *ceh-10 :: GFP* marker of the CAN neuron, with the N2M males, which have no marker therefore, all of the male offspring can only be heterozygous for the marker (Equation 2).

#### First Step: N2M x BV117

$$\frac{+}{+} \times \frac{ceh-10::GFP}{Ceh-10::GFP} \xrightarrow{yields} \frac{+}{ceh-10::GFP}$$

##### Equation 2: Cross 1

The first step crossed N2M males with BV117 females to produce a worm strain that was heterozygous for the *ceh-10* marker.

The next step was to cross the male progeny from the first cross to BW315 hermaphrodites to introduce the *mig-10 (ct41)*. In this cross, all the male offspring will be heterozygous for *mig-10 (ct41)*; however they could either be heterozygous for the *ceh-10::GFP* marker or homozygous without in (Equation 3). In order to pick the males with the *ceh-10::GFP* marker, the worms were observed under a fluorescent microscope where the *ceh-10 ::GFP* marker was visible, though fainter than it would be in a homozygous worm.

## Second Step: First Cross progeny X BW315

$$\frac{+}{+} \cdot \frac{+}{Ceh-10::GFP} \times \frac{Mig-10 (ct41)}{Mig-10 (ct41)} \cdot \frac{+}{+} \xrightarrow{\text{yields}}$$

$$\frac{+}{Mig-10 (ct41)} \cdot \frac{+}{Ceh-10::GFP} \text{ (and } \frac{+}{Mig-10 (ct41)} \cdot \frac{+}{+} \text{)}$$

### Equation 3: Cross 2

The male progeny from Cross 1 were crossed with the BW 315 hermaphrodites to introduce the *mig-10 (ct41)* mutation. All the cross progeny must be heterozygous for *mig-10*, however only half of the offspring will be heterozygous for the *ceh-10::GFP* marker and the other half will not have the marker at all. In order to pick offspring with the marker, worms were observed under a fluorescent microscope and selected for expression of GFP fluorescence.

The male progeny from cross 2 that were selected with the *ceh-10::GFP* marker were then crossed with the *pgp-12 ::GFP* hermaphrodites to try and homozygous the *mig-10 (ct41)* mutation as well as to introduce the *pgp-12::GFP* marker which allows to select for the homozygous mutants (Cross 3). The *pgp-12 ::GFP* marker causes the excretory cell to fluoresce and when the *mig-10 (ct41)* mutation is homozygous, the excretory cell is visibly truncated. Only worms with the truncated excretory cell and that expressed the *ceh-10::GFP* marker were selected for the next cross.

## Third Step: heterozygous progeny x BC06288

$$\frac{+}{Mig-10 (ct41)} \cdot \frac{+}{Ceh-10::GFP} \cdot \frac{+}{+} \times \frac{Mig-10 (ct41)}{Mig-10 (ct41)} \cdot \frac{+}{+} \cdot \frac{pgp-12::GFP}{pgp-12::GFP} \xrightarrow{\text{yields}}$$

$$\frac{Mig-10 (ct41)}{Mig-10 (ct41)} \cdot \frac{+}{Ceh-10::GFP} \cdot \frac{+}{pgp-12::GFP}$$

### Equation 4: Cross 3

The male progeny from cross 2 that displayed the *ceh-10::GFP* marker were crossed into the BC06288 strain. This cross allowed for the *mig-10 (ct41)* mutation to become homozygous as well as introducing the *pgp-12::GFP* which allows for the selection of *mig-10 (ct41)* homozygous mutants by fluorescing the excretory cell which becomes visibly truncated in the mutant. The selected offspring from this cross displayed both the truncated excretory cell and the *ceh-10::GFP* marker.

The final stage in the cross was to select the desired offspring from the previous step and allow them to self fertilize until they produced offspring that were homozygous for each marker. This was achieved by picking individual hermaphrodites to single plates and allowing them to self cross. The progeny were then observed and selected for ones displaying all the markers. The

process is continued until all progeny from a singled hermaphrodite display all the markers and the markers are known to be homozygous.

## Fourth Step: Single Cross 3 progeny until homozygous

$$\frac{Mig-10(ct41)}{Mig-10(ct41)} \cdot \frac{+}{Ceh-10::GFP} \cdot \frac{+}{pgp-12::GFP} \times \frac{Mig-10(ct41)}{Mig-10(ct41)} \cdot \frac{+}{Ceh-10::GFP} \cdot \frac{+}{pgp-12::GFP} \xrightarrow{\text{yields}}$$

$$\frac{Mig-10(ct41)}{Mig-10(ct41)} \cdot \frac{ceh-10::GFP}{Ceh-10::GFP} \cdot \frac{pgp-12::GFP}{pgp-12::GFP}$$

### Equation 5: Cross 4

Hermaphrodites from cross 3 that have both the truncated excretory cell and the *ceh-10::GFP* marker were picked to individual plates and allowed to self fertilize. The offspring were singled and self crossed until all markers were known to be homozygous.

## 3.2 Simulation Methodology

This section outlines the computer science methodology and approach to the simulation. The various programming languages and algorithms used to develop the model are outlined in the following sections.

### 3.2.1 Programming Languages and Libraries

The simulation was built for the web, which brought with it a great advantage in ease of access. HTML5 is the programming language the web is built on, almost every web page renders a HTML dom. HTML5 has a few great updates; most important to simulation is the canvas element. This allows the page to rapidly draw and redraw to the screen, which is needed to produce any kind of real time graphical user interface (GUI). To get 3D drawing an OpenGL library is needed.

WebGL is a toolkit specifically designed for HTML5 allowing access to the powerful OpenGL bindings that make rapid 3D rendering possible. These bindings are used to accelerate the arduous linear algebra needed to collapse a 3D model to a 2D screen through a computer's Graphical Processing Unit (GPU). On top of the WebGL toolkit Three.js was used. This application-programming interface (API) takes the WebGL toolkit and gives access to powerful generalized methods like setting up a scene or shaders. Three.js is a JavaScript API and because JavaScript and HTML5 go hand in hand naturally the bulk of the simulation was written in JavaScript.

JavaScript is the behind the scenes workhorse that makes the web 2.0 possible; although it may not be the most elegant of languages it serves its function quite nicely. But the web site needs to be hosted somewhere for that, a Ruby on Rails app was implemented. This rails app is heroku.com, a site which specializes in hosting rails apps. All Rails apps need a Database to run as suggested by Heroku, PostgreSQL is used here. The rails app part of SIM worm manages all server requests and handles the database. All together this simulation is written in three different languages utilizing a number of external Libraries. The full list of Libraries and Languages can be found in appendix E.

### **3.2.2 Algorithms**

In order to render a three-dimensional representation of the *C. elegans* a number of algorithms are employed. There are two main steps in getting the image to the computer screen; first, creating a three dimensional description, and second, collapsing that 3D description into a 2D one. The latter involves a lot of linear algebra and is handled entirely by OpenGL bindings. The first part however requires a number of difficult tasks on the part of this model. It has to figure out where the nucleus of each cell is and where the edge of the cell is and how much they push

on other cell membranes; this is done with a Metaball algorithm. Then a polygon mesh (a series of triangles) has to be created so that the OpenGL bindings can render the frame; this is done with a marching cube algorithm.

### *3.2.2.1 Metaballs*

Metaballs are a mathematical representation of an object in N-dimensional space (Blinn 1982). For the purposes of our simulation our Metaballs are in three dimensions x,y, and z. Each Metaball has three major components, a center, density function and threshold. Metaballs are great for modeling soft body deformation because although only a small amount of data is stored very complex structures can take form through Metaballs interaction with one another.

A Metaball calculates its surface by computing  $\text{density}(x,y,z) > \text{threshold}$  (Blinn, 1982) . If the density function given a point x,y,z is greater than the threshold then that point is inside the object otherwise it is outside. A density function takes in a point x,y,z and turns that into a scalar value, the distance from the center which can then be put through a function (a Gaussian function was used for this simulation) to normalize the value. If this value is above the metaball's threshold then that point is inside the object; otherwise, it is outside.



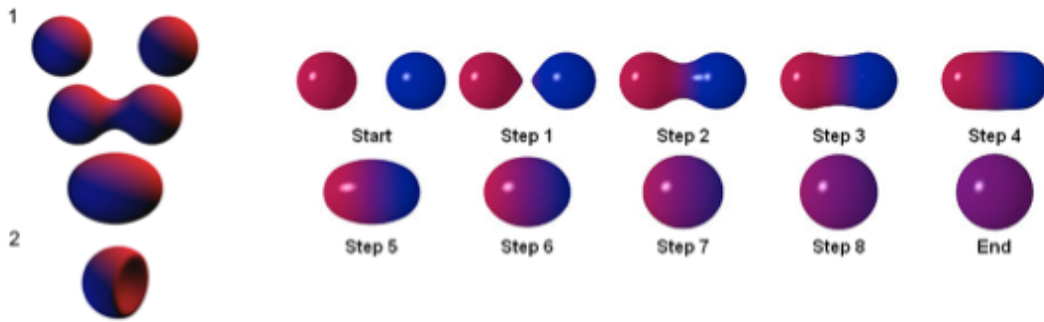


Figure 9: Visual representation of Metaball algorithm

Metaballs are computer algorithms used to create organic looking three-dimensional objects. The three main parts of a metaballs include the center, the density function and the threshold. These algorithms can use positive and negative indentations of each other to create unique shape changes. Multiple metaballs are often combined or split apart to model soft body deformations.

When two or more Metaballs interact there are a number of ways this can happen, primarily they combine or subtract from one another. When two Metaballs are part of the same shape the density function values can be added or the larger of the two can be used to test if it is larger than the threshold; any function that does not shrink the combined density function value will work (Figure 9). In the case of separate shapes, interacting and drawing the edge where density functions overlap will produce larger shapes but any combination of density function values that shrink the combined value will result in smaller shapes (Blinn, 1982).

### 3.2.2.2 Marching Cubes

The marching cube algorithm allows a Metaball to be turned into a polygon mesh approximation. Polygon meshes are the mathematical representation of a 3D object that WebGL expects in order for it to render (Lorenson & Cline 1987). A scene is first divided in a number of cubes or Voxels. By taking a cube, which has eight corners each of which can either, be inside or outside a Metaball there are 256 possible states for the cube to be in. The idea being that a number of triangles should cover the corners that are inside and leave exposed those that aren't.

There are fifteen unique shapes that will cover all possibilities (Figure 10). Because two adjacent cubes share four corners when their triangles are calculated they fit together like a puzzle piece. The smaller each cube the more detail is provided and the smoother the object becomes (Lorensen & Cline 1987).

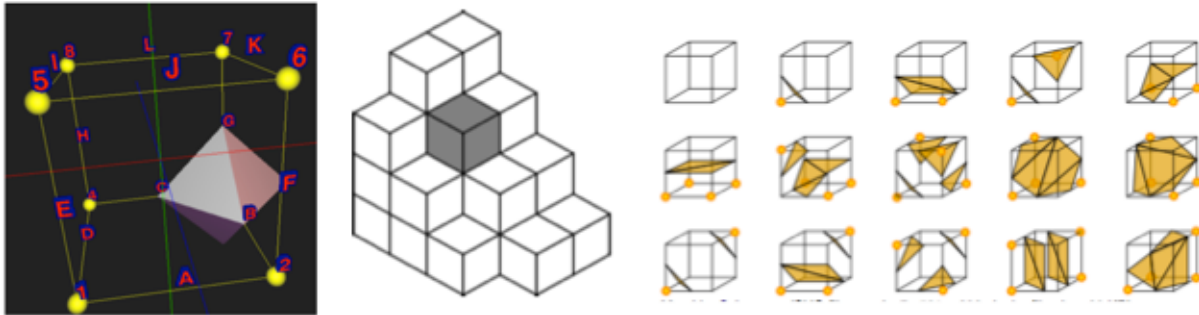


Figure 10: Visual representations of Marching Cubes  
Marching cubes was an algorithm developed by Blinn that allowed metaballs to be turned into a polygon mesh approximations. Marching cubes allow a larger object to be broken up into smaller parts so surfaces can be formed using polygon shapes. The combination of additional marching cubes allows for sharper detail of cell shapes.

## 4 Simulation Design

Background research was conducted in order to develop a better understanding, not only of the developmental aspects of *C. elegans*, also what could be accomplished in the simulation. We first needed to decide what exactly we wanted to see in the simulation and how we would go about portraying this in the simulation in an accurate way. It was important for us to decide which factors and details were important for the biological process to take place but not necessarily needed in the simulation. Early in design development another important thing taken into consideration was the controls and tools that would be available for the user to operate the simulation.

### 4.1 Design

The design of the simulation neatly breaks down into four sections; Data structure, Simulation, Rendering, and User Interactions. Data structure is how the information about cells is organized such as their lineage, shape positions and movements. The simulation is how this data changes with time, which is used to build a structure that can be rendered. Rendering is the process of getting the simulation to draw on the screen. User interactions are what the user can do to control the simulation and its rendering process.

#### 4.1.1 Data Structure

At the heart of the simulation is a Cell object. Cells have a number of properties for rendering purposes such as a name, tissue type, size and location and a few for simulation purposes: daughter cells, chemical levels and active genes. Figure 11 gives an overview of the structure of

the components of the cell object. The gene object is not necessarily analogous to the genes in the actual *C. elegans* and only a small set of phenotypes are implemented. A gene object has identifier an identifier property and zero or more signal objects and zero or more trigger objects. Signal objects represent the signaling capability of cells in biology; in this simulation they have a chemical name, and are received or sent at a certain rate. The trigger object has a listener(s) and event(s): this produces an, if-then like logic. There are two main types of listeners, timed and signal. The time listener trips if the simulation time is over or below a given time value. A signal listener fires when a specified chemical level is above or below a given signal value. The events that a trigger can fire are, migrations, flip genes, divisions and deaths. The Migration event tells the cell to move in one of three ways, absolute, relative, or along a signal gradient. The flip gene event activates or deactivates a gene in the cell. The division event starts the cell on dividing into its two children. The death event tells the cell to start dying. Each event has duration to complete between instant and indefinite.

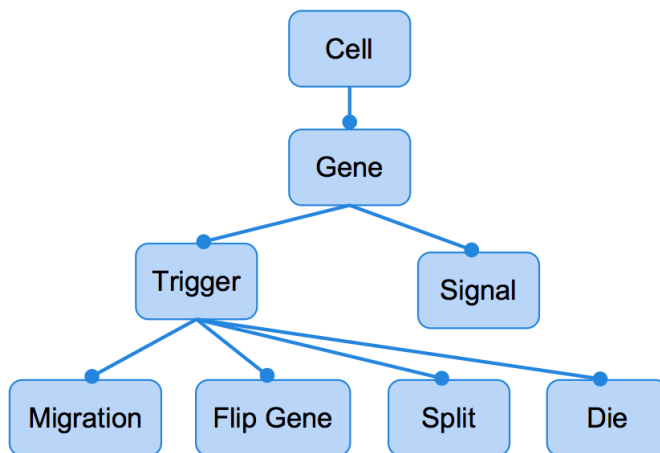


Figure 11: Approach to data structure

A general outline of the data structure used in the development of the simulation. The cell object contains many genes and the gene object contains the triggers and signals. The signal object represents a cell's ability to send and receive chemical signals. The trigger objects then listens for events to happen such as time passing or a certain value or a signal level passing a threshold. Once either of these happens, the events that belong to the trigger object will fire. Migration events will create movements within the cell, flip gene events are responsible for turning on and off a gene. Split events are responsible for cell divisions and die events will induce cell death.

### 4.1.2 Simulation

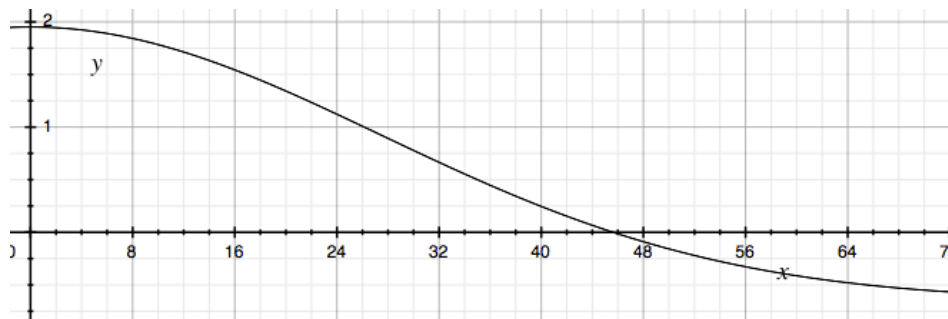
Our simulation is a continuous simulation built from a recursive function that takes a cell and the time the simulation is supposed to stop. Once the data is in a format that the Simulation can handle the information needs to be processed in such a way that it can be rendered. The recursive function pushes cells onto a stack of currently live cells. A recursive function essentially traverses the lineage tree showing which cells are currently alive. Each time the push cell function is called it figures out what events have fired and performs them. A recursive function is a function that calls itself, so if the cell is still alive at the end of these events, it is added to the currently alive stack; other wise the push cell function is called on the children cells. Once this function returns there is a list of currently alive cells each of which have information like their position and size; this is used to render the embryo.

### 4.1.3 Rendering

The rendering phase starts by slicing the scene into a number of cubes or Voxels. The simulation then loops through each voxel calculating whether or not the edge of the cell is present. The total runtime is  $O(x*y*z*n^2)$  To do this at each cube it must run through each cell twice asking it to compute a density function for that point in space, with the outside loop being the drawn cell and the inside simulating the cells pushing up against each other. This is done with a density

function of  $f(r) = \frac{520}{\text{number of cells}} * e^{-\frac{0.002}{\text{volume}}*r^2} - \frac{520}{4*(\text{number of cells})}$  for each density function for

each Metaball (Figure 12).



**Figure 12: Density function**

The graph displays the density function of the metaballs for a cell. On the horizontal axis is the distance from the nuclei; the cell edge is where the function crosses zero.

Where  $r$  is the distance from the given point to the center of the Metaball. For each cube every cell is looped through twice once for the cell that is being drawn and once for the cells that are acting on it. The volume preservation and soft body collision works by forcing an edge when the largest density value for all other functions is larger than the combined value. The combined value is the original density function's value minus a percentage of the value of all other cells density functions that are below the threshold of zero. The other edge of the cell is where the combined value is zero. This prevents two cells that are pushing up against each other from overlapping and extends the cell membrane on the other side, creating the appearance of volume preservation.

## 4.2 User Interaction and User Interface

There are three main ways the user can interact with the simulation, navigation, clipping and coloring. A timeline also allows the users to navigate to a specific time in development. The Navigation tab allows the user control over the camera and model rotation. The clipping feature allows the user to slice away sections of the embryo to get a better look at the inside. The coloring tab allows the user to customize the color of cells in the simulation.

## 4.2.1 Navigation

The simulation allows complete mobility of the camera, allowing movement in the x, y and z position. Using point and click, the camera can be rotated in any direction to view the developing embryo at any angle. The user is also given the ability to zoom in or out the window and initiate autorotation of the embryo in the x,y or z axis. Navigation controls can be seen Figure 13.

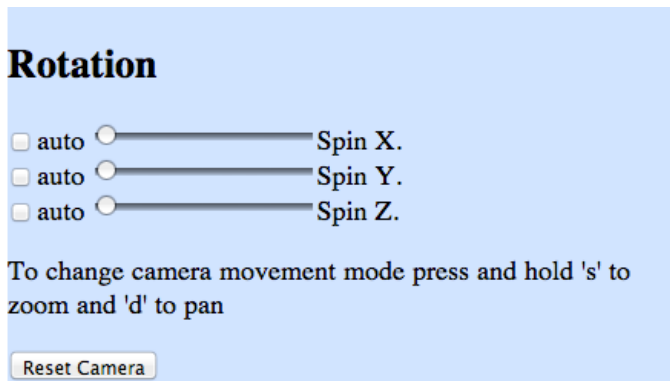


Figure 13: Screenshot of navigation controls

Full control of the camera was provided in the x, y and z positions. Click and drag navigation gives the user full mobility of the embryo with the option to auto rotate in the desired direction. The camera can be zoomed in closer to view specific cells in the embryo or zoomed out for a wider view of the entire embryo.

## 4.2.2 Clipping

Clipping allows the user to take cross sections of the embryo and see the interior of the egg.

Sections of the embryo are cut away from opposite sides of the embryo leaving a thin slice of the embryo allowing interior views of the embryo. This is achieved by limiting the number of voxels in the scene without changing the size of each voxel. A user can slice from either side of any of the three axis. A screenshot of the clipping control user interface is available in Figure 14.

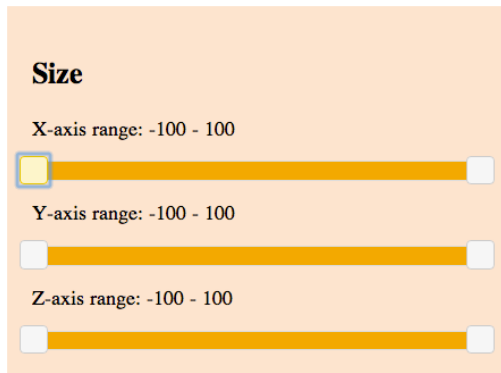


Figure 14: Screenshot of clipping controls

Cross sections of the embryo can be taken by using the clipping features provided. By cutting away sections of embryo from one or either side of the embryo the user is able to view the shape of cells in the interior of the embryo. Cross sections can be taken along any axis, can be one sided, or a slice from both side as thick or thin as the user chooses.

Figure 15 shows screenshots of the clipping abilities and screenshots of cross sections taken in both versions of the three dimensional model. Version One of the three dimensional simulation is shown in Panes 1, 2 and 3 which do not show lighting and in panel 4 is a cross section of Version 2 with lighting implemented.

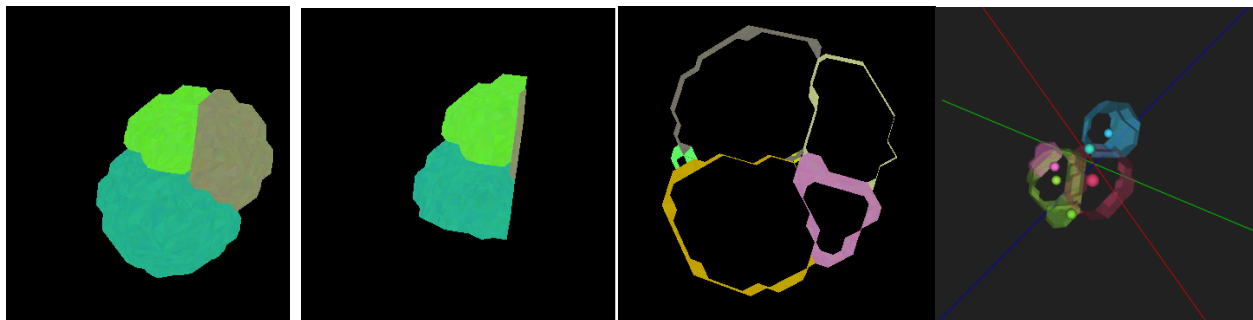


Figure 15: Clipping tool in action

The first panel shows the entire model without any clipping. The remaining panes are examples of different options the user has in viewing section of the model. The first three panels are from Version One of our three dimensional model, where the last panel shows a cross sections from an updated version of the simulation with the nuclei positions and x,y, z axes visible to the viewer as points of reference.

### 4.2.3 Coloring

As cells divide in the simulation, different colors are assigned to new cells as new divisions occur so there is no repetition in coloring. The user is given the ability to color cells as they wish (Figure 16). Specific cells can be colored by searching for them in the drop down cell name



menu. Additionally the user can choose to color a specific cell type a certain color if the user would like to focus specifically on one type of cell. Coloring of the cells allows better differentiation between cell types and allows the user to make color changes on the fly. Built into the cell coloring section of the simulation is an option for hiding cells. The user is given the ability to hide certain cells in the simulation at any time. If the user would like to exclude a certain cell from view, checking the hide check box will allow the user to focus on only the desired cells. The user is also able to choose to draw the nuclei of the cell. This option is not affected by whether or not the cell membranes are drawn. The coloring scheme of any given cell can be cascaded down to all of its children in the lineage trees, which allows the user to easily identify the offspring of precursor or founder cells.

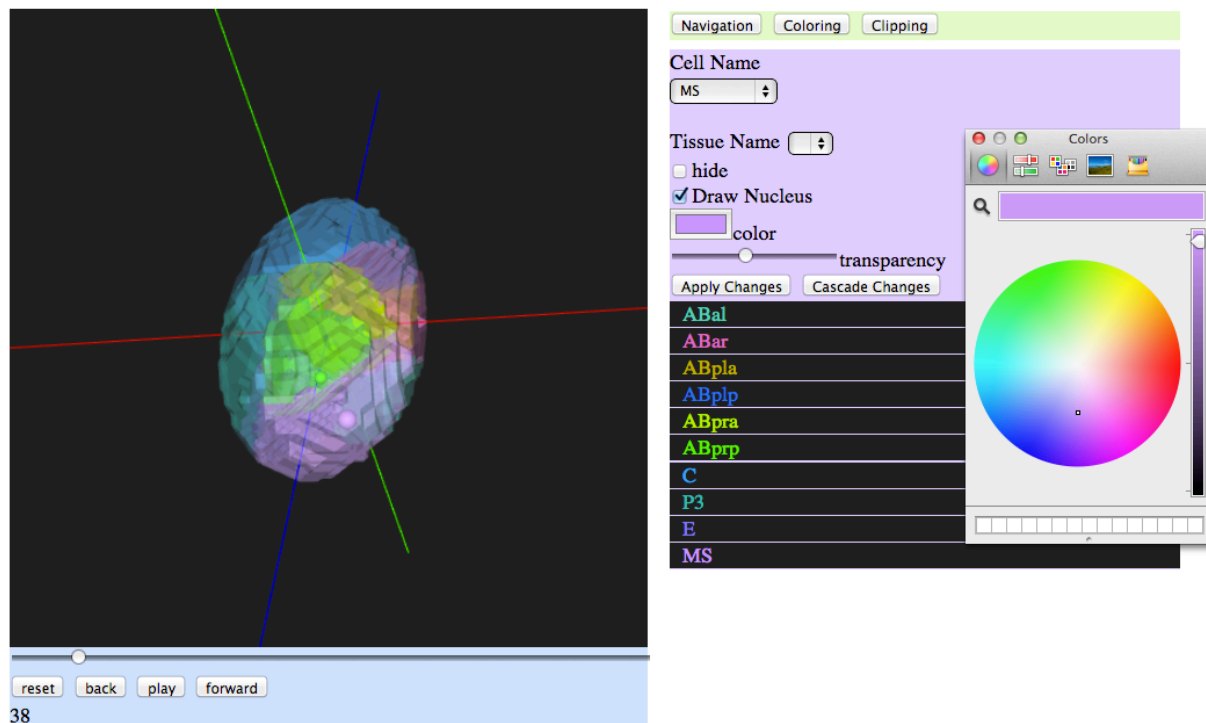


Figure 16: Cell coloring feature

Each new cell is assigned a new color by default; however, the model allows for the user to select any color for any individual cell by selecting the cell name from the list in black on the right then selecting the desired color using the color palate. The user is also able to cascade color changes so that all the descendents of the selected cell will have the same color, allowing the cell lineage to be tracked throughout development.

#### 4.2.4 Programmed cell death:

In the simulation cells were modeled after the real development of *C. elegans* where cells follow a normal course of cell death. During the development of *C. elegans* the organism undergoes programmed cell death that is necessary for the completion of the embryo development. Due to the importance of cell death in development we needed a way to incorporate this into the model.

In the model programmed cell death works to model actual cell death where neighboring cells engulf the apoptotic cells (Conradt & Xue, 2005). In the simulation, the nucleus appears to move over to the center of the neighboring cell allowing it to ingest the neighboring dead cell. The dying cell will begin to shrink leaving a smaller cell surrounded by various neighboring cells. Slowly a neighboring cell will begin to engulf and take over the smaller cell, eliminating the presence of the apoptotic cell in the embryo. The simulation still receives the nuclear coordinates of dying cells and therefore considers them “alive” as their volume shrinks to nothing. Once there is no longer data for the co-ordinate the cell no longer exists in the simulation. The cell engulfing the dead cell expands as the dying cell shrinks, showing that the apoptotic cell is being engulfed.

#### 4.2.5 Photo Gallery

A photo gallery is available to the user on the website where the simulation is hosted (Figure 17). The photo gallery displays various different images of *C. elegans* embryonic development, cell positions and movements, morphogenesis and timelines of embryo development. These images are available for the user to learn more about the *C. elegans* development process through pictures and to be able to compare the simulation with real life images and figures.

- general pictures
  - cell positions and movements
  - excretory cell
  - general structures
  - f1 larva general
  - late.pharynx
  - pharynx
  - sorting by cell fate
- muscle
  - muscle cross section
  - somatic muscle development
- neurons
  - brain and neurons worm
  - embryo neuron development
  - embryo neurons
  - neurons larva
  - neurons
- overview of development
  - early neuron development
  - early up to 28 cells
  - morphogenesis
  - stages 2
  - stages 3
  - stages
  - time line
- ventral closing
  - Embryo epidermis ventral enclosure
  - epithelial enclosure
  - epithelial morphogenesis
  - ventral closing
  - ventral enclosure
- cell ingresson
- embryo/epidermis and gut
- embryonic elongation

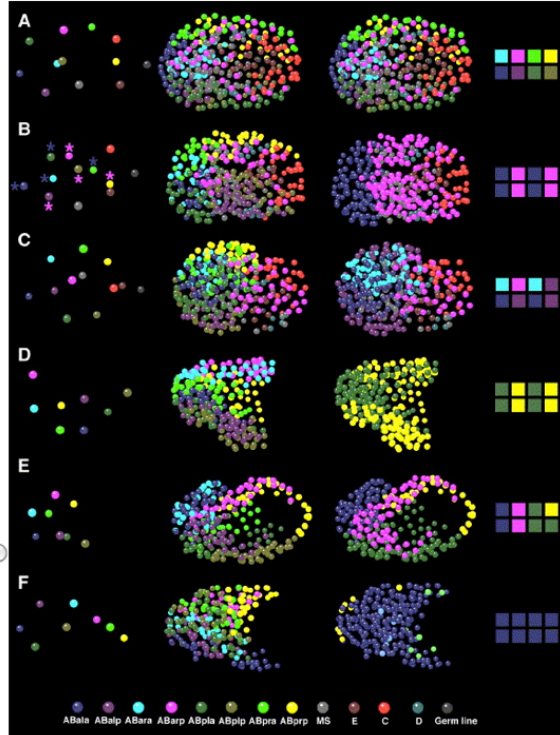


Figure 17: Photo gallery as seen on [www.simworm.herokuapp.com](http://www.simworm.herokuapp.com)

The photo gallery on the website contains various images of different stages of *C. elegans* development. The images can be used to learn more about the development of *C. elegans* as well as to provide users a quick comparison of the model to figures from *C. elegans* research and real life images of developing embryos.

### 4.3 Online Capability

One of the main goals of our project was to be able to use this simulation as a tool for research and education. Hosting the model online is something we decided to do from early on in the simulation that would allow us to address these two goals even better. The idea was for researchers to be able access the model from wherever they may be regardless if they have the simulation saved to their computer. The ability to access it on the Internet allows the simulation to reach a greater audience of biologists that can share it with other fellow biologists or educators and access it to provide feedback for future improvements. Figure 18 is a view of the current website for our simulation running on the Internet.

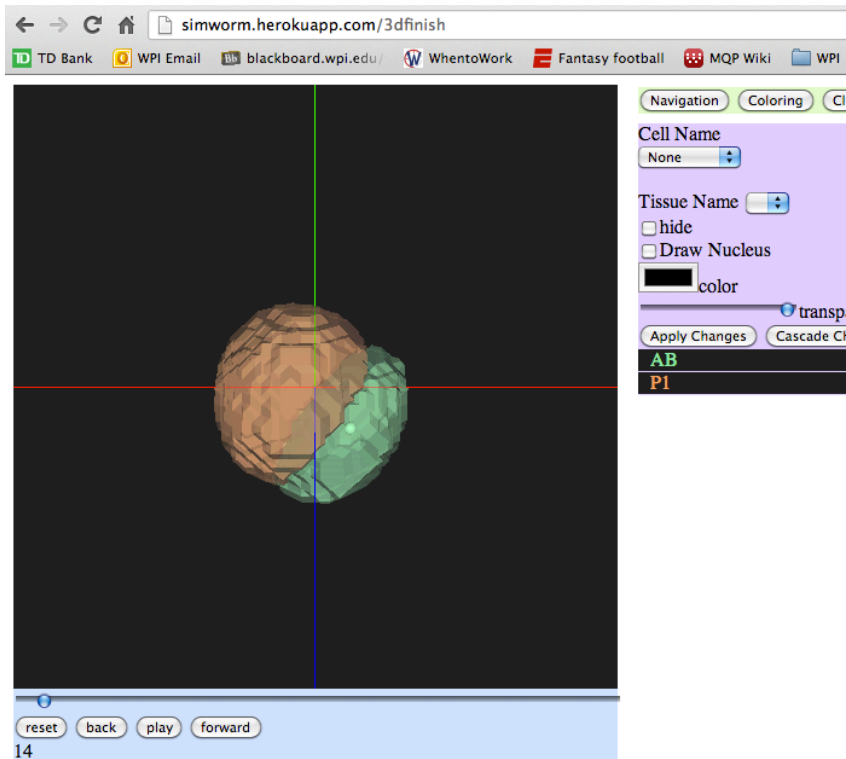


Figure 18: Simulation as seen on Google Chrome at [www.simworm.herokuapp.com/3dfinish](http://www.simworm.herokuapp.com/3dfinish)

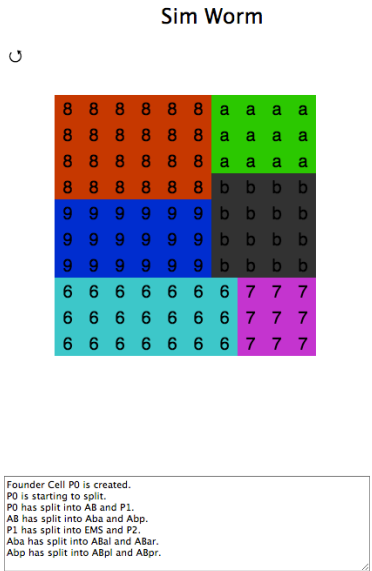
The simulation is available online and can be accessed online by anyone without the need of installing the simulation to your computer. As the simulation progresses and develops, hosting it online will allow the simulation to reach a greater audience and allow researchers from around the world to share their experiences and their findings.

## 4.6 Cell Visualization

### 4.6.1 Two Dimensional Model

The simulation was first created as a two-dimensional model (Figure 19). Initially, the model was created this way to make sure data was being taken in correctly and that everything was performing as expected. Taking into account the different percentages of the cell volume splits and the lineaging information that was known starting from the founder cells, the initial divisions of the simulation were run in 2D. A text dialog box was created to output information about the cell divisions that were occurring at the time and what events had already transpired in the simulation. New cells introduced into the simulation as a result of a cell division were colored

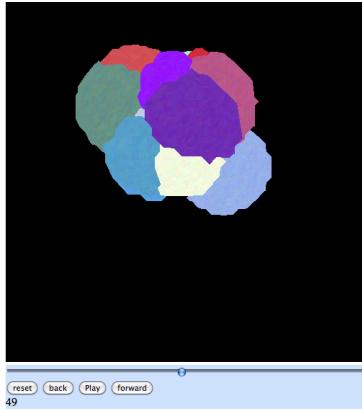
with a different color. This insured that everything was behaving correctly and that the lineage was being followed properly and the divisions were occurring as they should. When the two-dimensional model was complete we were prepared to make the transition to a three-dimensional model.



**Figure 19: Two-dimensional simulation of early divisions**  
 The simulation was first developed as a two-dimensional model for debugging purposes to test that the data being input into the model was performing as expected. A dialog box was used to ensure events were occurring correctly and in the right sequence. Once the simulation was believed to be running properly the simulation was transitioned to three dimensions.

### 4.6.2 Three Dimensional Model

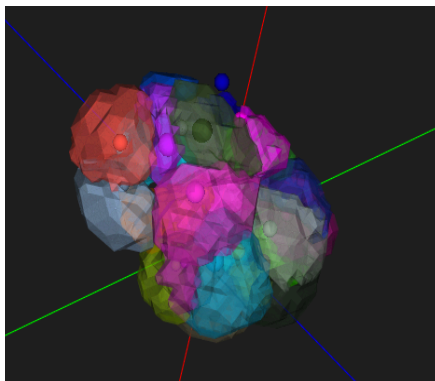
The simulation was eventually transitioned to a three-dimensional model that began to more accurately represent *C. elegans* embryogenesis (Figure 20). The use of the marching cubes algorithm in conjunction with metaballs allowed cells to transition from distinct polygons to more natural cell shapes. At this stage in the simulation, the embryo looked a bit misshapen with certain cells eventually moving outside the perimeter of the eggshell.



**Figure 20: Three-dimensional model - version one**

Version one of the three-dimensional model was developed using a combination of two computer algorithms, metaballs and marching cubes. The combination of these two algorithms allowed cells in the simulation to transition from distinct polygons to more natural cell shapes.

The second version of the three-dimensional model implemented some key changes and improvements (Figure 21).



**Figure 21: Three-dimensional model - version two**

In Version two of the three-dimensional model x, y and z-axis were made visible for better orientation of the embryo. The addition of lighting and transparency made the three-dimensional shape and form of the embryo more apparent and easier to view. By changing the transparency of individual cells, sections of the embryo can be seen more clearly and nuclei positions are better visualized.

X, Y and Z-axes are visible in the current simulation so the user is aware of the orientation of axes and embryo at all times. One major noticeable difference was the addition of lighting.

Lighting provided a more realistic and three-dimensional feel to the simulation especially when autorotation of the embryo is enabled. In addition to lighting, a transparency setting was added to allow changes to the transparency of certain cells to reveal accurate and visible nuclei positions

in the cells. Nuclei positions were not visible before, but using the iSPIM data, nuclei position and nuclear radius values were used to rendered nuclei as a generic sphere object in each cell.

## 5 Results and Discussion

### 5.1 Previous Model Comparisons

Compared to the previous model, there are many ways in which our model simulates the development of *C. elegans* in a more realistic way. Although certain signaling and gene expression is present in the previous model and is not currently in ours, visually the model is more accurate and behaves more similarly to that of an actual *C. elegans* embryo (Figure 22).

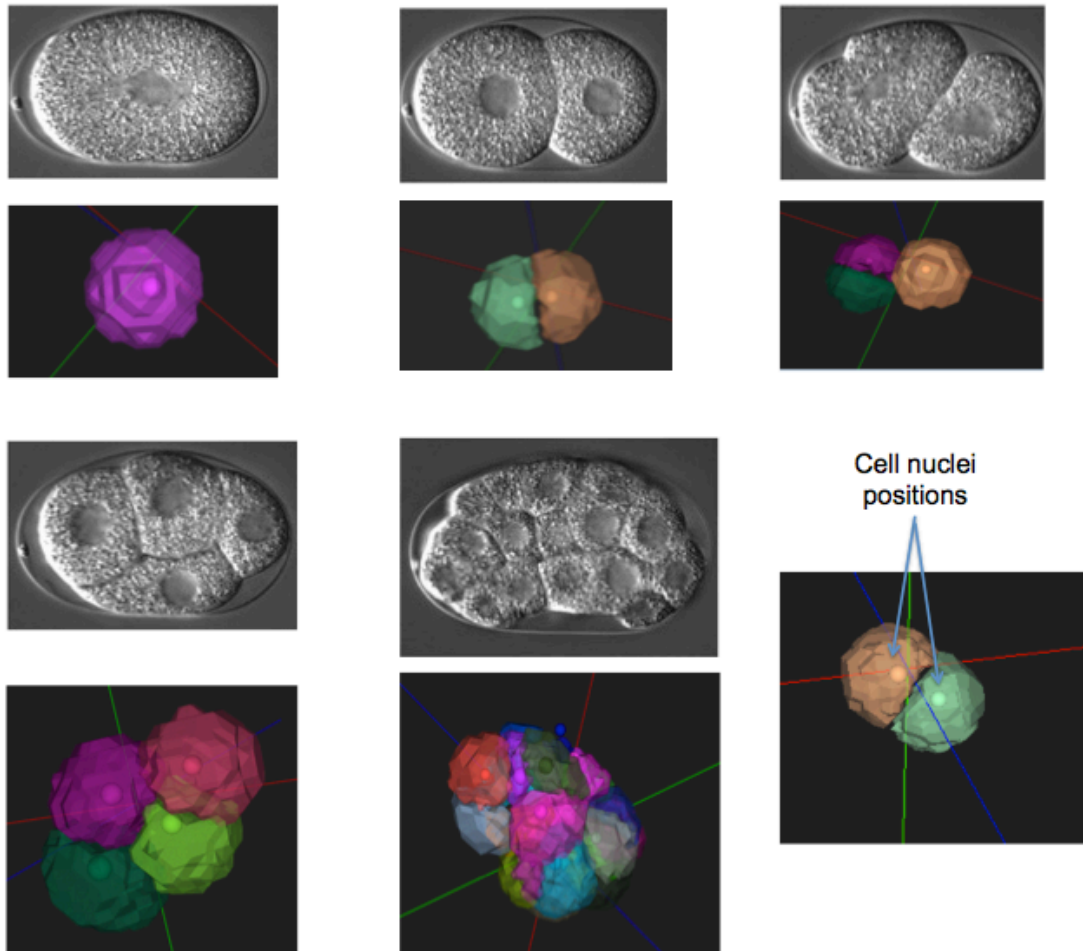


Figure 22: Laboratory micrographs *C. elegans* compared to screenshots of simulation  
Microscopy images of various stages of *C. elegans* development are shown in gray scale side by side with screenshots of the simulation. The model is represented as a three-dimensional image and the microscope images are shown in two dimensions however, the accuracy of the model is apparent. Specific nuclei positions within the cells are also seen within the model.



In our model, the cell positions are more accurate because they are taken from nuclei positions from an actual worm. In the previous simulation, certain cells were found to extend outside of the eggshell after a certain number of cell divisions had occurred. Our model has all the cells fitting within the eggshell shape. Figure 23 shows a comparison of our model and the previous model at the 30 cell stage. Our model shows colored cells represented using metaballs that more accurately represent the shape of cells at different stages of development rather than just cells dividing into spheres. The use of multiple metaballs per cell creates a more realistic appearance of the cell divisions throughout the development process. Programmed cell death is also more realistically portrayed, as dying cells slowly shrink down and appear to be engulfed by neighboring cells.

The soft body detection of our model allows the surfaces of cells to interact in a more natural way within the embryo. Cells that result from a division are introduced into the model in a way that preserves the natural positioning of the cells within the embryo without disturbing neighboring cells or creating unnecessary gaps or spacing between cells. When comparing the most recent simulation with real life embryo movies or still images, the cell shapes and cell movements of the organism are found to be very genuine.

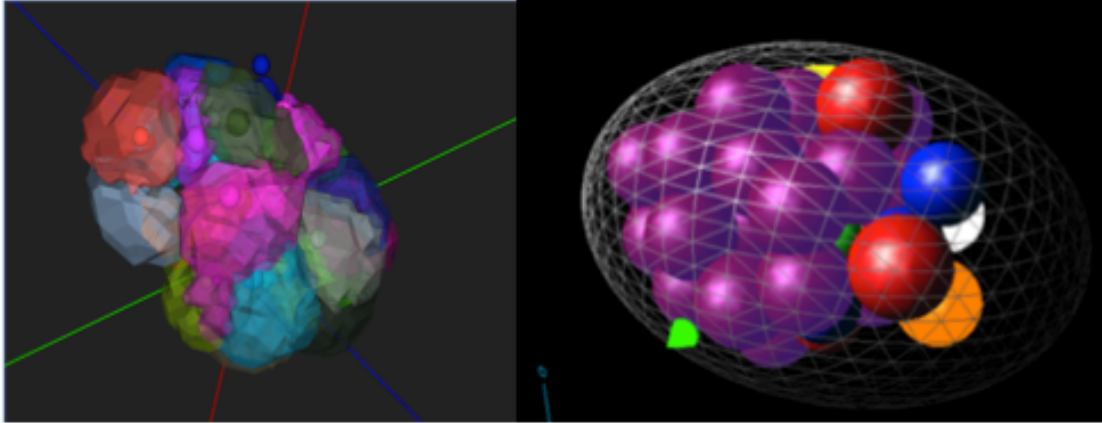


Figure 23: Current simulation compared to previous simulation

The current simulation is shown side by side with the previous model. The current model has a more realistic visualization of how cells change shape as they push against one another in the cell, as opposed to the spherical representation in the old model. Cells in the newer model also appear to stay contained within the restrictions of the eggshell better than the previous model.

## 5.2 Existing Infrastructure

The current simulation has the infrastructure available for the addition of some features that unfortunately were not incorporated into the simulation. Because the *mig-10 (ct41)* strain was not entirely generated in lab we were unable to collect data from the mutant strain to incorporate into the model. Despite the lack of mutant data, the infrastructure exists for the addition of future data that will be collected and used in the model to create a mutant simulation. This new data and mutant simulation will be used to help further our understanding of the *mig-10* mutation and neuronal migrations of the CAN neuron.

We were not able to implement cell signaling in our model but the infrastructure exists for cells to both send and receive signal cues. With the *C. elegans* lineage completely known, there are many ways in which the implementation of certain genes in the model could have been linked and used to trigger the differentiation of certain cell types or certain key events. Although this was ultimately not completed, the internal infrastructure of the model is built in such a way that

the addition of gene expression and protein signaling would have been the next step in the model's development. Down the road, we will have the ability to portray more cell movements and shape changes in the model based on specific programmed signal cues now that the infrastructure is available for future projects to implement.

### 5.3 Embryonic Marker

One goal of the project was to create a strain containing the *mig-10 (ct41)* mutant along with nuclear markers, to allow us to collect images of development in the mutant background that would be compared to the simulation. We were successful in creating two target strains: the RY1220 strain, which contains the *mig-10 (ct41)* mutation, the two GFP markers (*ceh-10::GFP* and *pgp-12::GFP*), and the RY1221 strain, which contains the mutation of interest, both GFP markers and the larval embryonic marker (*his-72:: HIS-72 :: wcherry*). We were unable to generate a strain that contained the embryonic marker due to initial difficulties in identifying the marker and later problems associated with its passaging from parents to offspring. Though we were unable to create the strain we wanted, the strains we did create will be able to be used by a future group to create the strain desired.

Throughout the experiments in lab many of the markers were crossed into the worm with ease. Many of these markers were found and identified very easily using the fluorescent dissecting scope. The marker however which caused the most problems was the embryonic marker. Unlike the red larval marker, the embryonic marker was difficult to spot because of the short presence it had in the overall development of the worm. The embryonic marker is only present in the earliest stages of egg development and never seen as an adult or even young adult. This marker is present in the earliest stages of the egg prior to any cell division or any folding stages. Due to the

timeframe we were allowed to search for the embryonic marker, accurate identification of the marker was found to be very difficult. Early in our lab work larval markers were very easy to classify on the dissecting microscope scope and initially we used the same microscope to identify the embryonic. This method was eventually deemed too inaccurate as the dissecting scope could allow us to view the eggs at an early enough stage to be sure we were seeing the embryonic marker not the larval marker. Below are the two subsequent methods used to verify if the marker was truly present.

### **5.3.1 Method 1: egg visualization on slides**

Eggs were mounted onto microscope slides using the egg prep procedure (see appendix). The embryos were then viewed on the slides under the compound microscope at varying magnifications. Using this method we identified embryos in the earliest stages of development to determine if the embryonic marker was indeed present.

### **5.3.2 Method 2: Embryo Visualization *in utero***

Gravid hermaphrodites (adult worms carrying fertilized eggs) were mounted on slides and views using a compound microscope. This allowed for eggs as well as oocytes (the unfertilized eggs) to be viewed all at once. Figure 24 shows a sketch of a gravid worm and labels the important identifying structures. Using the spermatheca (referred to in the figure as “sperm”) as a reference point, eggs were observed developing within the worm. Having the whole spectrum of embryo development in view under the compound scope insured a better view and a much more effective procedure for determining the presence of the desired marker, as seen in Figure 25, which compares the control worm in which the marker is present and the heterozygous worm, which does not show the marker.

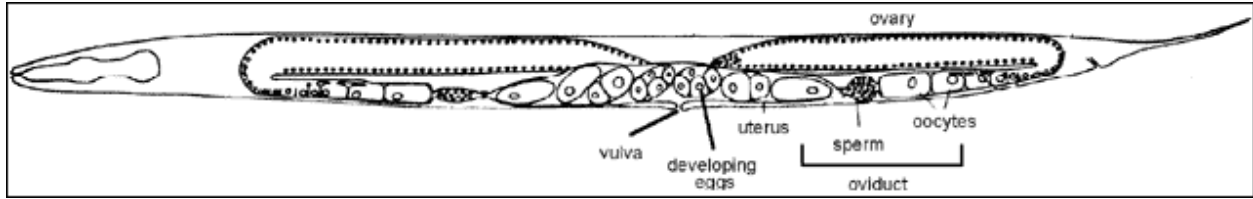


Image taken from [avery.rutgers.edu](http://avery.rutgers.edu), "C. elegans as a Model System"

**Figure 24: Gravid adult worm anatomy**

The adult hermaphrodite reproductive system is comprised of unfertilized gametes (oocytes), which are stored along the side of the worm. The spermatheca, labeled in the figure as sperm, fertilizes the oocytes as they enter the uterus and the developing embryos remain in the uterus until they are pushed out the vulva as the worm lays the eggs. The spermatheca can be used as a reference point to distinguish oocytes from embryos when viewing a gravid hermaphrodite on a microscope.



**Figure 25: Comparison of homozygous and heterozygous *pie-1::H2B::wcherry* embryonic marker view in gravid hermaphrodite**

Panel A shows the the cross progeny of N2M and BV117, which is known to be heterozygous for the embryonic *pie-1::H2B::wcherry* marker and the larval *his-72::HIS-72::wcherry* marker. The red shown in the body comes from the larval *his-72::HIS-72::wcherry* marker, while the oocytes shown no fluorescence indicating that *pie-1::H2B::wcherry* is not visible in the heterozygous state. Panel B is the control and homozygous for both markers. The *his-72::HIS-72::wcherry* marker is seen in the body of the adult worm and the oocytes and embryos are expressing the embryonic *pie-1::H2B::wcherry* marker.

Panel A of Figure 25 shows the progeny of an N2 male and a BV117 female that is heterozygous for the marker. The red that is visible in the body of the worm is from the larval marker, but the oocytes and the embryos are clearly lacking any red marker, showing that the embryonic marker is not visible in the heterozygous form. Panel B is an image of the control BV117 strain that shows both the larval marker in the body of the worm and the embryonic marker in the oocytes and embryos.

### 5.3.3 Cross and future recommendations

We were unable to generate the data for the *mig-10 (ct41)* mutant worm due to our difficulties in getting the embryonic marker to be homozygous. When we were unable to see the marker, we restarted the cross to make sure the missing marker was not due to human error, but due to problems passaging the marker. In order to overcome the challenge faced by crossing this marker, we propose the use of a linked marker indicator in the future to help force homozygosity for the red embryonic marker. We know that the marker is located on chromosome IV, so we suggest creating a male line that would have a fluorescent marker for something else located on the chromosome, close enough to be linked to the embryonic marker. Then, crossing that into the BV117 strain and letting the cross-progeny self-cross, we would be able to pick from ones that should have to be homozygous for the embryonic marker by not having the indicator marker present.

## 5.4 Future Model Implementations

Despite a realistic simulation result there are still certain things that we would like to see changed and implemented in the future to improve the simulation experience for the user. We hope that our recommendations to the next group will be taken into account and used to enhance the accuracy and scientific and educational value of the simulation.

Our version of the simulation is a very deterministic model. We were unable to implement some of the features that would allow the simulation to run inferred from minimal data points. We have implemented the framework that would allow for future incarnations to run simulations that would interpret the data from finite rules the user would be able to program. We would

especially like to see these rules implemented for features such as modeling the complexities associated with wnt signaling and expression patterns, showing how the signals, or mutant variations, can affect neuronal development and migrations. It is our hope that future models will display cell migrations and shape changes based on cues and interactions with other cells rather than just directing cells specifically where to migrate.

Some of the earlier cell divisions in the model make some of the earlier cells appear to be rather spherical when their shape should mold more to the shape of the eggshell. In the future we would like to see a secondary density function implemented to compensate for the unnatural shape of the cells during the earlier divisions in development. This may be achieved by assigning a different density function to these earlier cells, which we know to be irregularly shaped. Additionally we would like to see all shape changes implemented for all cell and tissue types. The current simulation is pretty accurate up to gastrulation but does not demonstrate some of the later stages of development such as ventral enclosure. With our focus on *mig-10* it would be nice to see more detailed neuronal migrations introduced into the model.

Some changes to the user interface could help better the simulation experience. We would like to see more detailed information pulled up for specific cells on the fly. The user would be able to select one cell by clicking on it and this would prompt a popup box outlining key information about the cell for example the founder cell, what tissue type the cell is and if the cell is known to die at some point during development. An improved cell counter and timer would help the user know exactly at what point the simulation is at without having to estimate cell numbers or the amount of time elapsed. Since the simulation does not run as efficiently as possible, the addition of the detail addressor would allow the user to detail certain parts of the simulation more than

others. With this addition, if the user were focusing on looking at a specific lineage of cells they would have the option of increasing the detail level of this portion of the embryo alone without having to render details in areas that are not of interest.

Improved control to the library of cells, divisions and cell names for the user would allow biologists to input more personalized findings and use the model in a way that they can tailor the model to their findings or area of focus. Currently the user does not have the ability to go in and change things in the library with ease but this ability would allow significant customization and usability.

One of the biggest implementations we are most excited about seeing is the introduction of an online community. Since the simulation has been available on the Internet since some of its earliest stages we would like to see this feature maximized. The hope is that an online community will be developed in the future to allow biologists to share modifications and changes of their own by uploading them to the Internet for others to see. Unique mutations and model modifications can be downloaded by others and run to learn about other biologist's research and unique mutation findings.



## 6 References

Blinn, J F. "A generalization of algebraic surface drawing." ACM Transactions on Graphics (TOG) 1.3 (1982): 235-256.

Bogdanova N, Jajosky J, Lloyd N, Stolzar L. Computer Simulation of *C. elegans* Embryogenesis, Major Qualifying Project. Rep. Worcester Polytechnic Institute: n.p., 2004. Print.

Bonabeau E. "Agent-based modeling: Methods and techniques for simulating human systems." Proceedings of the National Academy of Sciences of the United States of America 99.Suppl 3 (2002): 7280-7287.

Chin-Sang I D, Chisholm A D. Form of the worm:: genetics of epidermal morphogenesis in *C. elegans*. Trends in Genetics. Volume 16, Issue 12, 1 December 2000, Pages 544-551.

Conradt B, Xue D. Programmed cell death (October 06, 2005), WormBook, ed. The *C. elegans* Research Community, WormBook, doi/10.1895/wormbook.1.32.1, <http://www.wormbook.org>.

Gilbert S F. Developmental Biology. 6th edition. Sunderland (MA): Sinauer Associates; 2000. Early Development of the Nematode *Caenorhabditis elegans*. Available from: <http://www.ncbi.nlm.nih.gov/books/NBK10011/>.

Harterink M, Kim D H, Middelkoop TC, Doan DT, van Oudenaarden A, Korswagen H C. Neuroblast Migration along the Anteroposterior Axis of *C. Elegans* Controlled by Opposing Gradients of Wnts and a Secreted Frizzled-related Protein. Journal. Development. 2011. Print.

"Introduction to *C. Elegans*." Introduction to *C. Elegans*. Rutgers University, n.d. Web. 24 Apr. 2013. <<http://avery.rutgers.edu/WSSP/StudentScholars/project/introduction/worms.html>>.

Lorenson W E, Cline H E. "Marching cubes: A high resolution 3D surface construction algorithm." ACM Siggraph Computer Graphics. Vol. 21. No. 4. ACM, 1987.

"Management Science - Computer Simulation." Reading. George Mason University Classweb Directories. George Mason University. Web. <<http://classweb.gmu.edu/aloerch/simulation540.pdf>>.

Manser J, Wood W B. "Mutations Affecting Embryonic Cell Migrations In *Caenorhabditis Elegans*." Developmental Genetics 11.1 (1990): 49-64. Print.

Murray, J I, Bao Z. "Automated Lineage and Expression Profiling in Live *Caenorhabditis Elegans* Embryos." Cold Spring Harb Protoc 2012.8 (2012).

Quinn C C, Wadsworth W G. "Axon Guidance: Asymmetric Signaling Orients Polarized Outgrowth." *Trends Cell Biol* 18.12 (2008): 597-603. Print.

Rutgers. "Introduction to C. Elegans." Introduction to C. Elegans. Rutgers the State University of New Jersey, 2012. Web.

<<http://avery.rutgers.edu/WSSP/StudentScholars/project/introduction/worms.html>>.

Sendoel, A. and Hengartner, M. O. 2009. Cell Death in C. elegans. eLS. Available from: <http://www.els.net/WileyCDA/ElsArticle/refId-a0021563.html>.

"Simulation Overview." Simulation Overview. Imagine That Inc., n.d. Web. 22 Apr. 2013. <[http://www.extendsim.com/sols\\_simoverview.html](http://www.extendsim.com/sols_simoverview.html)>.

Silhankova M, Korswagen H. "Migration of Neuronal Cells along the Anterior–posterior Body Axis of C. Elegans: Wnts Are in Control." *Current Opinion in Genetics & Development* 17.4 (2007)

Sulston JE, Schierenberg E, White JG, Thomson JN (1983). The Embryonic Cell Lineage of the Nematode *Caenorhabditis elegans*. *Development* 100: 64-119.

Wormatlas. "INTRODUCTION TO C. Elegans ANATOMY." Wormatlas *Caenorhabditis Elegans*. 2006. Web.<http://www.wormatlas.org/ver1/handbook/anatomyintro/anatomyintro.htm>>.

Wu Y, Ghitani A, Christensen R, Santella A, Du Z, Rondeau G, Bao Z, Colón-Ramos D, Shroff H. "Inverted selective plane illumination microscopy (iSPIM) enables coupled cell identity lineaging and neurodevelopmental imaging in *Caenorhabditis elegans*." *Proceedings of the National Academy of Sciences* 108.43 (2011): 17708-17713.

Zinovyeva A Y, Yamamoto Y, Sawa H, Forrester W C. Complex Network of Wnt Signaling Regulates Neuronal Migrations During *Caenorhabditis Elegans* Development. Tech. N.p.: Genetic Society of America, 2008. Print.

# Appendix A: Crosses

## Cross 1:

Creating: RY1221

$$\frac{Mig-10 (ct41)}{Mig-10 (ct41)} \cdot \frac{ceh-10::GFP}{Ceh-10::GFP} \frac{pgp-12::GFP}{pgp-12::GFP} \cdot \frac{Pie-1::wcherry}{Pie-1::wcherry} \cdot \frac{his-72::wcherry}{his-72::wcherry}$$

## Step 1: N2M x RY 1220

$$\frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \times \frac{Mig-10 (ct41)}{Mig-10 (ct41)} \cdot \frac{ceh-10::GFP}{Ceh-10::GFP} \cdot \frac{pgp-12::GFP}{pgp-12::GFP} \frac{+}{+} \cdot \frac{+}{+} \xrightarrow{yields}$$

$$\frac{+}{Mig-10 (ct41)} \cdot \frac{+}{Ceh-10::GFP} \cdot \frac{+}{pgp-12::GFP} \cdot \frac{+}{+} \cdot \frac{+}{+}$$

## Step 2: Cross 1 Progeny x BV117

$$\frac{+}{Mig-10 (ct41)} \cdot \frac{+}{Ceh-10::GFP} \cdot \frac{+}{pgp-12::GFP} \cdot \frac{+}{+} \cdot \frac{+}{+} \times \frac{+}{+} \frac{ceh-10::GFP}{Ceh-10::GFP} \cdot \frac{+}{+} \frac{Pie-1::wcherry}{Pie-1::wcherry} \cdot \frac{his-72::wcherry}{his-72::wcherry} \xrightarrow{yields}$$

$$\frac{+}{Mig-10 (ct41)} \cdot \frac{ceh-10::GFP}{Ceh-10::GFP} \cdot \frac{+}{pgp-12::GFP} \cdot \frac{+}{Pie-1::wcherry} \cdot \frac{+}{his-72::wcherry}$$

## Step 3: Single Cross 2 progeny until homozygous

$$\frac{+}{Mig-10 (ct41)} \cdot \frac{ceh-10::GFP}{Ceh-10::GFP} \cdot \frac{+}{pgp-12::GFP} \cdot \frac{+}{Pie-1::wcherry} \cdot \frac{+}{his-72::wcherry} \times \text{self} \xrightarrow{yields}$$

$$\frac{Mig-10 (ct41)}{Mig-10 (ct41)} \cdot \frac{ceh-10::GFP}{Ceh-10::GFP} \cdot \frac{pgp-12::GFP}{pgp-12::GFP} \cdot \frac{Pie-1::wcherry}{Pie-1::wcherry} \cdot \frac{his-72::wcherry}{his-72::wcherry}$$

## Cross 2:

Attempting:

$$\frac{Mig-10 (ct41)}{Mig-10 (ct41)} \cdot \frac{ceh-10::GFP}{Ceh-10::GFP} \cdot \frac{Pie-1::wcherry}{Pie-1::wcherry} \cdot \frac{his-72::wcherry}{his-72::wcherry}$$

### Step 1: N2N x BV117

$$\frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \times \frac{+}{+} \cdot \frac{ceh-10::GFP}{Ceh-10::GFP} \cdot \frac{+}{+} \cdot \frac{Pie-1::wcherry}{Pie-1::wcherry} \cdot \frac{his-72::wcherry}{his-72::wcherry} \xrightarrow{\text{yields}}$$

$$\frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+}$$

### Step 2: Cross 1 Progeny with RY1221

$$\frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \times \frac{Mig-10 (ct41)}{Mig-10 (ct41)} \cdot \frac{ceh-10::GFP}{Ceh-10::GFP} \cdot \frac{pgp-12::GFP}{pgp-12::GFP} \cdot \frac{+}{+} \cdot \frac{his-72::wcherry}{his-72::wcherry} \xrightarrow{\text{yields}}$$

$$\frac{+}{+} \cdot \frac{ceh-10::GFP}{Ceh-10::GFP} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{his-72::wcherry}{his-72::wcherry}$$

### Step 3: Single Cross 3 Progeny until homozygous

$$\frac{+}{+} \cdot \frac{ceh-10::GFP}{Ceh-10::GFP} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{+}{+} \cdot \frac{his-72::wcherry}{his-72::wcherry} \times \text{self} \xrightarrow{\text{yields}}$$

$$\frac{Mig-10 (ct41)}{Mig-10 (ct41)} \cdot \frac{ceh-10::GFP}{Ceh-10::GFP} \cdot \frac{Pie-1::wcherry}{Pie-1::wcherry} \cdot \frac{his-72::wcherry}{his-72::wcherry}$$

## Appendix B: Supplementary Images

Image 1: Gravid adult worms looked under microscope for embryonic marker in developing embryos and oocytes



Image 2: Embryos viewed for embryonic marker microscope under compound microscope

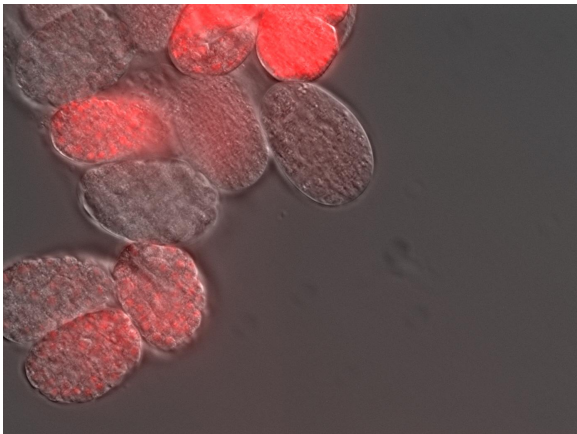
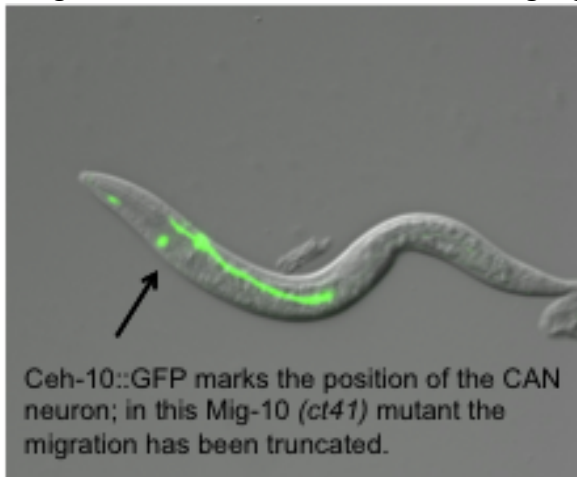



Image 3: GFP marker in mutant worm highlights CAN neuron and truncation of excretory cell



# Appendix C: Presentation Poster



**WPI**

## Modeling Development in *C. elegans*

Daniel Brandon BB '13, Josie Cromartie BB '13, William Decker CS '13  
 Advisors: Professor Elizabeth Ryder - Department of Biology and Biotechnology  
 Professor Mathew Ward - Department of Computer Science

---

### Abstract

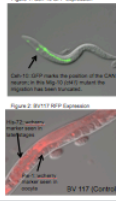
This project aimed to simulate the embryonic development of *Caenorhabditis elegans* from fertilization to hatching to serve as a dynamic research and educational aid that could be added to or modified by future research. Existing knowledge from the literature and mutant strains generated in lab were used to design and refine the processes modeled. The simulation was rendered using metaballs and marching cubes to provide a three-dimensional computer simulation that can help further understanding of the development process.

### Methods – Lab

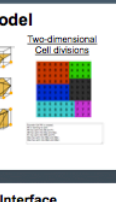
**Strain Design**

1. Cross *gls-1* RNAi with *gls-1* RNAi
2. Cross *gls-1* RNAi with *gls-1* RNAi
3. Cross *gls-1* RNAi with *gls-1* RNAi
4. Cross *gls-1* RNAi with *gls-1* RNAi
5. Cross *gls-1* RNAi with *gls-1* RNAi

**Figure 1. *gls-1* GFP Expression**

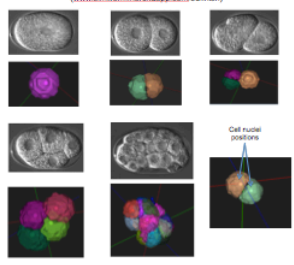


**Figure 2. *gls-1* GFP Expression**



### Computer Simulation

(www.silvaworm.herokuapp.com/@rindsh)



Cell nuclei positions

### Background

#### Embryogenesis in *C. elegans*

- Larvae approximately 14 hours
- The majority of the cells are generated in the first 5 hours of development

#### Gastrulation

- At the 28 cell stage, cells will eventually form the gut and internal structures begin to migrate into a space in the interior of the embryo known as the blastocoel.
- The major germ layers are established, providing the foundation for organogenesis

#### Ventral enclosure

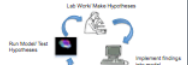
- Marks the end of gastrulation as the cells forming the hypodermis migrate to enclose the embryo and cover the opening through which the gut precursors entered the blastocoel

#### Mig-10

- Encodes for a protein that is required for cell migration and axon outgrowth
- Mig-10 (g41) mutation results in truncation of neuronal migration and the excretory cell


#### Simulation

- All the model is developed and tested, it is compared to the real life data
- If the model does not match, then the model is refined and retested.




### Methods – Model

#### Marching cubes



#### Two-dimensional Cell divisions



### Conclusions and Recommendations

- The model runs from fertilization through gastrulation, and has the infrastructure for the addition of specific gene expression and important protein signaling


#### Future Implementations

- Continue the simulation through hatching
- Cell movements and shape changes based on programmed signal cues
- Be able to model neuron migration and study the effects of the mig-10 (g41) mutation.
- Ability to render frames within a desired range since buffering simulation can take time
- Add cell counter and time in simulation window
- Online community where users can share their modifications with others

### Features and User Interface


#### Coloring

The coloring tool allows the user to set colors for individual cells or whole organs. The "toggle changes" option allows the user to apply colors to cell lineages.




#### Cross Section Capability

The dipping tool allows the user to view different cross sections along any axis of the model at any point.



#### Navigation

The user interface allows you to manually adjust the position of the model using sliders or by clicking and dragging on the model in the window.



### Project Goals and Objectives

#### Goals of lab:

- To generate mutant strains and use expressed markers to test model predictions

#### Goals of simulation:

- To develop a computer simulation that accurately depicts the development of *C. elegans* for future scientific and educational purposes

### Acknowledgments

- Anthony Santella and Zhong Bao for help with lineaging and strain generation
- Yiyoung Wu and Harl Shrof for their help with embryo movies and data collection

# Appendix D: Procedures

## Maintaining a line:

From a growing plate (just starved or still with a little food left on it, with worms of all ages)  
Pick three L4's or young adults from the plate to a new plate

From a very starved plate (no food, mostly dauer worms on the plate)  
Cut out a small chunk of agar, using a flamed spatula. Transfer the chunk to a new plate.

\*Don't forget to label the bottom of all plates with the strain name and date

## Making an N2 mating plate

1. Get N2 males from someone who has an active mating plate
2. Label the plate "N2 Males" Pick 10 young males to do it. Don't worry if you take some eggs or hermaphrodites along with the males
3. Label another plate "N2M" and date it. Pick the males to it. No extraneous eggs or hermaphrodites should come along with the males this time
4. Pick three N2 L4 hermaphrodites to a new plate. again, don't worry if extraneous worms come along
5. Repick the three hermaphrodites to the N2M plate
6. Place the plate into the incubator. If you put the plate at 20 degrees celsius you will need to make another mating plate in four days. If you put the plate at 15 degrees, you will need to make another in a week. When making a new plate, keep the previous one, in case the new mating plate does not take.

## **Bleaching Worms to Remove Contaminants**

1. Make bleach solution on a day when you have at least 10 gravid hermaphrodites of your contaminated strain

### **Bleach solution recipe (1ml)**

675  $\mu$ l M9

200  $\mu$ l Bleach

125  $\mu$ l 5M NaOH

Mix well in an Eppendorf Tube

2. Put a drop of the bleach solution onto a new plate away from the food

3. Pick about 10 gravid hermaphrodites to the bleach drop. Try not to deposit too much E. Coli in the spot with them. The worms will thrash in the bleach solution, and eventually die. Throw away the bleach solution; it should be made up fresh every day that you use it.

4. Check the plate the next day. The eggs in the hermaphrodites will have hatched and L1s should be making their way to the food spot. (sometimes this may not happen for another day). Pick L1s to a new plate.

5. You may need to do more than one bleaching of your strain to get rid of persistent contaminants



## Appendix E: References and sources for programming Language's and Libraries

Ruby on Rails - <http://rubyonrails.org/>

HTML5 - <http://www.w3.org/html/wg/drafts/html/master/>

jQuery - <http://jquery.com/>

Three.js - <http://threejs.org/>

Heroku - <https://get.heroku.com/>

JavaScript - <http://www.w3.org/standards/webdesign/script.html>

PostgreSQL - <http://www.postgresql.org/>

OpenGL - <http://www.opengl.org/>

## Appendix F: User's Guide

1. Install Google chrome; Firefox and Opera work but require the WebGL extension to be manually installed
2. Open Chrome and navigate to <http://simworm.herokuapp.com/3dfinish>
3. To change the time which is rendered click and drag the slider at the bottom of the page.
4. To move the camera click and drag in the rendered image; holding down 's' will change to zoom mode, holding down 'd' will change to pan mode
5. To rotate the embryo click on the navigation button in the upper right of the screen, then click and drag any of the three sliders on the right side of the screen.
6. To change the color of the cells click on the coloring button in the upper right. Select a cell to edit by the drop down menu or clicking on a cell listed in lower right of the screen. By clicking on the color box the user can select a new color. Clicking apply changes will apply changes and cascade changes will change that cell and all its progeny.
7. To clip the rendering click on the clipping button in the upper right and then click and drag either end of the any of the three sliders on the right side of the screen.

## Appendix G: Programmer's Guide

1. Getting the rails app to run on local host
  - a. This is a good tutorial for getting started with Ruby on rails.  
[http://guides.rubyonrails.org/getting\\_started.html](http://guides.rubyonrails.org/getting_started.html)
  - b. The git application is needed for version control and pushing to heroku, here is a tutorial on setting that up. <https://help.github.com/articles/set-up-git>
  - c. If you are on mac I suggest rvm <https://rvm.io/rvm/install/>
  - d. To start the sever, in Terminal
    - i. navigate to the folder 'SimWormWeb'
    - ii. enter 'bundle install'
    - iii. enter 'rake db:create'
    - iv. enter 'rake db:migrate'
    - v. enter 'rails s'
  - e. Now go to chrome and visit local host
2. Describing the files in the zip
  - a. SimWorm is work done in A-term 2012 (python)
  - b. SimWorm C is work done in C-term 2013 (python)
  - c. SimWormWeb is where the rails app is (most current version of sim)
  - d. ZD\_RW103.... Is the data given to us about the nuclei centers in there is a file called re.py and it processes these rat data files.
  - e. The other files are general background
3. Code base in SimWormWeb
  - a. File hierarchy is standard rails (Model, View, Controller)  
<http://www.tutorialspoint.com/ruby-on-rails/rails-directory-structure.htm>
  - b. App/assets/javascript/3Dfinish.js is the code base for the most recent version of Simworm which is fairly well commented.
    - i. At the top of the file are some constants and prototypes
    - ii. All the jQuery stuff is for html manipulation
  - c. App/assets/javascript/Marching\_cubes.js is code for the marching cube algorithm.
4. Making changes in the database
  - a. Ultimately a database seed file should be created but for now all changes are made manually in the rails console.