



Genesis

Final Report

Sam Cohen	slc2206	Manager
Michael Wang	mlw2167	Systems Architect
Leo Stilwell	ls3223	Systems Architect
Jason Zhao	jsz2107	Language Guru
Saahil Jain	sj2675	Tester

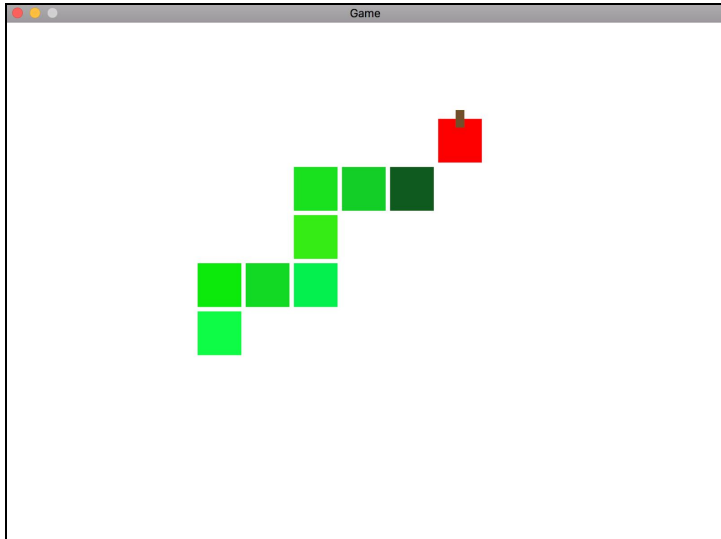
Table of Contents

Introduction	4
Language Tutorial	6
Language Manual	10
Project Plan	20
Architectural Design	54
Test Plan	56
Lessons Learned	64
Appendix	66

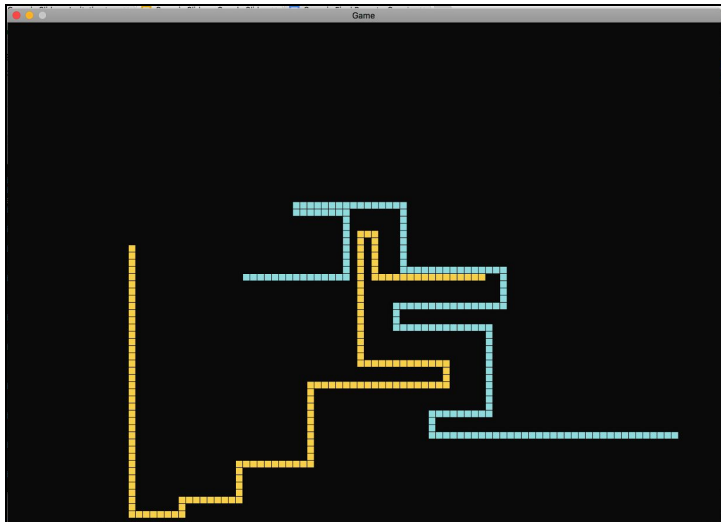
Introduction

Genesis is a special-purpose language inspired by popular game engines like Unity. The primary goal of Genesis is to facilitate the development of 2D interactive games. It provides a simple interface for users to process key input and it automatically handles graphic rendering details. Some examples of games that can be made with genesis are:

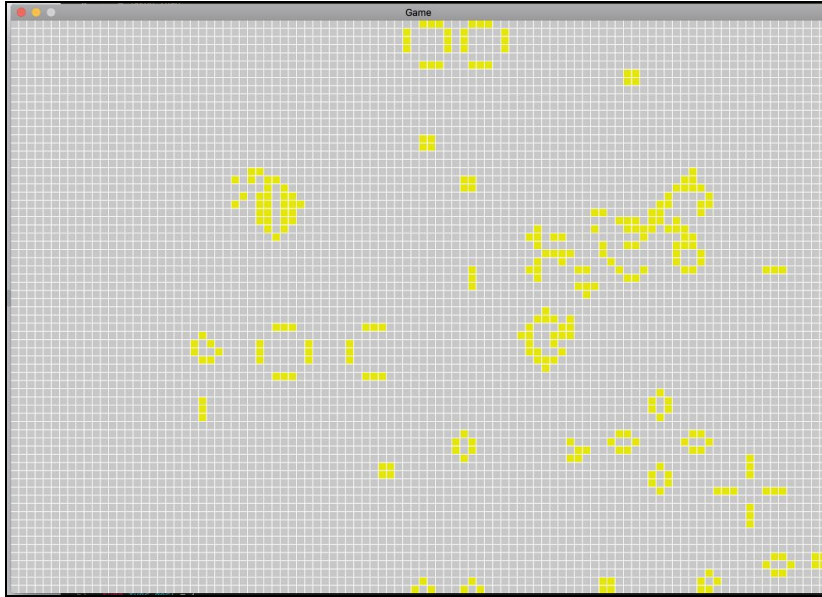
Snake:



Tron:



Conway's Game of Life:



Setup

Before using Genesis, there are a few housekeeping tasks that must be performed.

- Our language has a dependency on the SDL library ([found here](#)). If using a linux distro, you can simply run the following command in terminal.

```
sudo apt-get install libsdl2-dev
```

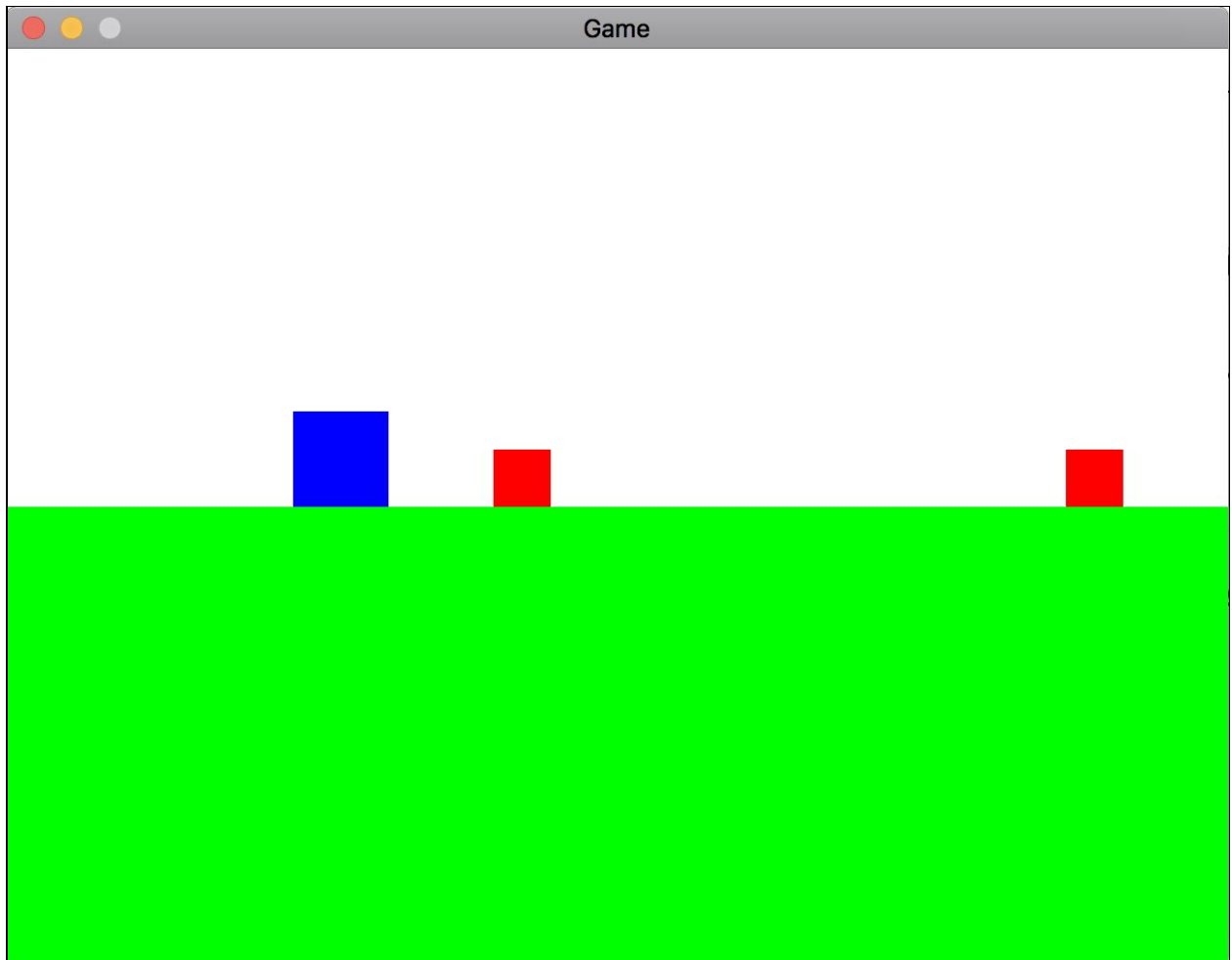
For other operating systems, see [SDL official documentation](#) for more detail.

If this step is performed properly, you should be able to type “make” in terminal to get the compiler up and running without any errors.

Language Tutorial

This tutorial will create the game jumpman, a game in which the user must avoid a series of incoming blocks by jumping over them.

The finished product will look like this:



To begin making the game, create a file named `jumpman.god`.

First, we'll declare the global variables needed by the entire game. We need:

- clusters to represent the player, the ground, and the incoming obstacles
- A boolean to keep track of whether the user is currently in the middle of a jump
- An integer to keep track of how many blocks the user has jumped over

```
cluster ground;  
cluster player;
```

```
cluster obstacle1;
cluster obstacle2;
bool jumping;
int score;
```

Next, we'll declare the main function of our program, which will initialize our clusters and start the game using the `startgame()` function:

```
int main(){
    color white;
    color green;
    color red;
    color blue;
    color yellow;

    white = #255, 255, 255#;
    red = #255,0,0#;
    green = #0,255,0#;
    blue = #0,0,255#;
    yellow = #255, 255, 0#;

    ground = $ 640, 240, 0, 240, 0, 0, green $;
    player = $ 50, 50, 150, 190, 0, 0, blue $;
    obstacle1 = $ 30, 30, 600, 210, 0, 0, red $;
    obstacle2 = $ 30, 30, 900, 210, 0, 0, red $;

    startGame(640, 480, white);
    return 0;
}
```

Next, we'll declare a function with the reserved function name `init()`, which will be called after the game window has been created and before any frames have been rendered. In the function, we'll draw all the clusters and initialize the *jumping* and *score* global variables we declared earlier.

```
void init(){
    ground.draw = true;
    player.draw = true;
    obstacle1.draw = true;
    obstacle2.draw = true;
    jumping = false;
    score = 0;
}
```

Finally, we'll declare a function with the reserved key name `update()`, and we'll make sure that the function matches the necessary function definition, which returns `void` and takes in an `int` that represents the current frame number.

```
void update(int f){
    //Remove the player and quit the game if the player hits an obstacle
    if ((player @ obstacle1) || (player @ obstacle2)){
        player.draw = false;
        print(score);
        quit();
    }

    //Accelerate the player object if it is currently jumping
    if (jumping){
        player.dy = player.dy + 2;
    }

    //Begin the jump animation if the user has pressed space
    if (keyDown("Space") && !jumping){
        player.dy = -25;
        jumping = true;
    }

    //Stop the user on the ground surface if it hits the ground.
    if (player @ ground){
        player.dy = 0;
        player.y = 190;
        jumping = false;
    }

    //Move the obstacle to the right side of the screen if it moves off
    if (obstacle1.x < -30){
        obstacle1.x = 640;
        score = score + 1;
    }
    if (obstacle2.x < -30){
        obstacle2.x = obstacle1.x + 225 + random(75);
        score = score + 1;
    }

    //Move the obstacles to the left
    obstacle1.x = obstacle1.x - 5;
    obstacle2.x = obstacle2.x - 5;
}
```

Finally, to run the program, navigate to the same directory as your jumpman.god file and execute the command:

```
./LetThereBe.sh jumpman.god
```

This will create an executable file named jumpman, which can be run like so:

```
./jumpman
```


Language Manual

Language Elements	10
Syntax	10
Reserved Keywords	10
Operators	11
Logical Operators	11
Comparison Operators	11
Arithmetic Operators	12
Game-specific Operators	12
Control Flow	13
Comments	13
Functions	14
User Defined Functions	14
Creating Functions	14
Calling Functions	14
Reserved Functions	14
Predefined Functions	15
Print Functions	15
Game Functions	15
Input Functions	16
Data Types	16
Primitives	16
Integers (keyword: int)	16
Floats (keyword: float)	16
String	17
Boolean	17
Casting	17
Complex Data Types	17
List	17
Color	17
Cluster	17
Making a New Cluster	18
Cluster properties	18

Language Elements

Syntax

Overall, Genesis is composed of three main elements: expressions, statements, and functions. Expressions can be further broken down into literals (integers, floating-point numbers, string, etc.), operations (addition, subtraction, comparison, etc.), and complex data type functions. The last group contains expressions for accessing the properties of Clusters and Lists. Statements are expressions that end in semicolons, or control-flow operations (eg. if-statements, loops, etc). Finally, functions form the remainder of the core syntax components of Genesis. More detailed descriptions of the language elements of Genesis are provided in the following sections.

Reserved Keywords

Keyword	Syntax	Semantics
new	new <primitive>[<int>;	The <i>new</i> keyword is used to initialize lists. Similar to the syntax of static arrays in Java, to create an array you specify the type of the array and the capacity of the array after the <i>new</i> keyword.
delete	delete <id>;	<i>delete</i> frees the memory occupied by the array <i>id</i> . This can only be used on arrays.

Operators

This section will be divided into 4 sections for organizational purposes, logical operators, comparison operators, math operators and custom, game-specific operators.

Logical Operators

Our language implements logical AND(\wedge), OR(\vee), and NOT(\neg).

Operator	Keyword
Logical AND (\wedge)	&&
Logical OR (\vee)	
Logical NOT (\neg)	!

These operators can be used in the following manner:

```
bool a = true;
bool b = false;
a && b; // Returns false
a || b; // Returns true
!a;     // Returns false
```

Comparison Operators

Our language contains all of the standard comparison operators, greater than, less than, equals, not equals, greater than or equals, less than or equals.

Operator	Language Symbol
Equals	==
Not Equals	!=
Greater Than	>
Less Than	<
Greater Than or Equals to	>=
Less Than or Equals to	<=

These operators can be used in the following manner:

```
int a = 1;
int b = 2;
a > b // Returns true
a == b // Returns false
a < b // Returns true
```

Arithmetic Operators

Our language implements the standard mathematical arithmetic operators including multiply, divide, add, subtract as well as a modulus operator.

Operator	Language Symbol
Multiply	*
Divide	/

Add	+
Subtract	-
Modulo	%

These operators can be used in the following manner:

```
int a = 1;
int b = 2;

a + a // Returns 2
b / a // Returns 2
a - a // Returns 0
```

Game-specific Operators

Our language implements two game specific functions, involving collisions between objects and key presses. These can be called within update functions.

Operator	Symbol
Collision Returns TRUE if two clusters are touching	@

Control Flow

Our language implements if statements, if-else statements, if-else statements, and while loops:

Control Structure	Description
<pre>if (<i>expression</i>) { <i>block</i> }</pre>	The code in <i>block</i> gets executed if <i>expression</i> evaluates to TRUE.
<pre>if (<i>expression 1</i>) { <i>block 1</i> } else { <i>block 2</i> }</pre>	If <i>expression 1</i> evaluates to TRUE, then <i>block 1</i> gets executed. Otherwise, <i>block 2</i> gets executed.
<pre>if (<i>expression 1</i>) {</pre>	If <i>expression 1</i> evaluates to TRUE, then <i>block 1</i>

<code>block 1</code> <code>} else if (<i>expression 2</i>) {</code> <code>block 2</code> <code>}</code>	gets executed. Otherwise, <i>block 2</i> gets executed if <i>expression 2</i> evaluates to TRUE.
<code>while (<i>expression</i>) {</code> <code>block</code> <code>}</code>	While <i>expression</i> evaluates to TRUE, block is executed.
<code>for (<i>expression 1</i> ; <i>expression 2</i> ; <i>expression 3</i>)</code> <code>{</code> <code>block</code> <code>}</code>	<i>expression 1</i> is an initialization expression. While <i>expression 2</i> is true execute the block, and on each execution of the block, run <i>expression 3</i>

Comments

Comments are written using the following operators. There are no nested comments.

Operator	Description
<code>//</code>	After the <code>//</code> symbol, anything written on the same line will be ignored
<code>/* */</code>	Anything between <code>/* */</code> will be ignored

Functions

User Defined Functions

Creating Functions

Users can define their own functions by using the format:

```
returnType functionName(argumentType argumentName, ...){ body }
```

For example:

```
int foo (int a, int b){
    return a;
}
```

Functions defined in the global scope can be called from any scope, while functions defined in classes are member functions of that class and can only be referred to when referencing an instance of that class.

Calling Functions

Calling a function involves writing the function name and then writing the arguments passed to the function in order, separated by commas inside of parenthesis. The function call will evaluate to the value returned by the function

For example:

```
int x = foo(5,6); //x equals 5
```

Reserved Functions

Definition	Description
int main()	The entry point of the program. Returns an integer number to the system.
void init()	A user defined function that automatically gets called immediately after the game is created, after the game window has been created and before any frames are rendered.
void update (int framenum)	A user defined function that is automatically called before every frame is rendered. The function takes in an integer that represents the total number of frames that have been rendered so far.

Predefined Functions

Print Functions

Definition	Arguments	Description
void print(int i)	i - the integer to print	Prints an integer to the console
void printf(float f)	f - the float to print	Prints a float to the console
void prints(string s)	s - the string to print	Prints a string to the console

Game Functions

Definition	Arguments	Description
------------	-----------	-------------

<code>void startGame(int width, int height, color background)</code>	width - the width in pixels of the game window height - the width in pixels of the game window background - the background color of the game window	Initializes a new game and creates the game window. After everything has been initialized, the user defined <code>init()</code> function is called once and <code>update()</code> is called on every frame.
<code>void quit()</code>		Closes the game window.
<code>void setFPS(int fps)</code>	fps - the frames per second at which frames will be rendered and update will be called	sets the frames per second that will be rendered by the game.
<code>int random(int max)</code>	max - the maximum valued random integer to generate	Generates a random integer between in the range [0, max).
<code>delete(cluster c)</code>	c - the cluster to delete	Deletes and deallocates memory for a cluster

Input Functions

Definition	Arguments	Description
<code>bool keyDown(string key)</code>	key - the name of the key to check	Returns true if the given key has been pressed for the first time.
<code>bool keyHeld(string key)</code>	key - the name of the key to check	Returns true if the given key has been held down continuously for consecutive frames.
<code>bool keyUp(string key)</code>	key - the name of the key to check	Returns true if the key is not currently pressed down.

The predefined input functions use the key names given by the SDL2 graphics library, which can be found at: https://wiki.libsdl.org/SDL_Keycode

Data Types

Our language implements 4 basic data types, integers, floats, booleans and strings, in addition to several complex data types. Keywords are type-sensitive.

Variable names can be any string composed of alphanumeric characters and “_” (underscore) as long as they are not a reserved name or keyword.

Primitives

Integers (keyword: int)

32-bit numerical integers stored using two’s-complement.

```
int a = 32;  
int b = -13;
```

Floats (keyword: float)

Double precision (64-bit) IEEE 754 floating point numbers.

```
float a = 32.4;
```

String

String must start with a alphabetic character and can then include any alphanumeric character afterwards.

```
string str = "Hello, World!";  
str.length(); // Evaluates to 13
```

Boolean

Can be either true or false.

```
boolean a = True;
```

Casting

All types are implicitly cast. For example, floats assigned to integers are automatically truncated.

Complex Data Types

List

Lists are statically-sized containers of objects stored on the heap. Lists provide no safety guarantees and are similar in semantics to C arrays allocated with `malloc()`. Because lists are stored on the heap, users must manage their own memory with the *new* and *delete* keywords.

```
int[] items;  
items = new int[10]; // Initialize a int list of size 10  
items[0] = 1; // Assign 1 to the first cell of items  
items[1] = 2; // items contain [1, 2, ...]  
delete items; // Free items
```

Color

Color objects are 3-integer tuples representing rgb values. It is defined using the pound symbol as follows, with the first, second, and third numbers representing the r, g, and b values:

```
color col = #110, 255, 0#;
```

Cluster

A cluster is a rectangular collection of pixels.

Making a New Cluster

New clusters are created using the dollar symbol. Its arguments are the cluster width, height, x position, y position, x velocity, y velocity, and color:

```
cluster c1 = $ 100, 100, 0, 0, 0, 0, col $;
```

Cluster properties

After being created, clusters and their properties can be modified using the `.` operator. Cluster properties are:

Property name	Type	Description
width	int	The width of the cluster, in pixels
height	int	The height of the cluster, in

		pixels.
x	int	The x position of the cluster, in pixels.
y	int	The y position of the cluster, in pixels
dx	int	The x velocity of the cluster. The cluster will automatically be moved with its set velocity.
dy	int	The y velocity of the cluster. The cluster will automatically be moved with its set velocity.
color	color	The color of the cluster
draw	bool	When set to true, the cluster will be displayed in the game window.

Project Plan

Planning Process

Our team established a common meeting time every Friday evenings to sync up and also to meet on Mondays after class. Meetings were used to assign work, discuss future direction of the project, as well as address any group problems / blockers.

Our team had a unique way of collaborating, which involved meeting for extended periods of time in order to reach major milestones. As opposed to splitting up work by file (scanner, semant, etc.), we mainly worked on implementing individual features / scenarios. Each group member was essential to the development process and had ownership over their specific features. Nevertheless, some specialization occurred, with some members focusing on the test platform and semantic checking while others focused on developing the graphics C library. All in all, each group member touched most every file (with the exception of some for graphics C). Each group member extensively helped others on blockers. Similarly, we worked collaboratively on the proposal and the LRM.

We worked iteratively by attempting to meet milestone goals and adjusting our next "sprint" goals based on results.

Programming Style Guide

For our style guide, we used the following rules:

1. Space indentation
2. Meaningful and concise variable names
3. Use of camel case for functions and variables in c
4. use of test- and fail- in test program names to specify whether tests were supposed to succeed or fail, respectively
5. The use of hyphen separated names for test cases
6. Similar naming of shared values between LLVM and C code
7. Code comments for confusing segments
8. The use of <type>Lit to specify tokens that hold values, and <type> to specify tokens that represent the type itself

Specification Process

Initially, we focused on specifying the functionality of our language, specifically thinking about what we wanted our language to be able to do. Our goal was to be able to create visual 2D Games, like Snake, Conway's Game of Life, and other such games. We then specified various features, as can be seen in our reference manual. Key focuses included ability to render frames, detect collisions, handle user input, and create game friendly design. Although we stayed true to our initial vision, our implementation of features ranging from design to syntax varied significantly as we developed our language.

Development Process

Our development process initially started by following the compiler stages, but we ended up implementing our language in terms of individual features / scenarios. We began by working on the AST and Scanner. However, we realized that implementing a feature (clusters, arrays, strings) would be more efficient given our setting. Thus, our development process was more parallel, and we were developing each of the stages of the compiler in conjunction, feature by feature.

Testing Process

After completing each individual feature, we created tests that ensured the functionality of that feature. Our tests checked for failure cases / edge cases as well as successes. We also checked that the output of errors was correct. Specifically, since our language dealt with visual components, we split our tests into visual as well as regression tests, which we describe further in the Test Plan.

Roles and Responsibilities

Formally, Sam was the project manager, Saahil was the tester, Leo / Michael were the system architects, and Jason was the language guru. In actuality, our roles / responsibilities were fluid and covered the different domains. However, each person was in charge of making sure that their particular formal responsibility was on track / implemented in the end. In the credits section underneath architectural design, we specify the more specific responsibilities. Generally, Michael and Leo focused on implementing the graphics library as well as its integration. Saahil, Sam, and Jason focused on the Genesis language / compiler stages / test platforms.

Project Timeline

The project timeline we aimed for is shown below.

Date	Milestone
September 26	Language proposal, Rough whitepaper
October 3	Language reference manual skeleton
October 16	Language reference manual completed
November 8	Compiler front end (parser, lexer)
November 27	Abstract syntax tree, data types, "hello world"
December 4	Link to graphics library, Codegeneration
December 11	Game features, Semantic checking
December 15	Regression testing, Visual testing
December 19	Sample games for demo
December 20	Final project report

Software Development Environment

Programming Language: C with SDL2 for graphics, Ocaml
Development environments: Vim, Emacs, and Sublime Text
Version control: Github

Project Log

As the log below shows, we have 255 commits along with several branches not shown over 3 months or so. Each member was heavily involved in the development process.

```
commit fcd782e5d0f26a59566ea005b0bae486665a43d6
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date:   Wed Dec 20 17:08:10 2017 -0500
```

Cleaning up codegen

```
commit 30f5242c84635985c95ab1c6df66afc732e263d7
Merge: 23ab09e 502dc67
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date:   Wed Dec 20 17:05:53 2017 -0500
```

Merge branch 'master' of github.com:samlouiscohen/Genesis

```
commit 23ab09ed3ef1f8a1b6afb36baad8b8cd0e7f2f3f
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date:   Wed Dec 20 17:05:42 2017 -0500
```

Adding author comments

```
commit 502dc67e83eae5582b153c63bee02329f002066a
Merge: 3ac2ab6 effe6d1
Author: Leo Stilwell <leostilwell@gmail.com>
Date:   Wed Dec 20 16:59:22 2017 -0500
```

demo version of snake

```
commit 3ac2ab6a2ed0609c84a9eefb72989b1859870e96
Author: Leo Stilwell <leostilwell@gmail.com>
Date:   Wed Dec 20 16:56:31 2017 -0500
```

final demo version of snake

```
commit 995b04e1007a95880aabf61a518b58e9854cbb42
Merge: af2e838 effe6d1
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date:   Wed Dec 20 16:54:07 2017 -0500
```

Merge branch 'master' of github.com:samlouiscohen/Genesis

commit af2e83887530b66af65fb706f18fedfa7fa86c86
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Wed Dec 20 16:53:11 2017 -0500

Finished tron demo

commit effe6d1e23565e4d3cb554b68afe30bce65ab45d
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 16:17:16 2017 -0500

added ignores to suppress unit

commit 754ec7b85d2ed520d3375073ba24b3866f89e3fd
Merge: 9424616 bea49f5
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 16:11:03 2017 -0500

merge error fix

commit 94246162c3f3848bc19584c14e8ccb0077c327d7
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 16:09:46 2017 -0500

Tweak gameover

commit 04e118c14f3ad3fe5cbab494a2b6c26d693854f4
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 16:07:55 2017 -0500

Fixed vtests and shortened certain display durations

commit d96d4ce7c7062046349f3b1db83d49e025e6599d
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 16:07:09 2017 -0500

Updated semantic checking for non-number type indexing and init with arrays

commit ca5eb3a62b6c86a5762b3107fdd37d5045ab258e
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 16:06:31 2017 -0500

Array tests for non-number type indexing and init

commit bea49f593d5e4b0ebbb405c9cc26c1d486adfe1e
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Wed Dec 20 14:18:55 2017 -0500

added food collision detection

commit 214d3d8b408a096d79154d20fa28b0cc333dc3c1
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 14:17:16 2017 -0500

Improved and cleaned up vtests with auto closing

commit d9e5be9b1d4bb65d8800f42a443089a9cebb0850
Merge: c31020b c95c811
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Wed Dec 20 13:50:58 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit c31020bc79344cedadf2e13a3b9f821216ca0e19
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Wed Dec 20 13:50:44 2017 -0500

added velocity tests

commit c95c8112b18bc9b1bf0222ba77d29c039ba60241
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Wed Dec 20 13:39:08 2017 -0500

added quit condition to test-random

commit 14911eca8a0d5a4796143695676b6d6ee445a4aa
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Wed Dec 20 13:35:54 2017 -0500

moved tests to vtest directory

commit abb107cd1aecc5e1908c9457def740f7bb49bbf6
Merge: 1da2507 8e985bf
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Wed Dec 20 13:34:17 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit 1da25073ac9cc78ca5bb356d38c0d1875894beb3
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Wed Dec 20 13:34:12 2017 -0500

tests

commit 8e985bfe7ebfaf17f23b74e8a7dbb7c78004081c
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Wed Dec 20 13:31:16 2017 -0500

finished property tests

commit 3a20af81a8108979009c5c01b338c570e423c9ad
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Wed Dec 20 13:21:57 2017 -0500

removed accidentally pushed files

commit cada22b2077ff5c8eab86b7a72007bf3b01c3b5e
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Wed Dec 20 13:20:09 2017 -0500

added property tests

commit 580516e20c380e5f91e5f352b58aeb14f53d1a02
Merge: e52b537 e6ce95b
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Wed Dec 20 12:22:54 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit e52b537ec636556a59e897dd5a1bf92bd15dc98d
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Wed Dec 20 12:22:37 2017 -0500

Fixed random seeding and setFPS

commit e6ce95b31ab6abc0f6dc7004bd912ff8eabc1724
Merge: 831f509 5008131
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Wed Dec 20 12:17:11 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit 831f50944c694923e894627ed054b280e704a7bf
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Wed Dec 20 12:16:43 2017 -0500

fix dx dy

commit 5008131b2381be716c443306a3df40d658e4fd19
Merge: 874ff70 08a7f9a
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 11:51:35 2017 -0500

Merge branch 'master' of [github.com:samlouiscohen/Genesis_2](https://github.com/samlouiscohen/Genesis_2)

commit 874ff7095b61c8049a4650957c23de77673fee07
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 11:51:29 2017 -0500

tweak

commit 08a7f9a05b8d741eb5d75f68a72e9fe6139f8289
Merge: 4c64773 ef5288f
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Wed Dec 20 11:47:35 2017 -0500

merge

commit ef5288fa1c6c750375e25f7e7a294c3c1842fe5e

Author: Michael Wang <michaelwangsu@gmail.com>
Date: Wed Dec 20 10:59:15 2017 -0500

signed code

commit 93c56d9a50d6ba17303885fc6d7db37e2b9632a3
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Wed Dec 20 10:52:12 2017 -0500

moved and fixed jumpman

commit 6d5fb10a55a31d210d63c43b05dcac012312f2f1
Merge: a555265 3c28c2b
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Wed Dec 20 10:42:38 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit a555265dc8f28cbbb323f6a8f4ea10545ec34f6f
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Wed Dec 20 10:42:28 2017 -0500

added setFPS

commit 3c28c2b13d593c1e2d63e1ee93c00355d9cb09b1
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Wed Dec 20 08:04:19 2017 -0500

Updating test after removing printf from library

commit fd4d22b6eafe33047b6d83296ed92c007e81fbac
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Wed Dec 20 08:02:43 2017 -0500

Nits

commit 2dc8f2a28a4ef536f388cee8456969481378e542
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Wed Dec 20 07:59:12 2017 -0500

Updating usage instructions

commit b5798ba6f1970d639dea609f0b641ae7e5bf0d9a
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Wed Dec 20 07:53:12 2017 -0500

Done with game of life

commit 4a7df5465f381baa58cb3f28b483121231e8de8f
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Wed Dec 20 07:52:54 2017 -0500

Unstyle code after executable name

commit e224da291949755ad896eee6246f9189b03a9421
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Wed Dec 20 07:52:29 2017 -0500

Color keyword collision bug fix

commit b134bdfd42a37a8fb8d7f34831481917d03e3e79
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Wed Dec 20 04:04:44 2017 -0500

Added fun build script, demo nits

commit b9fe027c734d1b63662625df31113263c31333fb
Merge: 11eca6d 1547210
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 03:31:05 2017 -0500

Merge branch 'master' of github.com:samlouiscohen/Genesis_2

commit 11eca6d05b087ea21e10d59de40d8f3d199cc511
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 03:30:56 2017 -0500

cleaned up

commit 1547210757d30d9440cfbaaef7603c2bd7bc3535
Merge: 78e27de c60804c
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Wed Dec 20 03:28:48 2017 -0500

Merge branch 'master' of github.com:samlouiscohen/Genesis

commit 78e27de141c4feda8625fe697c0a2f2ba3dcdaaa
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Wed Dec 20 03:28:30 2017 -0500

Removing non-demo apps

commit c60804cc18f4e6a88f906507b57046e6d7cbc3e8
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Wed Dec 20 03:25:31 2017 -0500

Fixing accidental bugs

commit 9856495c7e4919600aa5f08bea5984c2cb9d0cc7
Merge: 203769c ca06985
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Wed Dec 20 03:23:47 2017 -0500

Merge branch 'master' of github.com:samlouiscohen/Genesis

commit 203769cac9163394230fa4926e94d7d4b28459bc

Author: zhaosjason <zhao.s.jason@gmail.com>
Date: Wed Dec 20 03:23:41 2017 -0500

Adding test

commit ca06985b14af08510e69d85f0e3af6c7faa4c1e4
Author: Jason Zhao <zhao.s.jason@gmail.com>
Date: Wed Dec 20 03:23:34 2017 -0500

Formatting

commit 93cc23f6685c020eadfac8c4a77e808d2c93f208
Merge: 63fb4ab 32beb16
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 02:26:13 2017 -0500

Merge branch 'master' of github.com:samlouiscohen/Genisis_2

commit 63fb4ab5e3a92b9a1f54ac724c1fdcfc01854306
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 20 02:26:05 2017 -0500

Finished Snake Game.

commit 32beb16d145bbde20e668f86cdb9d7385e80a092
Author: Jason Zhao <zhao.s.jason@gmail.com>
Date: Tue Dec 19 23:34:12 2017 -0500

Sematic check for void arrays

commit 28b09c8d688024eaa2543bd71ba363fb9a21047d
Author: Jason Zhao <zhao.s.jason@gmail.com>
Date: Tue Dec 19 22:29:59 2017 -0500

Updating installation instructions for osx and updating testall.sh link

commit 31930b3d13904e1c5cc2823eefb4968cd4262399
Merge: b6776f6 1af5a50
Author: zhaosjason <zhao.s.jason@gmail.com>
Date: Tue Dec 19 21:13:18 2017 -0500

Merge branch 'clib_formatting'

commit 1af5a50524116370370071dbdd00a1e96ec1a877
Author: zhaosjason <zhao.s.jason@gmail.com>
Date: Tue Dec 19 21:10:21 2017 -0500

Fixing collision bug

commit a5513eb733d5c031d5fbd2f2c8e807b5dddcd9e
Author: zhaosjason <zhaosjason@gmail.com>
Date: Tue Dec 19 21:06:47 2017 -0500

Fixing syntax errors

commit b6776f646fee22486c54182020347556b76ed60e
Author: zhaosjason <zhao.s.jason@gmail.com>
Date: Tue Dec 19 20:49:26 2017 -0500

Added array delete functionality

commit 5745749ba5340bbcd6cb95d6e4bf99db282ee256
Author: Jason Zhao <zhao.s.jason@gmail.com>
Date: Tue Dec 19 20:45:25 2017 -0500

dx dy bug fix

commit 29d42bf4355be55507f768210600ba052936e225
Author: Jason Zhao <zhao.s.jason@gmail.com>
Date: Tue Dec 19 20:42:01 2017 -0500

Formatting

commit 4c647739900dbf2b0e20fba8285295691a451c55
Merge: 1eec7b6 0048278
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Tue Dec 19 20:40:07 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit 1eec7b681f0d8af8d69bb81e5e1682b71a3e699a
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Tue Dec 19 20:38:47 2017 -0500

added missing dx dy functionality

commit a402e2a5bbfb488bcc4a850d8509814e994b490f
Author: Jason Zhao <zhao.s.jason@gmail.com>
Date: Tue Dec 19 20:28:00 2017 -0500

Initial work on delete arrays

commit 0048278bd3795522f8d842bcc5802584d6ac2032
Author: Sam Cohen <samsam489@gmail.com>
Date: Tue Dec 19 18:50:25 2017 -0500

Makefile nits

commit 2b90d711da3ee60c3be20a8c86f04057c1f223fb
Author: Jason Zhao <zhao.s.jason@gmail.com>
Date: Tue Dec 19 05:09:00 2017 -0500

Formatting nits

commit 3a5d95921e80f8bea0398cb3f31ca8e215cc4ace
Author: zhaosjason <zhao.s.jason@gmail.com>

Date: Tue Dec 19 05:04:48 2017 -0500

Added declaration / assignment mixing

commit 76de04e61717c599d2804d9d9bfb0f9aacbafa01

Author: Michael Wang <michaelwangsu@gmail.com>

Date: Mon Dec 18 17:24:00 2017 -0500

added quit() function

commit 4cf6f05e5d969f37312530bc746b928c5656bdb2

Author: Michael Wang <michaelwangsu@gmail.com>

Date: Mon Dec 18 16:58:19 2017 -0500

added jumpman game

commit 26b4ec3ad01875f7a31c9ed5b9f88522b32afdcb

Merge: 5d37f9a 27be181

Author: Michael Wang <michaelwangsu@gmail.com>

Date: Mon Dec 18 14:38:46 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit 5d37f9a49740d8ee1e98161454c68fee8611ceac

Author: Michael Wang <michaelwangsu@gmail.com>

Date: Mon Dec 18 14:38:35 2017 -0500

updated key press functions

commit 27be181a6c4fd1c493a4041fb8b052effc5c3b99

Author: Leo Stilwell <leostilwell@gmail.com>

Date: Mon Dec 18 13:56:10 2017 -0500

updated snake demo

commit d43048a65e4da6dbb1f4e677d536cbe8309f2d53

Author: zhaosjason <zhaos.s.jason@gmail.com>

Date: Sun Dec 17 23:23:31 2017 -0500

Adding modulo operator

commit e85e7aa2ed5ea617e8f9d908ce8bd7a5284f3b5a

Author: zhaosjason <zhaos.s.jason@gmail.com>

Date: Sun Dec 17 22:36:41 2017 -0500

Cleaning up repo

commit 8d031da174c0c11465fc9c80e11c7f0cfbbfe71f

Merge: 6cb3e25 511d9e1

Author: zhaosjason <zhaos.s.jason@gmail.com>

Date: Sun Dec 17 22:34:33 2017 -0500

Merge branch 'master' of [github.com:samlouiscohen/Genesis](https://github.com/samlouiscohen/Genesis)

commit 6cb3e2561ff240a1771745a7dad7d386d4537f68
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sun Dec 17 22:34:27 2017 -0500

Nits

commit 511d9e15f83835a78bddba694dd15d74ff80e128
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sun Dec 17 22:18:51 2017 -0500

fixed some bugs in code. much of update is commented out so we can isolate getting the movement working

commit 765518da33cae1c6a10eaeeca30e261ce4f9b24f
Merge: bce78f2 eb0057d
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sun Dec 17 21:51:38 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit bce78f2a32d7b22f576957b5a2e81a215eb27b36
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sun Dec 17 21:51:33 2017 -0500

added demo directory

commit eb0057d1bf2ce33297caaeed8c5ec5526f9d9119
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sun Dec 17 21:38:46 2017 -0500

Renaming test

commit d645901fb0b740eb6e9d43e72abdd6a1cf8bc34c
Merge: 98a40e4 3ccc231
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sun Dec 17 21:37:53 2017 -0500

Merge branch 'array_tests'

Conflicts:
semant.ml

commit 3ccc2318a0e49368e4c9f17556eece8a1dadbbec
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sun Dec 17 21:33:41 2017 -0500

Added global array tests, fixed global array bug.

commit 98a40e4e73f441a5ac07fc47fa0a9709baf7a9f7
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 22:09:27 2017 -0500

Remove unnecessary comments

commit 2576b54c59e2cd7553fd0855a18d26b2f9772f00
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 22:08:28 2017 -0500

Edit sample program

commit bb4f87f0ed93d703eb77a02a842d253f5d68f0b9
Merge: e37c2c6 2ac1aae
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 22:07:00 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit e37c2c6dd8758dd8b6c2b25f001ecc9270074fee
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 22:06:45 2017 -0500

Fixed warnings on semant.ml

commit 2ac1aaefc09ddf0d83ce01e5ea38e06fd6dd826
Author: Sam Cohen <samsam489@gmail.com>
Date: Sat Dec 16 22:04:18 2017 -0500

added array tests

commit 5f31c867c88b771baf5e581ffb5e8ce23d2b89aa9
Merge: acdd343 2904a60
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 22:00:37 2017 -0500

Fixed semant.ml

commit acdd343aad9baa46d4c8315c79b1b3bd0a482a6d
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 21:58:03 2017 -0500

Implement static semantic checking for Color and Cluster

commit 2904a60b76334d291befa4adbf2032d7aee5b5ff
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sat Dec 16 21:44:24 2017 -0500

Fixed compilation warnings

commit 9a503146bdc6cd80c7d27bf55d2082be1e21bbb3
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 20:46:42 2017 -0500

added missing getY function

commit 132d2b27e23fa82c2de9245524f899d4beaeba1d

Author: Michael Wang <michaelwangsu@gmail.com>
Date: Sat Dec 16 19:24:35 2017 -0500

removed yacc files

commit 1b2432d810369ee601cd33b51eadd392c27db3ef
Merge: 5a1e298 bb422c1
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Sat Dec 16 19:21:29 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit 5a1e29858f197bf7a2f6208b13995b16c1f7e99f
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Sat Dec 16 19:21:25 2017 -0500

Removed sr conflict for collision

commit bb422c1d3e0ea908ff2714d987e16c7b46b7ca1e
Merge: 87a52f1 d165392
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 19:16:21 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit 87a52f1b16e4f2a78a513de7b3fccf10ed0c60f6
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 19:16:07 2017 -0500

Add static semantic checking for init and update

commit d165392a4c633ce3d845973c8d4e08cfabfc5fb4
Author: Sam Cohen <samsam489@gmail.com>
Date: Sat Dec 16 19:04:47 2017 -0500

fixed test mismatch

commit 4cf69f641dc32732e193bd8b6f79d81e0d74f8c6
Merge: b70a944 f8420b1
Author: Sam Cohen <samsam489@gmail.com>
Date: Sat Dec 16 18:56:31 2017 -0500

Merge branch 'master' of [github.com:samlouiscohen/Genesis_2](https://github.com/samlouiscohen/Genesis_2)

Conflicts:
semant.ml

commit b70a944febb830e83c83f2095e173c52592892e1
Merge: e0899be 091247e
Author: Sam Cohen <samsam489@gmail.com>
Date: Sat Dec 16 18:55:19 2017 -0500

Merge branch 'implementing_arrays'

Conflicts:

tests/test-array-init.god

commit 091247eca9d9aa424dbaa18fb6ea36d1936a20b6

Author: Sam Cohen <samsam489@gmail.com>

Date: Sat Dec 16 18:52:11 2017 -0500

Merge master into implementing_arrays

commit f8420b1df0df4cb225de2316ec6b1d361e768a58

Merge: ca88a62 e832475

Author: saahil9jain <saahil9jain@gmail.com>

Date: Sat Dec 16 18:41:07 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit ca88a6212891df81e57f2e3a2fd16ea2952a2e2b

Author: saahil9jain <saahil9jain@gmail.com>

Date: Sat Dec 16 18:40:51 2017 -0500

Add string test: strings work :)

commit 2c05676ba3f9bd75f7e580357bc096db7e598560

Author: Sam Cohen <samsam489@gmail.com>

Date: Sat Dec 16 18:35:17 2017 -0500

Allow for passing arrays into functions

commit e8324753f658fe1ee6249815af9975a0ffa9251f

Merge: 31d8911 3d0654b

Author: Leo Stilwell <leostilwell@gmail.com>

Date: Sat Dec 16 18:26:10 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit 31d89119a51098d45cc8858bc0e3b2a4c380c64a

Author: Leo Stilwell <leostilwell@gmail.com>

Date: Sat Dec 16 18:26:02 2017 -0500

added draw flag getter/setter

commit 49f7be1c4d706d23b78e9e54f14247102a2b07e9

Author: Leo Stilwell <leostilwell@gmail.com>

Date: Sat Dec 16 18:25:18 2017 -0500

added tests for collision and random function

commit 3d0654b16a016a03f62a367fd9dd5029448a408e

Author: Michael Wang <michaelwangsu@gmail.com>

Date: Sat Dec 16 17:52:37 2017 -0500

Connected deleteCluster function

commit de985c52e5cd931252d234040f1856278fdb54af
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Sat Dec 16 17:42:17 2017 -0500

made collision operator

commit db4dd2d011c6afde403ef6cb0ebd6f63da6227f6
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Sat Dec 16 17:41:54 2017 -0500

made collision operator

commit ca575d560d79304085a31f919f15e82e81da7762
Merge: 6adae44 994437e
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Sat Dec 16 17:22:22 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit 6adae448423d5b5dc20496cb8e5ac8da68610da7
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Sat Dec 16 17:21:50 2017 -0500

added setter and getter in codegen and semant for draw

commit 994437eff737f025ddc42017e754b084d89d3026
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 17:15:32 2017 -0500

updates with drawin functionality and a test for a cluster moving

commit a2268ee627532342ec66bef95f096f263e001065
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 16:51:13 2017 -0500

fixed clusters not being drawn

commit 0f32ccaa6329790c47a3c70f2de0dd8aa940e9da
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 16:41:32 2017 -0500

updated remaining tests

commit 7d8854804d7e55d0860c428f3d9967596f80d8e5
Merge: 7251f21 f8d69fa
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Sat Dec 16 16:32:45 2017 -0500

Merge branch 'master' into removeSRConflicts

commit 7251f21510fc2a17d74d74172ef6839deba5bd33

Author: Michael Wang <michaelwangsu@gmail.com>
Date: Sat Dec 16 16:29:17 2017 -0500

Removed error in semant.ml

commit f8d69fa54a6b564489428b36557f4768e14980da
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 16:26:26 2017 -0500

updated tests with new syntax

commit 0cd62fa9550945ae18fb1c0845ebbb424e6b8f97
Author: Michael Wang <michaelwangsu@gmail.com>
Date: Sat Dec 16 16:25:55 2017 -0500

removed sr conflicts

commit fd0c9e34ed7e223ef5635d094799545adfe2fda3
Merge: 058d8d3 9a7bdb6
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 15:28:09 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit 058d8d3e31997f40a4ff57b7143546c5a7fbc239
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 15:27:49 2017 -0500

added drawset and draw function

commit 9a7bdb6cbea563ffeff04d70ace536061b69a02a
Merge: c2dc798 526a0fd
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 15:26:00 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit c2dc79859f2dba0451565e117b678c33421b9c6e
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 15:25:45 2017 -0500

Update makefile

commit 13fb53ca55367b646bf3cc48ecdc3053cd7b0046
Merge: 8a9dcc1 526a0fd
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Sat Dec 16 15:22:39 2017 -0500

merged ast

commit 8a9dcc1f87bf72e277761b5aaf209f381fc284a5
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Sat Dec 16 15:21:50 2017 -0500

starting removal of sr conflicts

commit 5011aaa1726f53325ab38a73a9060f8785274583
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Sat Dec 16 15:17:59 2017 -0500

Adding property assignment

commit 526a0fd28a7ecc523e2f75f4803f4f4df31dfef2
Merge: f68fae2 14d5fba
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 15:07:59 2017 -0500

fixed conflicts

commit f68fae2619ee1fdd7b9e7be368bd2560d8e1e0be
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 15:05:45 2017 -0500

added function to check if a SDL_Rect is within another SDL_Rect. can be used to detect game winning condition

commit 14d5fbac2dc4c370110530f03d26aa0bc04d6ccf
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sat Dec 16 14:58:20 2017 -0500

Nits

commit 8bc5a0480394df2823451cefc23dd0e8a831f5a1
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sat Dec 16 14:51:17 2017 -0500

Cleanup

commit d331c9e5ea409e2fc545470fc4b44c1a0a4260b0
Merge: 9dc7010 303d81d
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sat Dec 16 14:48:30 2017 -0500

Merge branch 'master' of github.com:samlouiscohen/Genesis

Conflicts:
ccode/genesis.c

commit 9dc70104895ea1655d44abcb5bc71776e0b12a66
Merge: 327d5ff 4825cea
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sat Dec 16 14:46:46 2017 -0500

Merge branch 'master' of github.com:samlouiscohen/Genesis

Conflicts:

```
.gitignore
ast.ml
ccode/genesis.c
codegen.ml
parser.mly
scanner.mll
```

```
commit 303d81dd9c1cec53e0d212b6837f2c6322414ddc
Merge: 95ceb32 1a5cd4d
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 14:42:31 2017 -0500
```

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

```
commit 95ceb3275ddddd71061597751361f7f2b0602c7b4
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 14:42:17 2017 -0500
```

add init and update

```
commit 1a5cd4dcebe4c7d44c1d25d84af4a69be4a812fd
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 14:39:58 2017 -0500
```

fixed typo

```
commit 327d5ff75ef9d355ea2ca732e2fe9d7a12e7a87e
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sat Dec 16 14:38:24 2017 -0500
```

Updating build process for Linux

```
commit aaa705c6c9a1ddf325c12f06787cbd02d911aa06
Merge: 0651396 c86852c
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 14:35:41 2017 -0500
```

Resolve test functionality on master

```
commit 4825cea28ef3f44091b03bdb4922905c7b1c87f8
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 14:34:45 2017 -0500
```

split setXY into setX and setY

```
commit c86852c333f554b547e9641db78e5580916d5920
Merge: 936b946 5a1ba62
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 14:31:09 2017 -0500
```

Merge branch 'addClusters' of <https://github.com/samlouiscohen/Genesis> into addClusters

```
commit 936b946c0d711a47837d1943245969c121352c7a
```

Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Dec 16 14:30:56 2017 -0500

Make cluster init / visualization work with update and init

commit 06513969d8770d938d99afaa15a7790059cb8c69
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 14:28:17 2017 -0500

fixed misc compilation errors

commit 3fa9d7bf5d9240d9424071fefff951d55f03d24d
Merge: e0899be 5a1ba62
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 14:26:17 2017 -0500

merge with addClusters

commit b26cc9282224daf4ac188c3f4de449fcbaeb9b99
Merge: e0899be 0efe860
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sat Dec 16 14:14:01 2017 -0500

Merge branch 'implementing_arrays'

commit 0efe8609711b48c18e4f56c6ea1ae11060cd7e8e
Author: Sam Cohen <samsam489@gmail.com>
Date: Sat Dec 16 14:12:37 2017 -0500

Nits

commit 5a1ba628b9007157a4379c5f806405d4e3c3cbc4
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 14:12:23 2017 -0500

updated function name for setColor

commit 27a3426a866e22df5e1a128e4d70038e02f94b52
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sat Dec 16 00:32:43 2017 -0500

Finished with arrays

commit e5ceb2eb2dea3a8680ce003b7c9efb5764c2bb1a
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 00:21:19 2017 -0500

fixed all warnings

commit d7070c5d851a15bfa62c1214bf88d9a3a4afdc95
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 00:18:50 2017 -0500

fixed warning

commit 3f348168bd8e922db045df0357e5cbbb93f1f654
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 16 00:17:04 2017 -0500

t

commit 48a8ade729f829ca0453a03d40c4bda4b8520b09
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Sat Dec 16 00:13:22 2017 -0500

added properties for not x and y values

commit ad9c8190b176c1b018d46f4911b7bd58cd55795e
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Fri Dec 15 23:34:25 2017 -0500

Added property access for x and y into ast, semant, and codegen

commit c873dedb325e41a13ad3b1258b0a02508b989116
Merge: c41cebe 7111b23
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Fri Dec 15 23:33:20 2017 -0500

Merge branch 'addClusters' of <https://github.com/samlouiscohen/Genesis> into addClusters

commit c41cebed2eaa1aef13c70354bec0a4ff2b6fb060
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Fri Dec 15 23:32:50 2017 -0500

Added property access for x and y into ast, semant, and codegen

commit 7111b232705df7b07aa6644a8497c9f2d4040229
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 23:24:50 2017 -0500

added collision function

commit c23aec5f2e5b672cf9bd6c3046dff46a819171a1
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 22:36:12 2017 -0500

implemented cluster getters/setters

commit fcc531d9cb80e984a05480b67246e052bd1322f9
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 21:37:18 2017 -0500

updated test-startgame

commit c8b020c3c05d461c466f327008f0f53499f47227
Merge: cf1cd45 6531891

Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 21:29:19 2017 -0500

Merge branch 'addClusters' of <https://github.com/samlouiscohen/Genesis> into addClusters

commit cf1cd453b237ef8edc1ab7ae7dabc61b983aa39d
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 21:29:06 2017 -0500

updated tests

commit 653189119bfaf752bdf6289a5abf5b691044c30c
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Fri Dec 15 21:18:20 2017 -0500

fixed some errors

commit eeeec5f29c90daf86c6ffd5ddfb632661ccd3475
Merge: 291fd1d c5a7166
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Fri Dec 15 21:11:44 2017 -0500

resolved merge conflicts

commit 291fd1db15dee9d89c06b6e535ce24441ef7636c
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Fri Dec 15 21:09:22 2017 -0500

fixed semantic error in random codegen

commit c5a71662618af47dad43ae4dd77f933698a7938c
Merge: 40fd9ca 9f90db6
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 19:35:48 2017 -0500

merge

commit 40fd9cae15da66c64999a5483ea38fd34d2f3363
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 19:34:09 2017 -0500

can call startgame from a genesis program to draw a shape

commit a7b6ce7164d680877297bf326be02cae5e49ae8c
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Fri Dec 15 19:24:49 2017 -0500

Added framerate, frame counter, and random generator

commit 9f90db69876aa6e8a3d741481217b34dc21c6874
Author: saahil9jain <saahil9jain@gmail.com>
Date: Fri Dec 15 19:18:41 2017 -0500

test updates

commit e0841c1753aab56387e00112e6ba72a78b0b23d6
Author: saahil9jain <saahil9jain@gmail.com>
Date: Fri Dec 15 19:14:18 2017 -0500

test case for cluster new

commit 88d74cbd3969e7fbe49aa4863e01a0086cd92dc3
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 18:25:37 2017 -0500

added code to free clusters in hash in close function

commit e90256fac9d026a5538756c0025a0757e8276ac6
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 18:04:45 2017 -0500

updated header files

commit d9eaaea1ac21351088c2bafef8e7b61a24be8012
Merge: 29bccbf 22d7067
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 18:01:07 2017 -0500

Merge branch 'addClusters' of <https://github.com/samlouiscohen/Genesis> into addClusters

commit 29bccbf18690cbc2236a74c5ea13ca02ca6eef6f
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 18:01:01 2017 -0500

added cluster add and remove functions

commit 22d70676df8541141a9022eb7401a31aae8a2884
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Fri Dec 15 18:00:52 2017 -0500

Checked ast files with creating clusters

commit e0899be586449b4f365386326c53d5451144b980
Merge: 6dd816c 0a747e4
Author: saahil9jain <saahil9jain@gmail.com>
Date: Fri Dec 15 17:48:29 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit 6dd816c463829fb109fe32ec757c8c28a8942a94
Author: saahil9jain <saahil9jain@gmail.com>
Date: Fri Dec 15 17:48:15 2017 -0500

sample program update

commit 0a747e41826011c549f6247dbfd70f7a15016b49

Merge: 539d3c6 8057e4a
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 16:58:21 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit 539d3c62e28d3c3765f634430463b73cbb426708
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Fri Dec 15 16:57:16 2017 -0500

added cluster add and remove functions

commit 8057e4a7bd957592c053c9c25b9a99afef42ae5f
Author: saahil9jain <saahil9jain@gmail.com>
Date: Fri Dec 15 16:40:13 2017 -0500

Create sample files / linking

commit e336ed40cd89b4466592fe9b2648b91897f5b7d6
Author: Michael Wang <michael@dyn-160-39-234-62.dyn.columbia.edu>
Date: Fri Dec 15 16:27:29 2017 -0500

added newCluster just for testing codegen

commit 8bbd1c1dac0f0862892cec2f26fdd0b152d34e97
Author: saahil9jain <saahil9jain@gmail.com>
Date: Fri Dec 15 16:15:02 2017 -0500

Update visual testing suite to work

commit 0d4d0421eb8fd6d01abee2e58fe649e3e371a5d8
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Fri Dec 15 15:13:04 2017 -0500

Updating readme again. Usage notes, and testing

commit ef9c5dd828be77b6c077e0e20f3e61ce71b762b8
Merge: 05a1c6e 1cb316b
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Fri Dec 15 14:53:22 2017 -0500

Merge branch 'master' of [github.com:samlouiscohen/Genesis](https://github.com/samlouiscohen/Genesis)

commit 05a1c6e3a09631e7daf7c5d221cb2b20755ee8dc
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Fri Dec 15 14:53:12 2017 -0500

Adding endorsements

commit 1cb316bc7c8b68ae2eafeea92bb058768566513a
Merge: c4a5ff4 4c68b11
Author: saahil9jain <saahil9jain@gmail.com>
Date: Fri Dec 15 14:40:51 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit c4a5ff473a4aa87c8bc77c6ff161f0000be6b477
Author: saahil9jain <saahil9jain@gmail.com>
Date: Fri Dec 15 14:40:35 2017 -0500

Implement god functionality

commit 97289ac67ed383ecaf84d1bad0266a90c4c8282b
Author: Sam Cohen <samsam489@gmail.com>
Date: Fri Dec 15 13:51:00 2017 -0500

Implementing arrayInit

commit 4c68b11d8c8bcafa480d01094f099a0a2a5513ee
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Fri Dec 15 00:00:32 2017 -0500

Updated dependencies

commit 2104af8c5423999b06db6c2c9e7dc6cd8e110078
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Thu Dec 14 23:59:03 2017 -0500

Removing unneeded files

commit 9ff7457244159032349579b366f55b94dc629bd5
Merge: e92fe47 e066f7e
Author: Jason Zhao <zhaos.s.jason@gmail.com>
Date: Thu Dec 14 20:05:05 2017 -0500

Merge pull request from [samlouiscohen/keyInput-test](#)

Resolved all merge conflicts. Master is now up-to-date.

commit e066f7ea48a21c5a70916745ddd3089894db9b7
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Thu Dec 14 19:58:39 2017 -0500

Nits

commit 14fd54e060d794d49fd25394a5400ec20fbd2677
Merge: 3242e7a bf3b350
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Thu Dec 14 19:53:13 2017 -0500

Merge branch 'michaelTest' into keyInput-test

commit bf3b3503dec3b535fdec719f8bc6586f1d1585de
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Thu Dec 14 19:44:58 2017 -0500

Cleaning up repo

```
commit 7e6f449d90f5efc8342e86682d5da61670f26ecf
Merge: 20ec8ab e92fe47
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Thu Dec 14 19:43:50 2017 -0500
```

Merge branch 'master' into michaelTest

```
commit 20ec8ababd027eb116d535d6537bf7ce182879c6
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Thu Dec 14 19:13:15 2017 -0500
```

fixing repo:

q

```
commit 3242e7a663a5644bdfad4a717bad9b42c7bff015
Merge: 515de60 f111e0a
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Thu Dec 14 18:59:24 2017 -0500
```

merge with michael branch

```
commit f111e0a1a43ccceea9a468203dd0383e970dbede
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Thu Dec 14 18:53:16 2017 -0500
```

initial clusters working

```
commit e92fe4715433de9afd8c011a5d149d584a6da593
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Thu Dec 14 18:32:46 2017 -0500
```

Fixed parsing error. Arrays can be declared now. Some initial work on array initialization, access, and assignment.

```
commit 515de6086343bebc41d070c723def8fa5243e9d7
Author: Michael Wang <michael@dyn-160-39-234-223.dyn.columbia.edu>
Date: Thu Dec 14 16:46:19 2017 -0500
```

Finished key input

```
commit 4f0262f5fd67e7d1b834bf847e3707f4bc6e40dc
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Thu Dec 14 16:04:44 2017 -0500
```

added UThash

```
commit c68ea9f791ad2077e5ed8f56e7b485ec01325967
Author: Michael Wang <michael@dyn-160-39-235-228.dyn.columbia.edu>
Date: Wed Dec 13 22:08:39 2017 -0500
```

added c code for key presses

commit 52d8fcf81175192485a2b83098cdb44cf2e1e056
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Wed Dec 13 21:53:02 2017 -0500

added clusters to ast, parser, scanner and codegen. modified makefile for c code

commit b3670fdb633d744d8980e3363a5e87d33d20d701
Merge: 541e859 7350e37
Author: zhaosjason <zhao.s.jason@gmail.com>
Date: Wed Dec 13 21:16:50 2017 -0500

Merge branch 'master' into adding_arrays

Conflicts:

- ast.ml
- codegen.ml
- parser.mly
- semant.ml

commit 040f8992f3024604843c87eca1594355b2cb93b9
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Wed Dec 13 21:08:22 2017 -0500

added Cluster to AST

commit 7350e3758804a132ae6258cbb398dc299a5d51c3
Merge: e21b4d5 eeb60bf
Author: zhaosjason <zhao.s.jason@gmail.com>
Date: Wed Dec 13 20:18:48 2017 -0500

Merge branch 'master' of github.com:samlouiscohen/Genesis

commit 541e85904b719ef8b5cc0dc98767fedd4a0f14ee
Author: Sam Cohen <samsam489@gmail.com>
Date: Wed Dec 13 20:18:38 2017 -0500

reverted to old global_vars definition

commit e21b4d5db3d018d83472bb8d57f3ce8b042ce861
Author: zhaosjason <zhao.s.jason@gmail.com>
Date: Wed Dec 13 20:18:01 2017 -0500

Updating genesis building

commit 503d229f2aa19529da844e73a7783d435325801c
Author: Jason Zhao <zhao.s.jason@gmail.com>
Date: Wed Dec 13 19:53:33 2017 -0500

Updating array stuff

commit bce0b9a790722dd1ab1b0d655a11a9121105f2d7
Author: Michael Wang <michael@dyn-160-39-235-228.dyn.columbia.edu>

Date: Wed Dec 13 19:28:14 2017 -0500

Added startgame

commit c32091b1b3fb8e43c132f098a0cf1afa9463c069
Author: Michael Wang <michael@dyn-160-39-235-228.dyn.columbia.edu>
Date: Wed Dec 13 17:51:09 2017 -0500

colors done

commit 035f513f3bb716c9f3e97dfcd2564be30922b62a
Author: Michael Wang <michael@dyn-160-39-235-37.dyn.columbia.edu>
Date: Tue Dec 12 17:26:36 2017 -0500

Color works

commit 1e4156539d581c451c50b9d505687f13cc179eac
Author: Michael Wang <michael@dyn-160-39-235-252.dyn.columbia.edu>
Date: Mon Dec 11 23:45:22 2017 -0500

passing structs works

commit eeb60bf36c8a9b2464bdfb9bd7a2cd4e44cfd46
Author: Jason Zhao <zhaos.jason@gmail.com>
Date: Sat Dec 9 19:44:18 2017 -0500

Updating gitignore

commit 79ad6a5cc849d79c02c83aab376c2990f105dfdd
Author: Jason Zhao <zhaos.jason@gmail.com>
Date: Sat Dec 9 19:43:56 2017 -0500

Removing binary

commit 7b55a4bbe49e0afcb16c73b71e0c34d0dfbce07a
Merge: 0097ffa d44f075
Author: Sam Cohen <samsam489@gmail.com>
Date: Fri Dec 8 21:54:44 2017 -0500

Merge branch 'master' of github.com:samlouiscohen/Genesis_2

commit 0097ffa8fe3ba7db87cf6406370770cecf8b4f7d
Author: Sam Cohen <samsam489@gmail.com>
Date: Fri Dec 8 21:54:33 2017 -0500

Working on Arrays

commit d44f0759cded9c34d1f1a443e3183e5ae3efb93c
Author: saahil9jain <saahil9jain@gmail.com>
Date: Fri Dec 8 19:04:30 2017 -0500

Comment out Sam's silliness

commit b50ed5bac053de919f979cf60df9517f476a2be2
Merge: 70bba97 f8c05b8
Author: saahil9jain <saahil9jain@gmail.com>
Date: Fri Dec 8 19:03:18 2017 -0500

Merge branch 'SDL_TEST'

commit f8c05b869a296e13ec46cdbfd52f03262d11bd9f
Author: saahil9jain <saahil9jain@gmail.com>
Date: Fri Dec 8 18:54:19 2017 -0500

Adapted code for MAC

commit 70bba97ea60e2db9cb6609ee7518b1728cb17a7a
Author: Sam Cohen <samsam489@gmail.com>
Date: Fri Dec 8 17:47:55 2017 -0500

added single line comments

commit 59e5560ae51183b3c60b044b58f9361880e9dc82
Author: Sam Cohen <samsam489@gmail.com>
Date: Fri Dec 8 17:36:06 2017 -0500

saving old attempt to add arrays

commit 9eb18acb7ae1153fc033c8700c6b6c914188ec73
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Sun Dec 3 01:08:49 2017 -0500

Nits

commit 1bf351661455f6de1eb138e095fe4c1e7009bc3a
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Sat Dec 2 20:27:23 2017 -0500

Fixed Ubuntu bug

commit 20f3f842e3e8c32f90fdb172b95a01a89a1a01
Author: Michael Wang <michael@dyn-160-39-235-166.dyn.columbia.edu>
Date: Sat Dec 2 20:05:48 2017 -0500

Nits

commit 26d5696d3d0c2722420ccb36954e72ee9460723c
Merge: 853b776 70d46c9
Author: Michael Wang <michael@dyn-160-39-235-166.dyn.columbia.edu>
Date: Sat Dec 2 20:04:36 2017 -0500

Merge branch 'master' of <https://github.com/samlouiscohen/Genesis>

commit 853b776c95fe03d98d8a33ae6e721ff8e90ec5ea
Author: Michael Wang <michael@dyn-160-39-235-166.dyn.columbia.edu>
Date: Sat Dec 2 19:59:55 2017 -0500

Linked LLVM to C SDL Calls, updated test suite

commit e2ce78fa528d8156c3f1cf9315f014d416b91768
Author: Sam Cohen <samsam489@gmail.com>
Date: Sat Dec 2 18:35:52 2017 -0500

added method to build uniform arrays and redefining global var values before assignment

commit 70d46c9ccb89b2956964736ffc388b2f3099f5f0
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Fri Dec 1 01:07:17 2017 -0500

Added string test

commit 2e068904ab27d5e40b8a19107d8bf96ed1516085
Author: saahil9jain <saahil9jain@gmail.com>
Date: Thu Nov 30 19:10:46 2017 -0500

Fix test typo

commit 818fd33bd63b5aa419bc1bbdc4571a7dd2cd8660
Author: saahil9jain <saahil9jain@gmail.com>
Date: Thu Nov 30 19:09:41 2017 -0500

Implemented strings

commit af6992cc0f532cbaccb33a77a0eda48b3e9cf3d4
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Tue Nov 28 17:22:56 2017 -0500

Finished float casting, updated tests

commit 2b41bfe310b8686ae08991645c95e2da7312040f
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Mon Nov 27 23:26:50 2017 -0500

Fixed boolean/float logic error. Makefile nits

commit 18600b7c9dfe47a1236f12480d085e17863f98e9
Merge: a69838d 87cdcba
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Mon Nov 27 23:06:36 2017 -0500

Merge branch 'master' of github.com:samlouiscohen/Genesis

commit a69838d9f439f2199e21cc7b093304e94ac76cba
Author: zhaosjason <zhaos.s.jason@gmail.com>
Date: Mon Nov 27 23:06:15 2017 -0500

Finished float semantic checking and implementation

commit 87cdcba65cf9717f4f28d4cd7d0665d4086b368

Author: Leo Stilwell <leostilwell@gmail.com>
Date: Mon Nov 27 22:47:02 2017 -0500

added color to codegen. need to add to ast still

commit e3fbdc365cc738a24ac8f9059de2a781e236f457
Merge: 6383615 ada4c75
Author: Sam Cohen <samsam489@gmail.com>
Date: Mon Nov 27 22:37:19 2017 -0500

Merge branch 'master' of github.com:samlouiscohen/Genesis_2

commit 6383615ccb42ca0a0f3db9a540d6b7a452d9f8ee
Author: Sam Cohen <samsam489@gmail.com>
Date: Mon Nov 27 22:37:14 2017 -0500

assignment type checking

commit ada4c753206ef50ae9d6f793f41965f318eab573
Author: Leo Stilwell <leostilwell@gmail.com>
Date: Mon Nov 27 22:21:33 2017 -0500

header file for genesis c code

commit 32d4ea1b948e1b57196467f695b5ec8f9e301110
Merge: 7784c27 fb2ab5d
Author: Jason Zhao <zhaos.jason@gmail.com>
Date: Mon Nov 27 21:08:51 2017 -0500

Merge branch 'master' of github.com:samlouiscohen/Genesis

commit 7784c27bef927a50cc69f115efce2e634bda9aba
Author: Jason Zhao <zhaos.jason@gmail.com>
Date: Mon Nov 27 21:08:38 2017 -0500

Just some formatting

commit fb2ab5dace99bfe7edc6dff6ad10e744586391a9
Author: Michael Wang <michael@dyn-160-39-234-79.dyn.columbia.edu>
Date: Sun Nov 19 18:32:09 2017 -0500

Added c code

commit 3e8077f5838bc249e4f46346b9b473f08090f8f6
Author: Jason Zhao <zhaos.jason@gmail.com>
Date: Sun Nov 19 18:10:58 2017 -0500

Nits

commit 0c5de3831ef6faefad30dbb309c5c17b5bdfe655
Author: Jason Zhao <zhaos.jason@gmail.com>
Date: Sun Nov 19 18:09:39 2017 -0500

Updating readme

commit 4590ecc899ba876f998d057e133cd01897429a95
Author: Jason Zhao <zhaos.jason@gmail.com>
Date: Sun Nov 19 17:57:20 2017 -0500

Organizing tests

commit 1c632be1ebbf6a9a7e045e1a078b8a4aa2b0e5a9
Author: Jason Zhao <zhaos.jason@gmail.com>
Date: Sun Nov 19 17:34:46 2017 -0500

Updated gitignore again

commit e434a9ef8a307337a85c1876f0bb74c59728053d
Author: Jason Zhao <zhaos.jason@gmail.com>
Date: Sun Nov 19 17:30:03 2017 -0500

Updating gitignore

commit abdf6e7b34c05d0f7d248cf37aa6daf0239baaa8
Author: Sam Cohen <samsam489@gmail.com>
Date: Sun Nov 19 17:24:19 2017 -0500

added test out for float

commit 9584a0ee06c7c0dbd6d2b8bbef8d805db9b4fda9
Author: Sam Cohen <samsam489@gmail.com>
Date: Sun Nov 19 16:55:19 2017 -0500

Added float print format

commit 698f7d2c5df23ab0c85d35c30e117eedc2700e95
Author: Sam Cohen <samsam489@gmail.com>
Date: Sun Nov 19 16:27:39 2017 -0500

Added float printing

commit 109a908d90fb93831d0c866ed10da87c446f41fd
Author: Sam Cohen <samsam489@gmail.com>
Date: Sun Nov 19 15:31:44 2017 -0500

added init and assignment for floats

commit 7d81a79d1ed63d7a903dc8a6e4652f9ac50d9a1a
Author: Sam Cohen <samsam489@gmail.com>
Date: Sun Nov 19 15:25:39 2017 -0500

Updated floats type

commit 73cbf2720a5be74a3983645376725e866ab3108d
Author: samlouiscohen <samsam489@gmail.com>
Date: Sun Nov 19 15:21:35 2017 -0500

Update README.md

commit 9392b990aef28d2bc63f468748630ecd0034cbb8
Author: saahil9jain <saahil9jain@gmail.com>
Date: Sat Nov 18 15:59:49 2017 -0500

Fix semant.ml

commit ef977b6a19a6b04b3b068f8250081fb5b0a2527b
Author: Sam Cohen <samsam489@gmail.com>
Date: Sat Nov 18 15:54:43 2017 -0500

removed repeated float

commit a78b2bf8c95e42a19cfa60746301fe7e92a45c61
Merge: 96e3fb2 5ff24ba
Author: Sam Cohen <samsam489@gmail.com>
Date: Sat Nov 18 15:50:23 2017 -0500

Merge branch 'master' of github.com:samlouiscohen/Genesis_2

commit 96e3fb26141a6463e02fd1d13fda2a6fce0738c0
Author: Sam Cohen <samsam489@gmail.com>
Date: Sat Nov 18 15:50:03 2017 -0500

floats in codegen

commit 5ff24ba13a3ecc7f33a3dc754719a00954f983f5
Author: saahil9jain <saahil9jain@gmail.com>
Date: Fri Nov 17 20:21:39 2017 -0500

Update codegen for float

commit 0d383573045945c238896ea7ec1e9bafaa2b0346
Author: Sam Cohen <samsam489@gmail.com>
Date: Fri Nov 17 20:00:10 2017 -0500

Added Float

commit 91a67621a7cacbe7f0822c5e73261a87519f6d35
Author: Sam Cohen <samsam489@gmail.com>
Date: Fri Nov 17 16:57:16 2017 -0500

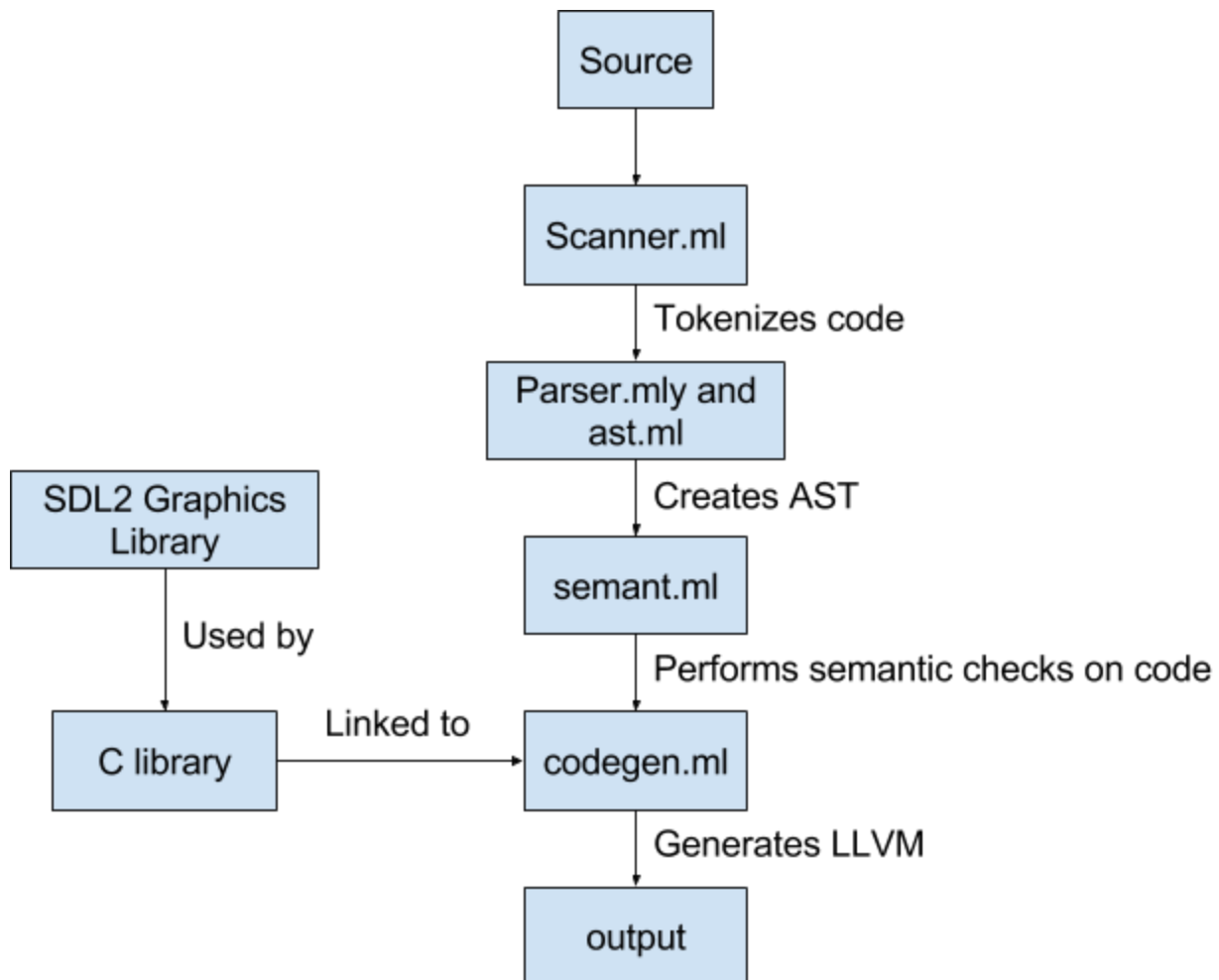
microc

commit 51e2b775e02907105db64bf2ba03b6d88385dfa3
Author: Sam Cohen <samsam489@gmail.com>
Date: Fri Nov 17 16:56:50 2017 -0500

first commit

Architectural Design

The processing of the program file occurs like so:



Credits:

Jason:

- Array declaration, initialization, and deletion
- Declaration / assignment intermixing
- Float / integer implicit casting
- Git repo manager, build files, test suite
- Tron and Game of Life Demos

Leo:

- Backend implementation of clusters
- SDL integration
- Snake demo

- Test Suite: visual tests

Michael:

- All files for implementing color.
- Scanner, Parser, Ast, Semant, and Codegen for everything involving clusters
- System loop and all architecture for game functions like init, update, startgame, and quit

Saahil:

- All files for implementing data types (string, etc.)
- Semantic checking, advanced data types / normal ones (cluster, color)
- Added "god" functionality :)
- Test suite: regressions, minorly on visual tests

Sam:

- Code Generation organization
- Robust Semantic Checking for array declaration, initialization, and assignment
- Scanner single-line comments and regex for data type implementation
- Print functionality

Test Plan

Test Suites

We used 3 test suites: one for regression testing (testing functionality from microc), one for testing game-specific features, like rendering and movement, and one for testing miscellaneous non-game related language features that we implemented. In our files, these are organized into folders named “rtests”, “vtests”, and “tests”, respectively.

Test Cases

We decided to test each feature by first setting our expectations for how a feature should and shouldn't be used. We then wrote tests for failure cases like faulty value passing as well as success cases, including integration tests with the other features that we implemented. For many of the visual tests, we were unable to test the functionality purely through output, so we made quick tests that we verified visually to confirm that features were working.

Testing Process

We automated our testing by creating files with our expected output from each program, and then we used a makefile with targets that represented each test suite to quickly run through all of the tests in each suite.

Credits:

Jason:

Leo: Snake demo, Visual Test suites

Michael: Jumpman Demo, various tests for clusters and colors

Saahil: Regression test suite (strings, etc.), Additions to visual test suite

Sam: Snake Demo, broad additions to Test, Regression, and Visual test suites

Example Sources and Targets:

Source 1:

```
cluster cl;  
cluster cl2;  
  
void init(){  
  
int main(){  
    color c;  
    color c2;  
    color c3;  
    c = #0,0,255#;  
    c3 = #255, 0, 0#;
```

```

    cl = $ 50, 50, 100, 100, 1, 0, c $;
    cl2 = $ 100, 100, 400, 100, 0, 0, c3 $;
    c2 = #0, 0, 0#;
    startGame(640, 480, c2);
    print(1);
    return 0;
}

void update(int f){
    if(cl @ cl2){
        cl.draw = false;
        cl2.draw = false;
        quit();
    }
    cl.x = cl.x + 10;
}

```

Target 1:

```

; ModuleID = 'Genesis'
target datalayout = "e-m:o-i64:64-f80:128-n8:16:32:64-S128"

%color = type { i32, i32, i32 }

@cl2 = global i32 0
@cl = global i32 0
@fmt = private unnamed_addr constant [4 x i8] c"%d\oA\oo"
@fmt.1 = private unnamed_addr constant [5 x i8] c"%lf\oA\oo"
@fmt.2 = private unnamed_addr constant [4 x i8] c"%s\oA\oo"
@fmt.3 = private unnamed_addr constant [4 x i8] c"%d\oA\oo"
@fmt.4 = private unnamed_addr constant [5 x i8] c"%lf\oA\oo"
@fmt.5 = private unnamed_addr constant [4 x i8] c"%s\oA\oo"
@fmt.6 = private unnamed_addr constant [4 x i8] c"%d\oA\oo"
@fmt.7 = private unnamed_addr constant [5 x i8] c"%lf\oA\oo"
@fmt.8 = private unnamed_addr constant [4 x i8] c"%s\oA\oo"

declare i32 @printf(i8*, ...)

declare i32 @printbig(i32)

declare i32 @initScreen(%color*, i32, i32)

declare void @startGame(%color*, i32, i32)

declare i1 @isKeyDown(i8*)

declare i1 @isKeyUp(i8*)

declare i1 @isKeyHeld(i8*)

declare i32 @newCluster(i32, i32, i32, i32, i32, i32, %color*)

declare void @deleteCluster(i32)

declare i32 @randomInt(i32)

```

```

declare i1 @detectCollision(i32, i32)

declare void @quitGame()

declare void @setFPS(i32)

declare i32 @getX(i32)

declare i32 @getY(i32)

declare i32 @getDX(i32)

declare i32 @getDY(i32)

declare i32 @getHeight(i32)

declare i32 @getWidth(i32)

declare %color* @getColor(i32)

declare i1 @getDraw(i32)

declare i32 @setX(i32, i32)

declare void @setY(i32, i32)

declare void @setDX(i32, i32)

declare void @setDY(i32, i32)

declare void @setHeight(i32, i32)

declare void @setWidth(i32, i32)

declare void @setColor(i32, %color*)

declare void @setDraw(i32, i1)

define void @update(i32 %f) {
entry:
    %f1 = alloca i32
    store i32 %f, i32* %f1
    %cl = load i32, i32* @cl
    %cl2 = load i32, i32* @cl2
    %col = call i1 @detectCollision(i32 %cl, i32 %cl2)
    br i1 %col, label %then, label %else

merge:
    ; preds = %else, %then
    %cl5 = load i32, i32* @cl
    %cl6 = load i32, i32* @cl
    %xVal = call i32 @getX(i32 %cl6)
    %tmp = add i32 %xVal, 10
    %o = call i32 @setX(i32 %cl5, i32 %tmp)
    ret void

```



```

then:                                ; preds = %entry
  %cl3 = load i32, i32* @cl
  call void @setDraw(i32 %cl3, i1 false)
  %cl24 = load i32, i32* @cl2
  call void @setDraw(i32 %cl24, i1 false)
  call void @quitGame()
  br label %merge

else:                                  ; preds = %entry
  br label %merge
}

define i32 @main() {
entry:
  %c = alloca %color*
  %c2 = alloca %color*
  %c3 = alloca %color*
  %color_tmp = alloca %color
  %clr_ptr = alloca %color*
  %r = getelementptr inbounds %color, %color* %color_tmp, i32 0, i32 0
  store i32 0, i32* %r
  %g = getelementptr inbounds %color, %color* %color_tmp, i32 0, i32 1
  store i32 0, i32* %g
  %b = getelementptr inbounds %color, %color* %color_tmp, i32 0, i32 2
  store i32 255, i32* %b
  store %color* %color_tmp, %color** %clr_ptr
  %o = load %color*, %color** %clr_ptr
  store %color* %o, %color** %c
  %color_tmp1 = alloca %color
  %clr_ptr2 = alloca %color*
  %r3 = getelementptr inbounds %color, %color* %color_tmp1, i32 0, i32 0
  store i32 255, i32* %r3
  %g4 = getelementptr inbounds %color, %color* %color_tmp1, i32 0, i32 1
  store i32 0, i32* %g4
  %b5 = getelementptr inbounds %color, %color* %color_tmp1, i32 0, i32 2
  store i32 0, i32* %b5
  store %color* %color_tmp1, %color** %clr_ptr2
  %i = load %color*, %color** %clr_ptr2
  store %color* %i, %color** %c3
  %c6 = load %color*, %color** %c
  %newClust = call i32 @newCluster(i32 50, i32 50, i32 100, i32 100, i32 1, i32 0, %color* %c6)
  store i32 %newClust, i32* @cl
  %c37 = load %color*, %color** %c3
  %newClust8 = call i32 @newCluster(i32 100, i32 100, i32 400, i32 100, i32 0, i32 0, %color* %c37)
  store i32 %newClust8, i32* @cl2
  %color_tmp9 = alloca %color
  %clr_ptr10 = alloca %color*
  %r11 = getelementptr inbounds %color, %color* %color_tmp9, i32 0, i32 0
  store i32 0, i32* %r11
  %g12 = getelementptr inbounds %color, %color* %color_tmp9, i32 0, i32 1
  store i32 0, i32* %g12
  %b13 = getelementptr inbounds %color, %color* %color_tmp9, i32 0, i32 2
  store i32 0, i32* %b13
  store %color* %color_tmp9, %color** %clr_ptr10
  %2 = load %color*, %color** %clr_ptr10
  store %color* %2, %color** %c2

```

```

%c214 = load %color*, %color** %c2
call void @startGame(%color* %c214, i32 640, i32 480)
%printf = call i32 @printf(i8* getelementptr inbounds ([4 x i8], [4 x i8]* @fmt.3, i32 0, i32 0),
i32 1)
ret i32 0
}

define void @init() {
entry:
ret void
}

```

Source 2:

```

void init(){}
void update(int f){

int main(){
int size;
int[] x;
int i;
size = 10;
x = new int[size];

for (i = 0; i < size; i = i + 1) {
x[i] = i + 1;
}
for (; i > 0; i = i - 1) {
print(x[i - 1]);
}
return 0;
}

```

Target 2:

```

; ModuleID = 'Genesis'
target datalayout = "e-m:0-i64:64-f80:128-n8:16:32:64-S128"

%color = type { i32, i32, i32 }

@fmt = private unnamed_addr constant [4 x i8] c"%d\oA\oo"
@fmt.1 = private unnamed_addr constant [5 x i8] c"%lf\oA\oo"
@fmt.2 = private unnamed_addr constant [4 x i8] c"%s\oA\oo"
@fmt.3 = private unnamed_addr constant [4 x i8] c"%d\oA\oo"
@fmt.4 = private unnamed_addr constant [5 x i8] c"%lf\oA\oo"
@fmt.5 = private unnamed_addr constant [4 x i8] c"%s\oA\oo"
@fmt.6 = private unnamed_addr constant [4 x i8] c"%d\oA\oo"
@fmt.7 = private unnamed_addr constant [5 x i8] c"%lf\oA\oo"
@fmt.8 = private unnamed_addr constant [4 x i8] c"%s\oA\oo"

```

```
declare i32 @printf(i8*, ...)
declare i32 @printbig(i32)
declare i32 @initScreen(%color*, i32, i32)
declare void @startGame(%color*, i32, i32)
declare i1 @isKeyDown(i8*)
declare i1 @isKeyUp(i8*)
declare i1 @isKeyHeld(i8*)
declare i32 @newCluster(i32, i32, i32, i32, i32, i32, %color*)
declare void @deleteCluster(i32)
declare i32 @randomInt(i32)
declare i1 @detectCollision(i32, i32)
declare void @quitGame()
declare void @setFPS(i32)
declare i32 @getX(i32)
declare i32 @getY(i32)
declare i32 @getDX(i32)
declare i32 @getDY(i32)
declare i32 @getHeight(i32)
declare i32 @getWidth(i32)
declare %color* @getColor(i32)
declare i1 @getDraw(i32)
declare i32 @setX(i32, i32)
declare void @setY(i32, i32)
declare void @setDX(i32, i32)
declare void @setDY(i32, i32)
```

```

declare void @setHeight(i32, i32)

declare void @setWidth(i32, i32)

declare void @setColor(i32, %color*)

declare void @setDraw(i32, i1)

define i32 @main() {
entry:
  %size = alloca i32
  %x = alloca i32*
  %i = alloca i32
  store i32 10, i32* %size
  %size1 = load i32, i32* %size
  %mallocsize = mul i32 %size1, ptrtoint (i32* @getelementptr (i32, i32* null, i32 1) to i32)
  %malloccall = tail call i8* @malloc(i32 %mallocsize)
  %o = bitcast i8* %malloccall to i32*
  store i32* %o, i32** %x
  store i32 0, i32* %i
  br label %while

while:
  ; preds = %while_body, %entry
  %i6 = load i32, i32* %i
  %size7 = load i32, i32* %size
  %tmp8 = icmp slt i32 %i6, %size7
  br i1 %tmp8, label %while_body, label %merge

while_body:
  ; preds = %while
  %i2 = load i32, i32* %i
  %tmp = add i32 %i2, 1
  %i3 = load i32, i32* %i
  %1 = load i32*, i32** %x
  %2 = getelementptr i32, i32* %1, i32 %i3
  store i32 %tmp, i32* %2
  %i4 = load i32, i32* %i
  %tmp5 = add i32 %i4, 1
  store i32 %tmp5, i32* %i
  br label %while

merge:
  ; preds = %while
  br label %while9

while9:
  ; preds = %while_body10, %merge
  %i15 = load i32, i32* %i
  %tmp16 = icmp sgt i32 %i15, 0
  br i1 %tmp16, label %while_body10, label %merge17

```

```

while_body10:                                ; preds = %while9
    %i11 = load i32, i32* %i
    %tmp12 = sub i32 %i11, 1
    %3 = load i32*, i32** %X
    %4 = getelementptr i32, i32* %3, i32 %tmp12
    %5 = load i32, i32* %4
    %printf = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4 x i8]* @fmt, i32 0,
i32 0), i32 %5)
    %i13 = load i32, i32* %i
    %tmp14 = sub i32 %i13, 1
    store i32 %tmp14, i32* %i
    br label %while9

merge17:                                     ; preds = %while9
    ret i32 0
}

define void @update(i32 %f) {
entry:
    %f1 = alloca i32
    store i32 %f, i32* %f1
    ret void
}

define void @init() {
entry:
    ret void
}

declare noalias i8* @malloc(i32)

```

Lessons Learned

Michael:

- Set hard deadlines for your team, and have everyone commit themselves to the same block of time each week to work on the project.
- First make a test for the goal you want to achieve, and then work on your program until the test works.
- It is not necessary to completely understand MicroC before you begin working on the project. As you try to implement basic functionalities, you will learn in the process.

Jason:

In addition to the prudent advice about starting early that everyone else has probably already said, I offer these words of wisdom:

- Working for a continuous 3 hours is better than three 1 hour blocks. A lot of the work you'll be doing will require lots of persistence and dedication. It's tempting to want to take a lot of frequent small breaks, but try to limit them to inbetween problems rather than when you get frustrated or tired. There is a lot of overhead in remembering what you did and who did what and where you were and what you've tried, so you can gain a lot of efficiency by cutting it out. Of course, sometimes it is good to take a break from a problem and there are diminishing returns so ultimately it's up to you to choose the right balance.
- Use other past projects as oracles, rather than infallible truths. These are an amazing resource for learning, but remember that even if they have a star, their implementation of a feature or design of their language is not the only way to do it! Many times there are better ways to do what you want!
- Don't give up! Learning a new language and (for many) a new programming paradigm in a semester is already tough enough, let alone implementing a compiler! The MicroC starter code looks intimidating when you first start, but after a few (hopefully continuous) hours of staring at it, it'll start making sense. Make sure you understand its quirks and limitations and how you can build off of them / get around them.
- Utilize the TAs. Don't be afraid to meet more than once a week and with TAs other than your own. Try to ask super specific questions and dive deep from the start. Talking in generalizations will only give you an inaccurate picture of your project. You have to actually start implementing it to get a real feel for how the process works.
- Keep a unified, central log of who's doing what, what you want to do, and what's been done. It's annoying at first, but if you're consistent it'll make the write-up later so much easier.

Sam:

- Don't get too caught up in following the exact specifications of your LRM. As the semester progresses, odds are you'll realize that initial language goals and formulations were naive or could be achieved in a more elegant way.
- At the beginning of the semester, each team member should play around with every aspect of the language design process. This will give each member some further insight into what job their skills best align with.
- Make sure to look through the projects that the current TAs worked on in prior semesters. Language design is a vast domain, and it's invaluable to know which TA is best equipped to answer your language-specific questions.
- Form a deliverable system to keep everyone accountable and up to date. This is also applicable to watching the recorded lectures immediately following an absence.

Leo

- KNOW/LEARN GIT!!!! or any version control system really. Some members of our group weren't too comfortable with git and it made everything more difficult than it had to be when it came to implementing features in an efficient modular fashion.
- While it is important to set internal deadlines, it is not a big deal if you need more time. This project is a huge undertaking and there are bound to be things that trip you up. Nonetheless, do make sure to start as early as possible.
- Make sure to be up to date with the lectures. Once you fall behind it is easy for the material to pile up.

Saahil

- Definitely invest as much time as possible into understanding the system / initial framework. It's easy to dive straight into the coding, working by analogy to other parts of the language. But, at some point, getting an understanding of how each of the layers interact and each line of code works pays dividends in the long run. In fact, with OCaml, this idea of "understanding" leads to elegant code. That being said, don't procrastinate too much and start early :).
- Also, in working with a team, make sure that you all can find to work in person! Our development flow involved working together as much as possible, and we found that the bouncing of ideas in proximity helped more than working in isolation.

Appendix

1) Genesis compiler

```
(* Top-level of the Genesis compiler: scan & parse the input,
    check the resulting AST, generate LLVM IR, and dump the module *)

module StringMap = Map.Make(String)

type action = Ast | LLVM_IR | Compile

let _ =
  let action = ref Compile in
  let set_action a () = action := a in
  let speclist = [
    ("-a", Arg.Unit (set_action Ast), "Print the SAST");
    ("-l", Arg.Unit (set_action LLVM_IR), "Print the generated LLVM IR");
    ("-c", Arg.Unit (set_action Compile),
     "Check and print the generated LLVM IR (default)");
  ] in
  let usage_msg = "usage: ./genesis.native [-a|-l|-c] [file.god]" in
  let channel = ref stdin in
  Arg.parse speclist (fun filename -> channel := open_in filename) usage_msg;
  let lexbuf = Lexing.from_channel !channel in
  let ast = Parser.program Scanner.token lexbuf in
  Semant.check ast;
  match !action with
  | Ast -> print_string (Ast.string_of_program ast)
  | LLVM_IR -> print_string (Llvm.string_of_llmodule (Codegen.translate ast))
  | Compile -> let m = Codegen.translate ast in
    Llvm_analysis.assert_valid_module m;
    print_string (Llvm.string_of_llmodule m)
```

2.) scanner.mll

```
(* Ocamllex scanner for Genesis *)

(*
  Authors:
  - Leon Stilwell
  - Michael Wang
  - Jason Zhao
  - Sam Cohen
*)

{ open Parser }

rule token = parse
```



```

| ' ' '\t' '\r' '\n' { token lexbuf } (* Whitespace *)
| "/"      { scomment lexbuf }      (* Comments *)
| "/*"     { comment lexbuf }      (* Comments *)
| '@'      { AT }
| '#'      { POUND }
| '$'      { DOLLAR }
| '('      { LPAREN }
| ')'      { RPAREN }
| '{'      { LBRACE }
| '}'      { RBRACE }
| '['      { LBRACKET }
| ']'      { RBRACKET }
| ';'      { SEMI }
| '%'      { MOD }
| '+'      { PLUS }
| '-'      { MINUS }
| '*'      { TIMES }
| '/'      { DIVIDE }
| '='      { ASSIGN }
| ','      { COMMA }
| '.'      { DOT }
| "=="     { EQ }
| "!="     { NEQ }
| '<'      { LT }
| "<="     { LEQ }
| ">"      { GT }
| ">="     { GEQ }
| "&&"     { AND }
| "||"     { OR }
| "!"      { NOT }
| "if"     { IF }
| "else"   { ELSE }
| "for"    { FOR }
| "while"  { WHILE }
| "return" { RETURN }
| "int"    { INT }
| "float"  { FLOAT }
| "string" { STRING }
| "bool"   { BOOL }
| "void"   { VOID }
| "color"  { COLOR }
| "cluster" { CLUSTER }
| "true"   { TRUE }
| "false"  { FALSE }
| "new"    { NEW }
| "delete" { DELETE }
| ['0'-'9']+ as lxm { LITERAL(int_of_string lxm) }
| ['0'-'9']+.'['0'-'9']+ as lxm { FLOATLIT(float_of_string lxm) }
| ['a'-'z' 'A'-'Z']['a'-'z' 'A'-'Z' '0'-'9' '_']* as lxm { ID(lxm) }
| ''' ([[ ^ ''' ] | "\\\""])* as strlit) ''' { STRINGLIT(strlit) }
| eof { EOF }
| _ as char { raise (Failure("illegal character " ^ Char.escaped char)) }

```

```

and comment = parse
  "*/" { token lexbuf }
| _    { comment lexbuf }

and scomment = parse
  "\n" { token lexbuf }
| _    { scomment lexbuf }

```

3) parser.mly

```

/* Ocaml yacc parser for Genesis */
/*
  Authors:
  - Michael Wang
  - Jason Zhao
  - Sam Cohen
*/

%{ open Ast %}

%token SEMI LPAREN RPAREN LBRACE RBRACE COMMA
%token PLUS MINUS TIMES DIVIDE MOD ASSIGN NOT
%token EQ NEQ LT LEQ GT GEQ TRUE FALSE AND OR
%token RETURN IF ELSE FOR WHILE INT FLOAT BOOL VOID STRING
%token LBRACKET RBRACKET COLOR CLUSTER NEW DOLLAR DOT POUND
%token AT DELETE
%token <string> PROPERTY
%token <int> LITERAL
%token <string> ID
%token <float> FLOATLIT
%token <string> STRINGLIT
%token EOF

%nonassoc NOELSE
%nonassoc ELSE
%right ASSIGN
%left OR
%left AND
%left EQ NEQ
%left LT GT LEQ GEQ
%left AT
%left PLUS MINUS
%left TIMES DIVIDE MOD
%right NOT NEG
%left DOT

%start program
%type <Ast.program> program

%%

```

```

program:
  decls EOF { $1 }

decls:
  /* nothing */ { [], [] }
  | decls vdecl { ($2 :: fst $1), snd $1 }
  | decls fdecl { fst $1, ($2 :: snd $1) }

fdecl:
  typ ID LPAREN formals_opt RPAREN LBRACE decl_list RBRACE
  { { typ = $1;
    fname = $2;
    formals = $4;
    locals = List.rev (fst $7);
    body = List.rev (snd $7) } }

formals_opt:
  /* nothing */ { [] }
  | formal_list { List.rev $1 }

formal_list:
  typ ID { [($1,$2)] }
  | formal_list COMMA typ ID { ($3,$4) :: $1 }

primitive:
  INT { Int }
  | FLOAT { Float }
  | BOOL { Bool }
  | VOID { Void }
  | STRING { String }
  | CLUSTER { Cluster }
  | COLOR { Color }

typ:
  primitive { $1 }
  | primitive LBRACKET RBRACKET { ArrayType($1) }

decl_list:
  /* nothing */ { [], [] }
  | decl_list vdecl { ($2 :: fst $1), snd $1 }
  | decl_list stmt { fst $1, ($2 :: snd $1) }

vdecl:
  typ ID SEMI { ($1, $2) }

stmt_list:
  /* nothing */ { [] }
  | stmt_list stmt { $2 :: $1 }

stmt:
  expr SEMI { Expr $1 }
  | RETURN SEMI { Return Noexpr }
  | RETURN expr SEMI { Return $2 }

```

```

| LBRACE stmt_list RBRACE { Block(List.rev $2) }
| IF LPAREN expr RPAREN stmt %prec NOELSE { If($3, $5, Block([])) }
| IF LPAREN expr RPAREN stmt ELSE stmt { If($3, $5, $7) }
| FOR LPAREN expr_opt SEMI expr SEMI expr_opt RPAREN stmt
  { For($3, $5, $7, $9) }
| WHILE LPAREN expr RPAREN stmt { While($3, $5) }

```

expr_opt:

```

/* nothing */ { Noexpr }
| expr { $1 }

```

expr:

```

LITERAL { Literal($1) }
| FLOATLIT { FloatLit($1) }
| STRINGLIT { StringLit($1) }
| TRUE { BoolLit(true) }
| FALSE { BoolLit(false) }
| POUND expr COMMA expr COMMA expr POUND { ColorLit($2, $4, $6) }
| DOLLAR
  expr COMMA expr COMMA expr COMMA expr COMMA
  expr COMMA expr COMMA expr
DOLLAR { ClusterLit($2, $4, $6, $8, $10, $12, $14)}
| ID { Id($1) }
| expr AT expr { Collision($1, $3) }
| expr PLUS expr { Binop($1, Add, $3) }
| expr MINUS expr { Binop($1, Sub, $3) }
| expr TIMES expr { Binop($1, Mult, $3) }
| expr DIVIDE expr { Binop($1, Div, $3) }
| expr MOD expr { Binop($1, Mod, $3) }
| expr EQ expr { Binop($1, Equal, $3) }
| expr NEQ expr { Binop($1, Neq, $3) }
| expr LT expr { Binop($1, Less, $3) }
| expr LEQ expr { Binop($1, Leq, $3) }
| expr GT expr { Binop($1, Greater, $3) }
| expr GEQ expr { Binop($1, Geq, $3) }
| expr AND expr { Binop($1, And, $3) }
| expr OR expr { Binop($1, Or, $3) }
| expr DOT ID { PropertyAccess($1, $3) }
| expr DOT ID ASSIGN expr { PropertyAssign($1, $3, $5) }
| MINUS expr %prec NEG { Unop(Neg, $2) }
| NOT expr { Unop(Not, $2) }
| ID ASSIGN expr { Assign($1, $3) }
| ID LPAREN actuals_opt RPAREN { Call($1, $3) }
| LPAREN expr RPAREN { $2 }
| NEW primitive LBRACKET expr RBRACKET { ArrayInit($2, $4) }
| DELETE ID { ArrayDelete($2) }
| ID LBRACKET expr RBRACKET ASSIGN expr { ArrayAssign($1, $3, $6) }
| ID LBRACKET expr RBRACKET { ArrayAccess($1, $3) }

```

actuals_opt:

```

/* nothing */ { [] }
| actuals_list { List.rev $1 }

```

```

actuals_list:
  expr          { [$1] }
| actuals_list COMMA expr { $3 :: $1 }

```

4.) codegen.ml

```

(* Code generation: translate takes a semantically checked AST and
   produces LLVM IR

```

```

    Authors: Michael Wang

```

```

*)

```

```

module L = LlvM

```

```

module A = Ast

```

```

module StringMap = Map.Make(String)

```

```

module String = String

```

```

let translate (globals, functions) =
  let context = L.global_context () in
  let the_module = L.create_module context "Genesis" in
  ignore(L.set_data_layout "e-m:o-i64:64-f80:128-n8:16:32:64-S128" the_module); (* sets data
layout to match machine *)

```

```

  let i64_t = L.i64_type context in
  let i32_t = L.i32_type context in
  let i8_t = L.i8_type context in
  let i1_t = L.i1_type context in
  let flt_t = L.double_type context in
  let pointer_t = L.pointer_type in
  let void_t = L.void_type context in

```

```

  let color_t = L.named_struct_type context "color" in
  L.struct_set_body color_t [| i32_t ; i32_t ; i32_t |] false; (* need to change here if
source file changes *)

```

```

  let col_ptr_t = pointer_t color_t in
  let cluster_t = i32_t in

```

```

let rec ltype_of_typ = function
  A.Int -> i32_t
| A.Float -> flt_t
| A.Bool -> i1_t
| A.Void -> void_t
| A.Cluster -> cluster_t
| A.String -> pointer_t i8_t
| A.ArrayType(t) -> pointer_t (ltype_of_typ t)
| A.Color -> col_ptr_t

```

```

in

```

```

(* Declare printf(), which the print built-in function will call *)

```

```

let printf_t = L.var_arg_function_type i32_t [| L.pointer_type i8_t |] in
let printf_func = L.declare_function "printf" printf_t the_module in

```

```

(* Declare the built-in printbig() function *)
let printbig_t = L.function_type i32_t [| i32_t |] in
let printbig_func = L.declare_function "printbig" printbig_t the_module in

let initScreen_t = L.function_type i32_t [| L.pointer_type color_t; i32_t; i32_t |] in
let initScreen_func = L.declare_function "initScreen" initScreen_t the_module in

let startGame_t = L.function_type void_t [| L.pointer_type color_t; i32_t; i32_t |] in
let startGame_func = L.declare_function "startGame" startGame_t the_module in

let isKeyDown_t = L.function_type i1_t [| pointer_t i8_t |] in
let isKeyDown_func = L.declare_function "isKeyDown" isKeyDown_t the_module in

let isKeyUp_t = L.function_type i1_t [| pointer_t i8_t |] in
let isKeyUp_func = L.declare_function "isKeyUp" isKeyUp_t the_module in

let isKeyHeld_t = L.function_type i1_t [| pointer_t i8_t |] in
let isKeyHeld_func = L.declare_function "isKeyHeld" isKeyHeld_t the_module in

let newCluster_t = L.function_type i32_t [|i32_t; i32_t; i32_t; i32_t; i32_t; i32_t;
col_ptr_t |] in
let newCluster_func = L.declare_function "newCluster" newCluster_t the_module in

let deleteCluster_t = L.function_type void_t [|i32_t|] in
let deleteCluster_func = L.declare_function "deleteCluster" deleteCluster_t the_module in

let randomInt_t = L.function_type i32_t [|i32_t|] in
let randomInt_func = L.declare_function "randomInt" randomInt_t the_module in

let detectCollision_t = L.function_type i1_t [|i32_t; i32_t|] in
let detectCollision_func = L.declare_function "detectCollision" detectCollision_t the_module
in

let quitGame_t = L.function_type void_t [| |] in
let quitGame_func = L.declare_function "quitGame" quitGame_t the_module in

let setFPS_t = L.function_type void_t [|i32_t|] in
let setFPS_func = L.declare_function "setFPS" setFPS_t the_module in

(* Getters *)
let getX_t = L.function_type i32_t [|i32_t|] in
let getX_func = L.declare_function "getX" getX_t the_module in

let getY_t = L.function_type i32_t [|i32_t|] in
let getY_func = L.declare_function "getY" getY_t the_module in

let getDX_t = L.function_type i32_t [|i32_t|] in
let getDX_func = L.declare_function "getDX" getDX_t the_module in

let getDY_t = L.function_type i32_t [|i32_t|] in
let getDY_func = L.declare_function "getDY" getDY_t the_module in

```

```

let getHeight_t = L.function_type i32_t [|i32_t|] in
let getHeight_func = L.declare_function "getHeight" getHeight_t the_module in

let getWidth_t = L.function_type i32_t [|i32_t|] in
let getWidth_func = L.declare_function "getWidth" getWidth_t the_module in

let getColor_t = L.function_type col_ptr_t [|i32_t|] in
let getColor_func = L.declare_function "getColor" getColor_t the_module in

let getDraw_t = L.function_type i1_t [|i32_t|] in
let getDraw_func = L.declare_function "getDraw" getDraw_t the_module in

(* Setters *)
let setX_t = L.function_type i32_t [|i32_t; i32_t|] in
let setX_func = L.declare_function "setX" setX_t the_module in

let setY_t = L.function_type void_t [|i32_t; i32_t|] in
let setY_func = L.declare_function "setY" setY_t the_module in

let setDX_t = L.function_type void_t [|i32_t; i32_t|] in
let setDX_func = L.declare_function "setDX" setDX_t the_module in

let setDY_t = L.function_type void_t [|i32_t; i32_t|] in
let setDY_func = L.declare_function "setDY" setDY_t the_module in

let setHeight_t = L.function_type void_t [|i32_t; i32_t|] in
let setHeight_func = L.declare_function "setHeight" setHeight_t the_module in

let setWidth_t = L.function_type void_t [|i32_t; i32_t|] in
let setWidth_func = L.declare_function "setWidth" setWidth_t the_module in

let setColor_t = L.function_type void_t [|i32_t; col_ptr_t|] in
let setColor_func = L.declare_function "setColor" setColor_t the_module in

let setDraw_t = L.function_type void_t [|i32_t; i1_t|] in
let setDraw_func = L.declare_function "setDraw" setDraw_t the_module in

(* Define each function (arguments and return type) so we can call it *)
let function_decls =
  let function_decl m fdecl =
    let name = fdecl.A.fname
    and formal_types =
      Array.of_list (List.map (fun (t,_) -> ltype_of_typ t) fdecl.A.formals)
    in let ftype = L.function_type (ltype_of_typ fdecl.A.typ) formal_types in
    StringMap.add name (L.define_function name ftype the_module, fdecl) m in
  List.fold_left function_decl StringMap.empty functions in

(* Cast int to float *)
let make_float var builder =
  if L.type_of var = flt_t then
    var
  else if L.type_of var = i32_t || L.type_of var = i64_t then
    L.build_sitofp var flt_t "" builder

```

```

else if L.type_of var = (pointer_t flt_t) then
  L.build_load var "" builder
else
  raise (Failure ("Unknown cast to float"))
in

(* Cast float to int, don't modify bools *)
let make_int var builder =
  if L.type_of var = i32_t || L.type_of var = i64_t ||
     L.type_of var = i8_t || L.type_of var = i1_t then
    var
  else if L.type_of var = flt_t then
    L.build_fptosi var i32_t "" builder
  else if L.type_of var = (pointer_t i32_t) || L.type_of var = (pointer_t i64_t) then
    L.build_load var "" builder
  else
    raise (Failure ("Unknown cast to int"))
in

(* Declare each global variable; remember its value in a map *)
let global_vars =
  let global_var m (t, n) =
    let init = match t with
      | A.ArrayType(_) -> L.const_pointer_null (ltype_of_typ t)
      | A.Color -> L.const_pointer_null (ltype_of_typ t)
      | A.String -> L.const_pointer_null (ltype_of_typ t)
      | _ -> L.const_int (ltype_of_typ t) 0
    in StringMap.add n (L.define_global n init the_module) m in
  List.fold_left global_var StringMap.empty globals in

(* Fill in the body of the given function *)
let build_function_body fdecl =
  let (the_function, _) = StringMap.find fdecl.A.fname function_decls in
  let builder = L.builder_at_end context (L.entry_block the_function) in

  let int_format_str = L.build_global_stringptr "%d\n" "fmt" builder in
  let float_format_str = L.build_global_stringptr "%lf\n" "fmt" builder in
  let string_format_str = L.build_global_stringptr "%s\n" "fmt" builder in

  (* Construct the function's "locals": formal arguments and locally
     declared variables. Allocate each on the stack, initialize their
     value, if appropriate, and remember their values in the "locals" map *)

  let local_vars =
    let add_formal m (t, n) p = L.set_value_name n p;
    let local = L.build_alloca (ltype_of_typ t) n builder in
    ignore (L.build_store p local builder);
    StringMap.add n local m in

  let add_local m (t, n) =
    let local_var = L.build_alloca (ltype_of_typ t) n builder
    in StringMap.add n local_var m in

```



```

let formals = List.fold_left2 add_formal StringMap.empty fdecl.A.formals
  (Array.to_list (L.params the_function)) in
List.fold_left add_local formals fdecl.A.locals in

(* Return the value (usually a memory address) for a variable or formal argument *)
let lookup n = try StringMap.find n local_vars
  with Not_found -> StringMap.find n global_vars
in

(* Get array value of name at index i *)
let get_array_element name i builder =
  let arr = L.build_load (lookup name) "" builder in
  let ptr = L.build_gep arr [| i |] "" builder in
  L.build_load ptr "" builder
in

(* Set array element of name at index i to val *)
let set_array_element name i v builder =
  let arr = L.build_load (lookup name) "" builder in
  let ptr = L.build_gep arr [| i |] "" builder in
  L.build_store v ptr builder
in

(* Initializes array of typ of size len *)
let init_array typ len builder =
  L.build_array_malloc (ltype_of_typ typ) len "" builder
in

(* Construct code for an expression; return its value *)
let rec expr builder = function
  A.Literal i -> L.const_int i32_t i
| A.StringLit s -> L.build_global_stringptr s "tmp" builder
| A.FloatLit fl -> L.const_float flt_t fl
| A.BoolLit b -> L.const_int i1_t (if b then 1 else 0)
| A.ColorLit (r, g, b) ->
  let ctmp = L.build_alloca color_t "color_tmp" builder in
  let cptr = L.build_alloca (L.pointer_type color_t) "clr_ptr" builder in
  let e1 = expr builder r
  and e2 = expr builder g
  and e3 = expr builder b in

  let rtmp = L.build_struct_gep ctmp 0 "r" builder in
  ignore (L.build_store e1 rtmp builder);
  let gtmp = L.build_struct_gep ctmp 1 "g" builder in
  ignore (L.build_store e2 gtmp builder);
  let btmp = L.build_struct_gep ctmp 2 "b" builder in
  ignore (L.build_store e3 btmp builder);
  ignore (L.build_store ctmp cptr builder);
  let colld = L.build_load cptr "" builder in
  colld

```

```

| A.ClusterLit (l, w, x, y, dx, dy, c)->
  let xPos = expr builder x in
  let yPos = expr builder y in
  let xVel = expr builder dx in
  let yVel = expr builder dy in
  let len = expr builder l in
  let wid = expr builder w in
  let color = expr builder c in

  L.build_call newCluster_func [| len; wid; xPos; yPos; xVel; yVel; color|] "newClust"
builder

| A.Collision (c1, c2) ->
  let c1' = expr builder c1 in
  let c2' = expr builder c2 in
  L.build_call detectCollision_func [|c1'; c2'|] "col" builder
| A.Noexpr -> L.const_int i32_t 0
| A.Id s -> L.build_load (lookup s) s builder
| A.Property _ -> L.const_int i32_t 0
| A.PropertyAccess(c, p) ->
  let cluster = expr builder c in
  (match p with
  | "x" -> L.build_call getX_func [|cluster|] "xVal" builder
  | "y" -> L.build_call getY_func [|cluster|] "yVal" builder
  | "dx" -> L.build_call getDX_func [|cluster|] "dxVal" builder
  | "dy" -> L.build_call getDY_func [|cluster|] "dyVal" builder
  | "height" -> L.build_call getHeight_func [|cluster|] "hVal" builder
  | "width" -> L.build_call getWidth_func [|cluster|] "wVal" builder
  | "clr" -> L.build_call getColor_func [|cluster|] "colVal" builder
  | "draw" -> L.build_call getDraw_func [|cluster|] "drawVal" builder
  | _ -> raise (Failure ("Property does not exist"))
  )
| A.PropertyAssign(c, p, e) ->
  let cluster = expr builder c in
  let e' = expr builder e in
  (match p with
  | "x" -> L.build_call setX_func [|cluster; e'|] "" builder
  | "y" -> L.build_call setY_func [|cluster; e'|] "" builder
  | "dx" -> L.build_call setDX_func [|cluster; e'|] "" builder
  | "dy" -> L.build_call setDY_func [|cluster; e'|] "" builder
  | "height" -> L.build_call setHeight_func [|cluster; e'|] "" builder
  | "width" -> L.build_call setWidth_func [|cluster; e'|] "" builder
  | "clr" -> L.build_call setColor_func [|cluster; e'|] "" builder
  | "draw" -> L.build_call setDraw_func [|cluster; e'|] "" builder
  | _ -> raise (Failure ("Property does not exist"))
  )
| A.ArrayAccess(s, e) -> get_array_element s (expr builder e) builder
| A.ArrayInit(typ, e) -> let len = (expr builder e) in init_array typ len builder
| A.ArrayDelete(s) -> L.build_free (L.build_load (lookup s) "" builder) builder
| A.ArrayAssign(s, lhs, rhs) -> set_array_element s (expr builder lhs) (expr builder
rhs) builder
| A.Binop (e1, op, e2) ->
  let e1' = expr builder e1

```

```

and e2' = expr builder e2 in
if (L.type_of e1' = i32_t && L.type_of e2' = i32_t) then
  (match op with
    A.Add      -> L.build_add
  | A.Sub      -> L.build_sub
  | A.Mult     -> L.build_mul
  | A.Div      -> L.build_sdiv
  | A.Mod      -> L.build_srem
  | A.And      -> L.build_and
  | A.Or       -> L.build_or
  | A.Equal    -> L.build_icmp L.Icmp.Eq
  | A.Neq     -> L.build_icmp L.Icmp.Ne
  | A.Less    -> L.build_icmp L.Icmp.Slt
  | A.Leq     -> L.build_icmp L.Icmp.Sle
  | A.Greater -> L.build_icmp L.Icmp.Sgt
  | A.Geq     -> L.build_icmp L.Icmp.Sge
  ) (make_int e1' builder) (make_int e2' builder) "tmp" builder
else if (L.type_of e1' = flt_t || L.type_of e2' = flt_t) then
  (match op with
    A.Add      -> L.build_fadd
  | A.Sub      -> L.build_fsub
  | A.Mult     -> L.build_fmud
  | A.Div      -> L.build_fdiv
  | A.Equal    -> L.build_fcmp L.Fcmp.Oeq
  | A.Neq     -> L.build_fcmp L.Fcmp.One
  | A.Less    -> L.build_fcmp L.Fcmp.Olt
  | A.Leq     -> L.build_fcmp L.Fcmp.Ole
  | A.Greater -> L.build_fcmp L.Fcmp.Ogt
  | A.Geq     -> L.build_fcmp L.Fcmp.Oge
  | _         -> raise (Failure ("incompatible operator-operand for number")) (* Should
never be reached *)
  ) (make_float e1' builder) (make_float e2' builder) "tmp" builder

else
  (match op with
    A.And      -> L.build_and
  | A.Or       -> L.build_or
  | _         -> raise (Failure ("incompatible operator-operand")) (* Should never be
reached *)
  ) e1' e2' "tmp" builder

| A.Unop(op, e) ->
let e' = expr builder e in
  (match op with
    A.Neg      -> L.build_neg
  | A.Not      -> L.build_not
  ) e' "tmp" builder

| A.Assign(s, e) -> let e' = expr builder e in
  if L.type_of (lookup s) = (L.pointer_type i32_t) then
    ignore (L.build_store (make_int e' builder) (lookup s) builder) (* Handle float to
int downcast *)
  else if L.type_of (lookup s) = (L.pointer_type flt_t) then

```

```

        ignore (L.build_store (make_float e' builder) (lookup s) builder) (* Handle int to
float upcast *)
        else
            ignore (L.build_store e' (lookup s) builder) (* Normal assignment for everything
else *)
            ; e' (* Fixes bug in test-func2 *)

| A.Call ("printf", [e]) ->
    L.build_call printf_func [| float_format_str ; make_float (expr builder e) builder
|]
    "printf" builder
| A.Call ("print", [e]) | A.Call ("printb", [e]) ->
    L.build_call printf_func [| int_format_str ; make_int (expr builder e) builder |]
    "printf" builder
| A.Call ("printbig", [e]) ->
    L.build_call printbig_func [| (expr builder e) |] "printbig" builder
(* external function -- for testing *)
| A.Call ("initScreen", [w; h; c]) ->
    let width = expr builder w
    and height = expr builder h
    and color = expr builder c in

    L.build_call initScreen_func [| color; width; height |] "initScreen" builder
| A.Call ("startGame", [w; h; c]) ->
    let width = expr builder w
    and height = expr builder h
    and color = expr builder c in
    (*
        ignore(L.set_alignment 8 color);
    *)

    L.build_call startGame_func [| color; width; height |] "" builder
| A.Call ("quit", []) ->
    L.build_call quitGame_func [||] "" builder
| A.Call ("setFPS", [e]) ->
    let fps = expr builder e in
    L.build_call setFPS_func [| fps |] "" builder
| A.Call ("remove", [c]) ->
    let cluster = expr builder c in
    L.build_call deleteCluster_func [|cluster|] "" builder
| A.Call ("keyDown", [s]) ->
    let keyName = expr builder s in
    L.build_call isKeyDown_func [|keyName|] "keyD" builder
| A.Call ("keyUp", [s]) ->
    let keyName = expr builder s in
    L.build_call isKeyUp_func [|keyName|] "keyU" builder
| A.Call ("keyHeld", [s]) ->
    let keyName = expr builder s in
    L.build_call isKeyHeld_func [|keyName|] "keyH" builder
| A.Call ("random", [e]) ->
    let maxInt = expr builder e in
    L.build_call randomInt_func [|maxInt|] "randInt" builder
| A.Call ("prints", [e]) ->
    L.build_call printf_func [| string_format_str ; (expr builder e) |] "printf" builder
| A.Call (f, act) ->

```

```

    let (fdef, fdecl) = StringMap.find f function_decls in
    let actuals = List.rev (List.map (expr builder) (List.rev act)) in
    let result = (match fdecl.A.typ with A.Void -> ""
    | _ -> f ^ "_result") in
    L.build_call fdef (Array.of_list actuals) result builder
in

(* Invoke "f builder" if the current block doesn't already
   have a terminal (e.g., a branch). *)
let add_terminal builder f =
  match L.block_terminator (L.insertion_block builder) with
Some _ -> ()
| None -> ignore (f builder) in

(* Build the code for the given statement; return the builder for
   the statement's successor *)
let rec stmt builder = function
  A.Block s1 -> List.fold_left stmt builder s1
| A.Expr e -> ignore (expr builder e); builder
| A.Return e -> ignore (match fdecl.A.typ with
  A.Void -> L.build_ret_void builder
| _ -> L.build_ret (expr builder e) builder); builder
| A.If (predicate, then_stmt, else_stmt) ->
  let bool_val = expr builder predicate in
let merge_bb = L.append_block context "merge" the_function in

let then_bb = L.append_block context "then" the_function in
add_terminal (stmt (L.builder_at_end context then_bb) then_stmt)
(L.build_br merge_bb);

let else_bb = L.append_block context "else" the_function in
add_terminal (stmt (L.builder_at_end context else_bb) else_stmt)
(L.build_br merge_bb);

ignore (L.build_cond_br bool_val then_bb else_bb builder);
L.builder_at_end context merge_bb

  | A.While (predicate, body) ->
let pred_bb = L.append_block context "while" the_function in
ignore (L.build_br pred_bb builder);

let body_bb = L.append_block context "while_body" the_function in
add_terminal (stmt (L.builder_at_end context body_bb) body)
(L.build_br pred_bb);

let pred_builder = L.builder_at_end context pred_bb in
let bool_val = expr pred_builder predicate in

let merge_bb = L.append_block context "merge" the_function in
ignore (L.build_cond_br bool_val body_bb merge_bb pred_builder);
L.builder_at_end context merge_bb

  | A.For (e1, e2, e3, body) -> stmt builder

```

```

    ( A.Block [A.Expr e1 ; A.While (e2, A.Block [body ; A.Expr e3]) ] )
in

(* Build the code for each statement in the function *)
let builder = stmt builder (A.Block fdecl.A.body) in

(* Add a return if the last block falls off the end *)
add_terminal builder (match fdecl.A.typ with
  A.Void -> L.build_ret_void
  | t -> L.build_ret (L.const_int (ltype_of_typ t) 0))
in

List.iter build_function_body functions;
the_module

```

4.) semant.ml

```

(* Semantic checking for the Genesis compiler
  Authors:
  - Michael Wang
  - Jason Zhao
  - Sam Cohen
  - Saahil Jain
*)

open Ast

module StringMap = Map.Make(String)

(* Semantic checking of a program. Returns void if successful,
  throws an exception if something is wrong.

  Check each global variable, then check each function *)

let check (globals, functions) =

  (* Raise an exception if the given list has a duplicate *)
  let report_duplicate exceptf list =
    let rec helper = function
      n1 :: n2 :: _ when n1 = n2 -> raise (Failure (exceptf n1))
      | _ :: t -> helper t
      | [] -> ()
    in helper (List.sort compare list)
  in

  let isNum varType = if (varType = Int || varType = Float) then true else false in

  (* Raise an exception if a given binding is to a void type *)
  let check_not_void exceptf = function
    (Void, n) -> raise (Failure (exceptf n))
    | _ -> ()
  in

```

```

(* Raise an exception of the given rvalue type cannot be assigned to
   the given lvalue type. (int can be assigned to float and vice versa) *)
let check_assign lValueType rValueType err =

  if ((isNum lValueType) && (isNum rValueType)) then lValueType
  else if lValueType = rValueType then lValueType else raise err
in

let check_assign_array lval rval err =
  if lval = rval then lval else raise err
in

let is_array_num theType err =
  if(isNum theType) then theType else raise err
in

(**** Checking Global Variables ****)

List.iter (check_not_void (fun n -> "illegal void global " ^ n)) globals;

report_duplicate (fun n -> "duplicate global " ^ n) (List.map snd globals);

(**** Checking Functions ****)

if List.mem "print" (List.map (fun fd -> fd.fname) functions)
then raise (Failure ("function print may not be defined")) else ();

report_duplicate (fun n -> "duplicate function " ^ n)
  (List.map (fun fd -> fd.fname) functions);

(* Function declaration for a named function *)
let built_in_decls = StringMap.add "print"
  { typ = Void; fname = "print"; formals = [(Int, "x")];
    locals = []; body = [] }

  (StringMap.add "printb"
  { typ = Void; fname = "printb"; formals = [(Bool, "x")];
    locals = []; body = [] }

  (StringMap.add "printf1"
  { typ = Void; fname = "printf1"; formals = [(Float, "x")];
    locals = []; body = [] }

  (StringMap.add "keyDown"
  { typ = Bool; fname = "keyDown"; formals = [(String, "keyName")];
    locals = []; body = [] }

  (StringMap.add "keyUp"
  { typ = Bool; fname = "keyUp"; formals = [(String, "keyName")];
    locals = []; body = [] }

  (StringMap.add "keyHeld"

```

```

{ typ = Bool; fname = "keyHeld"; formals = [(String, "keyName")];
  locals = []; body = [] }

  (StringMap.add "initScreen"
  { typ = Int; fname = "initScreen"; formals = [(Int, "width"); (Int, "height"); (Color,
"c")]);
  locals = []; body = [] }

  (StringMap.add "startGame"
  { typ = Void; fname = "startGame"; formals = [(Int, "width"); (Int, "height"); (Color,
"c")]);
  locals = []; body = [] }

  (StringMap.add "quit"
  { typ = Void; fname = "quit"; formals = [];
  locals = []; body = [] }

  (StringMap.add "setFPS"
  { typ = Void; fname = "setFPS"; formals = [(Int, "fps")];
  locals = []; body = [] }

  (StringMap.add "delete"
  { typ = Void; fname = "delete"; formals = [(Cluster, "c")];
  locals = []; body = [] }

  (StringMap.add "random"
  { typ = Int; fname = "random"; formals = [(Int, "max")];
  locals = []; body = [] }

  (StringMap.add "prints"
  { typ = Void; fname = "prints"; formals = [(String, "x")];
  locals = []; body = [] }

  (StringMap.singleton "printbig"
  { typ = Void; fname = "printbig"; formals = [(Int, "x")];
  locals = []; body = [] })))))))))))))
in

(*Add the built-in functions to the function declaration list*)
let function_decls = List.fold_left (fun m fd -> StringMap.add fd.fname fd m)
  built_in_decls functions
in

(*Grab the function declaration given its name*)
let function_decl s = try StringMap.find s function_decls
  with Not_found -> raise (Failure ("Must define the function " ^ s))
in

let _ = function_decl "main" in (* Ensure "main" is defined *)

let _ = function_decl "update" in (* Ensure "update" is defined *)

let _ = function_decl "init" in (* Ensure "init" is defined *)

```



```

let check_function func =

  List.iter (check_not_void (fun n -> "illegal void formal " ^ n ^
    " in " ^ func.fname)) func.formals;

  report_duplicate (fun n -> "duplicate formal " ^ n ^ " in " ^ func.fname)
    (List.map snd func.formals);

  List.iter (check_not_void (fun n -> "illegal void local " ^ n ^
    " in " ^ func.fname)) func.locals;

  report_duplicate (fun n -> "duplicate local " ^ n ^ " in " ^ func.fname)
    (List.map snd func.locals);

  (* Type of each variable (global, formal, or local *)
  let symbols = List.fold_left (fun m (t, n) -> StringMap.add n t m)
    StringMap.empty (globals @ func.formals @ func.locals )
  in

  let type_of_identifier s =
    try let id_typ = StringMap.find s symbols in
      id_typ
    with Not_found -> raise (Failure ("undeclared identifier " ^ s))
  in

  let type_of_identifier_array s =
    try let id_typ = StringMap.find s symbols in
      (match id_typ with
        ArrayType(t) -> t
        | _ -> id_typ
      )
    with Not_found -> raise (Failure ("undeclared identifier " ^ s))
  in

  let verify_array s =
    try let id_typ = StringMap.find s symbols in
      (match id_typ with
        ArrayType(t) -> t
        | _ -> raise (Failure (s ^ " is not an array! What are you doing?"))
      )
    with Not_found -> raise (Failure ("undeclared identifier " ^ s))
  in

  let verify_array_init t =
    (match t with
      Void -> raise (Failure ("The Lord does not allow void arrays..."))
      | _ -> t
    )
  in

  let type_of_property s =
    (match s with

```

```

    | "x" -> Int
    | "y" -> Int
    | "dx" -> Int
    | "dy" -> Int
    | "height" -> Int
    | "width" -> Int
    | "clr" -> Color
    | "draw" -> Bool
    | _ -> raise (Failure ("property \'' ^ s ^ '\' is not defined"))
  )
in

(* Return the type of an expression or throw an exception *)
let rec expr = function
  Literal _ -> Int
  | FloatLit _ -> Float
  | StringLit _ -> String
  | Boollit _ -> Bool

  | ColorLit(r,g,b) -> let t1 = expr r and t2 = expr g and t3 = expr b in
    if (t1 = Int && t2 = Int && t3 = Int) then Color
    else raise (Failure ("expected an int input for type color"))

  | ClusterLit(l,w,x,y,m,n,c) -> let t1 = expr l and t2 = expr w and t3 = expr x and t4 =
expr y and t5 = expr m and t6 = expr n and t7 = expr c in
    if (t1 != Int) then raise (Failure ("expected an int input for type cluster, but you
inputted: " ^ string_of_expr l))
    else
      if (t2 != Int) then raise (Failure ("expected an int input for type cluster, but you
inputted: " ^ string_of_expr w))
      else
        if (t3 != Int) then raise (Failure ("expected an int input for type cluster, but you
inputted: " ^ string_of_expr x))
        else
          if (t4 != Int) then raise (Failure ("expected an int input for type cluster, but you
inputted: " ^ string_of_expr y))
          else
            if (t5 != Int) then raise (Failure ("expected an int input for type cluster, but you
inputted: " ^ string_of_expr m))
            else
              if (t6 != Int) then raise (Failure ("expected an int input for type cluster, but you
inputted: " ^ string_of_expr n))
              else
                if (t7 != Color) then raise (Failure ("expected a color input for type cluster, but you
inputted: " ^ string_of_expr c))
                else
                  Cluster

  | Collision _ -> Bool
  | PropertyAccess (_, s) -> type_of_property s
  | PropertyAssign (_, s, e) -> let eType = expr e and pType = type_of_property s in
    if (eType != pType) then raise (Failure ("Incorrect type for assignment to property "
^ s)) else

```

```

pType
| Property _ -> raise (Failure ("Properties must be associated with an object"))
| Id s -> type_of_identifer s
| ArrayAccess(s, idx) -> let nameType = type_of_identifer_array s and i = expr idx in
  ignore(is_array_num i (Failure ("Thou shall not index an array with a non-number type
(array access)")));
  nameType

| ArrayAssign(s, idx, e) -> let lt = type_of_identifer_array s and rt = expr e and i =
expr idx in
  ignore(is_array_num i (Failure ("Thou shall not index an array with a non-number type
(array assign)")));
  check_assign_array lt rt (Failure ("Thou shall not assign mismatched array types"))

| ArrayInit(t, size) -> let typ = ArrayType(verify_array_init t) and len = expr size in
  ignore(is_array_num len (Failure ("Thou must use a num-type for size when initializing
an array")));
  typ

| ArrayDelete(s) -> verify_array s
| Binop(e1, op, e2) as e -> let t1 = expr e1 and t2 = expr e2
in
(match op with
  Add | Sub | Mult | Div | Mod when t1 = Int && t2 = Int -> Int
| Add | Sub | Mult | Div when isNum t1 && isNum t2 -> Float
| Equal | Neq when t1 = t2 -> Bool
| Less | Leq | Greater | Geq when isNum t1 && isNum t2 -> Bool
| And | Or when t1 = Bool && t2 = Bool -> Bool
| _ -> raise (Failure ("illegal binary operator " ^
  string_of_typ t1 ^ " " ^ string_of_op op ^ " " ^
  string_of_typ t2 ^ " in " ^ string_of_expr e))
)
| Unop(op, e) as ex -> let t = expr e in
  (match op with
    Neg when t = Int -> Int
  | Neg when t = Float -> Float
  | Not when t = Bool -> Bool
  | _ -> raise (Failure ("illegal unary operator " ^ string_of_uop op ^
    string_of_typ t ^ " in " ^ string_of_expr ex)))
  | Noexpr -> Void
  | Assign(var, e) as ex -> let lt = type_of_identifer var
    and rt = expr e in
    check_assign lt rt (Failure ("illegal assignment " ^ string_of_typ lt ^
      " = " ^ string_of_typ rt ^ " in " ^
      string_of_expr ex))
  | Call(fname, actuals) as call -> let fd = function_decl fname in

  (*Check mismatching number of arguments*)
  if List.length actuals != List.length fd.formals then
    raise (Failure ("expecting " ^ string_of_int
      (List.length fd.formals) ^ " arguments in " ^ string_of_expr call))

```

```

(* Iterate over args and check if each of those passed match *)
else
  List.iter2 (fun (ft, _) e -> let et = expr e in
    ignore (check_assign ft et
      (Failure ("illegal actual argument found " ^ string_of_ttyp et ^
        " expected " ^ string_of_ttyp ft ^ " in " ^ string_of_expr e))))
    fd.formals actuals;
  fd.ttyp
in

let check_bool_expr e = if expr e != Bool
  then raise (Failure ("expected Boolean expression in " ^ string_of_expr e))
  else () in

(* Verify a statement or throw an exception *)
let rec stmt = function
  Block s1 -> let rec check_block = function
    [Return _ as s] -> stmt s
  | Return _ :: _ -> raise (Failure "nothing may follow a return")
  | Block s1 :: ss -> check_block (s1 @ ss)
  | s :: ss -> stmt s ; check_block ss
  | [] -> ()
in check_block s1
| Expr e -> ignore (expr e)
| Return e -> let t = expr e in if t = func.ttyp then () else
  raise (Failure ("return gives " ^ string_of_ttyp t ^ " expected " ^
    string_of_ttyp func.ttyp ^ " in " ^ string_of_expr e))

| If(p, b1, b2) -> check_bool_expr p; stmt b1; stmt b2
| For(e1, e2, e3, st) -> ignore (expr e1); check_bool_expr e2;
  ignore (expr e3); stmt st
| While(p, s) -> check_bool_expr p; stmt s
in

stmt (Block func.body)

in
List.iter check_function functions

```

5.) ast.ml

(* Abstract Syntax Tree and functions for printing it

Authors:

- Michael Wang
- Jason Zhao
- Sam Cohen
- Saahil Jain
- Leon Stilwell

*)

```

type op = Add | Sub | Mult | Div | Mod | Equal | Neq | Less | Leq | Greater | Geq | And | Or
type uop = Neg | Not

```

```

type typ =
    Int
  | Float
  | String
  | Bool
  | Void
  | Color
  | Cluster
  | ArrayType of typ

```

```

type bind = typ * string

```

```

type expr =
    Literal of int
  | StringLit of string
  | FloatLit of float
  | BoolLit of bool
  | ColorLit of expr * expr * expr
  | ClusterLit of expr * expr * expr * expr * expr * expr * expr
  | Property of string
  | Collision of expr * expr
  | Id of string
  | Binop of expr * op * expr
  | Unop of uop * expr
  | Assign of string * expr
  | Call of string * expr list
  | PropertyAccess of expr * string
  | PropertyAssign of expr * string * expr
  | ArrayInit of typ * expr
  | ArrayDelete of string
  | ArrayAssign of string * expr * expr
  | ArrayAccess of string * expr
  | Noexpr

```

```

type stmt =
    Block of stmt list
  | Expr of expr
  | Return of expr
  | If of expr * stmt * stmt
  | For of expr * expr * expr * stmt
  | While of expr * stmt

```

```

type func_decl = {
    typ : typ;
    fname : string;
    formals : bind list;
    locals : bind list;
    body : stmt list;
}

```

```

type program = bind list * func_decl list

```

```

(* Pretty-printing functions *)

let string_of_op = function
  Add -> "+"
  | Sub -> "-"
  | Mult -> "*"
  | Div -> "/"
  | Mod -> "%"
  | Equal -> "=="
  | Neq -> "!="
  | Less -> "<"
  | Leq -> "<="
  | Greater -> ">"
  | Geq -> ">="
  | And -> "&&"
  | Or -> "||"

let string_of_uop = function
  Neg -> "-"
  | Not -> "!"

let rec string_of_typ = function
  Int -> "int"
  | Bool -> "bool"
  | String -> "string"
  | Void -> "void"
  | Float -> "float"
  | Color -> "color"
  | Cluster -> "cluster"
  | ArrayType(t) -> "ArrayType:" ^ string_of_typ t

let rec string_of_expr = function
  Literal(l) -> string_of_int l
  | FloatLit(f) -> string_of_float f
  | StringLit(s) -> s
  | BoolLit(true) -> "true"
  | BoolLit(false) -> "false"
  | ColorLit(r,g,b) -> "<" ^ string_of_expr r ^ "," ^ string_of_expr g ^ "," ^ string_of_expr
b ^ ">"
  | ClusterLit(l, w, _, _, _, c) -> "Cluster $" ^ string_of_expr l ^ "," ^ string_of_expr w
^ "," ^ string_of_expr c ^ "$"
  | Collision(c1, c2) -> string_of_expr c1 ^ "@" ^ string_of_expr c2
  | PropertyAccess(e, p) -> string_of_expr e ^ p
  | PropertyAssign(e1, p, e2) -> string_of_expr e1 ^ "." ^ p ^ "=" ^ string_of_expr e2
  | Property(s) -> s
  | Id(s) -> s
  | Binop(e1, o, e2) ->
    string_of_expr e1 ^ " " ^ string_of_op o ^ " " ^ string_of_expr e2
  | Unop(o, e) -> string_of_uop o ^ string_of_expr e
  | Assign(v, e) -> v ^ " = " ^ string_of_expr e
  | Call(f, el) ->
    f ^ "(" ^ String.concat ", " (List.map string_of_expr el) ^ ")"
  | ArrayAccess(s, e) -> s ^ "[" ^ string_of_expr e ^ "]"

```

```

| ArrayAssign(s, e1, e2) ->
  s ^ "[" ^string_of_expr e1 ^"] ="^ string_of_expr e2
| ArrayInit(typ, len) -> string_of_typ typ ^ "[" ^ string_of_expr len ^ "]"
| ArrayDelete(s) -> "delete " ^ s
| Noexpr -> ""

let rec string_of_stmt = function
  Block(stmts) ->
    "{\n" ^ String.concat "" (List.map string_of_stmt stmts) ^ "}\n"
| Expr(expr) -> string_of_expr expr ^ ";\n";
| Return(expr) -> "return " ^ string_of_expr expr ^ ";\n";
| If(e, s, Block([])) -> "if (" ^ string_of_expr e ^ ")\n" ^ string_of_stmt s
| If(e, s1, s2) -> "if (" ^ string_of_expr e ^ ")\n" ^
  string_of_stmt s1 ^ "else\n" ^ string_of_stmt s2
| For(e1, e2, e3, s) ->
  "for (" ^ string_of_expr e1 ^ " ; " ^ string_of_expr e2 ^ " ; " ^
  string_of_expr e3 ^ ") " ^ string_of_stmt s
| While(e, s) -> "while (" ^ string_of_expr e ^ ") " ^ string_of_stmt s

let string_of_vdecl (t, id) = string_of_typ t ^ " " ^ id ^ ";\n"

let string_of_fdecl fdecl =
  string_of_typ fdecl.typ ^ " " ^
  fdecl.fname ^ "(" ^ String.concat ", " (List.map snd fdecl.formals) ^
  ")\n{\n" ^
  String.concat "" (List.map string_of_vdecl fdecl.locals) ^
  String.concat "" (List.map string_of_stmt fdecl.body) ^
  "}\n"

let string_of_program (vars, funcs) =
  String.concat "" (List.map string_of_vdecl vars) ^ "\n" ^
  String.concat "\n" (List.map string_of_fdecl funcs)

```

6.) genesis.c

```

/*
  Authors:
  - Leon Stilwell
  - Michael Wang
  - Jason Zhao
*/

#include <SDL2/SDL.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "genesis.h"

extern void update(int frame);
extern void init();

```

```

int initScreen(color *c, int width, int height);

void static close();
void quitGame();
void clearScreen();
void showDisplay();

// Global vars
int framesPerSec = 60;
const Uint8 *keyStates = NULL;

Uint8 *prevKeyStates = NULL;
char *additionalKeyNames[] = {"Up", "Down", "Left", "Right", "Space", "Escape"};

int keyStateLen;

SDL_Window *gWindow = NULL;
SDL_Renderer *gRenderer = NULL;

cluster_t *clusters = NULL;

int backgroundR = 0xFF;
int backgroundG = 0xFF;
int backgroundB = 0xFF;

int cluster_id = 0;
int frameNum = 0;
int quit = 0;

void setFPS(int fps){
    framesPerSec = fps;
}

void quitGame(){
    quit = 1;
}

// SDL Initialization
int initScreen(color *c, int width, int height){
    // Initialize SDL
    if (SDL_Init( SDL_INIT_VIDEO ) < 0) {
        const char *err = SDL_GetError();
        fprintf(stderr, "SDL could not initialize! SDL Error: %s\n", err);
        return -1;
    }

    // Set texture filtering to linear
    if (!SDL_SetHint(SDL_HINT_RENDER_SCALE_QUALITY, "1")) {
        fprintf(stderr, "Warning: Linear texture filtering not enabled!");
    }

    // Create window
    gWindow = SDL_CreateWindow("Game", SDL_WINDOWPOS_UNDEFINED,

```



```

        SDL_WINDOWPOS_UNDEFINED, width,
        height, SDL_WINDOW_SHOWN);
if (!gWindow) {
    const char *err = SDL_GetError();
    fprintf(stderr, "Window could not be created! SDL Error: %s\n", err);
    return -1;
}

// Create renderer for window
gRenderer = SDL_CreateRenderer(gWindow, -1, SDL_RENDERER_ACCELERATED);
if (!gRenderer) {
    const char *err = SDL_GetError();
    fprintf(stderr, "Renderer could not be created! SDL Error: %s\n", err);
    return -1;
}

// Set background color
backgroundR = c->r;
backgroundG = c->g;
backgroundB = c->b;

// Clear background
clearScreen();
showDisplay();

return 0;
}

//Clears everything drawn on the screen
void clearScreen() {
    SDL_SetRenderDrawColor(gRenderer, backgroundR, backgroundG, backgroundB, 0xFF);
    SDL_RenderClear(gRenderer);
}

void drawRectangle(int x, int y, int w, int h, int r, int g, int b) {
    SDL_Rect rect = { x, y, w, h };
    SDL_SetRenderDrawColor(gRenderer, r, g, b, 0xFF);
    SDL_RenderFillRect(gRenderer, &rect);
}

int inGoalArea(SDL_Rect rect, SDL_Rect goal) {
    int leftBlock = rect.x;
    int rightBlock = rect.x + rect.w;
    int bottomBlock = rect.y;
    int topBlock = rect.y + rect.h;

    int leftGoal = goal.x;
    int rightGoal = goal.x + goal.w;
    int bottomGoal = goal.y;
    int topGoal = goal.y + goal.h;

    if (bottomBlock > bottomGoal && topBlock < topGoal &&
        leftBlock > leftGoal && rightBlock < rightGoal) {

```

```

        return 1;
    }

    return 0;
}

void showDisplay(){
    SDL_RenderPresent(gRenderer);
}

// Must be called when program finishes
void static close(){
    SDL_DestroyRenderer(gRenderer);
    SDL_DestroyWindow(gWindow);
    gWindow = NULL;
    gRenderer = NULL;

    SDL_Quit();

    free(prevKeyStates);

    cluster_t *temp, *currentCluster;
    HASH_ITER(hh,clusters,currentCluster,temp) {
        HASH_DEL(clusters,currentCluster);
        free(currentCluster);
    }
}

void pollEvents(){
    int bytes = keyStateLen * sizeof(*keyStates);
    memcpy(prevKeyStates, keyStates, bytes);

    SDL_Event event;
    while(SDL_PollEvent(&event)) {
        if (event.type == SDL_QUIT) {
            quit = 1;
        }
    }
}

bool isKeyHeld(char *key) {
    SDL_Scancode code = SDL_GetScancodeFromName(key);
    return keyStates[code] && prevKeyStates[code];
}

bool isKeyDown(char *key) {
    SDL_Scancode code = SDL_GetScancodeFromName(key);
    return keyStates[code] && (!prevKeyStates[code]);
}

bool isKeyUp(char *key) {
    SDL_Scancode code = SDL_GetScancodeFromName(key);
    return (!keyStates[code]) && prevKeyStates[code];
}

```

```

}

void startGame(color *c, int width, int height) {
    quit = 0;
    srand(time(NULL));
    initScreen(c, width, height);

    // Update screen
    showDisplay();
    keyStates = SDL_GetKeyboardState(&keyStateLen);
    prevKeyStates = malloc(keyStateLen * sizeof(Uint8));

    // Initialize any user defined data
    init();

    // Main loop - ie. run the game
    while (!quit) {
        int msPerFrame = (int) (1000 / framesPerSec);

        frameNum += 1;
        unsigned int frameStart = SDL_GetTicks();

        // Check for user input
        pollEvents();

        // Redraw the screen
        clearScreen();

        // Draws all clusters in hash
        cluster_t *cl;

        for (cl = clusters; cl != NULL; cl = cl->hh.next) {
            if (cl->draw == 1) {
                int x = cl->x + cl->dx; int y = cl->y + cl->dy;
                int h = cl->height; int w = cl->width;
                int r = cl->color.r; int g = cl->color.g; int b = cl->color.b;
                //update cluster x,y
                cl->x = x; cl->y = y;
                drawRectangle(x, y, h, w, r, g, b);
            }
        }

        update(frameNum);
        showDisplay();

        /* Cap the rate of execution of loop */
        unsigned int frameTime = SDL_GetTicks() - frameStart;
        if (frameTime < msPerFrame) {
            SDL_Delay(msPerFrame - frameTime);
        }
    }

    // Tear down

```

```

    close();
}

// Called from newCluster. DO NOT CALL OTHERWISE
int create_id() {
    int temp = cluster_id;
    cluster_id = cluster_id + 1;

    return temp;
}

int randomInt(int max) {
    return (rand() % max);
}

int newCluster(int length, int width, int x, int y, int dx, int dy, color *color) {
    cluster_t *cluster = malloc(sizeof(cluster_t));
    cluster->height = length;
    cluster->width = width;
    cluster->x = x;
    cluster->y = y;
    cluster->dx = dx;
    cluster->dy = dy;
    cluster->color = *color;
    cluster->id = create_id();
    cluster->draw = 1;

    HASH_ADD_INT(clusters, id, cluster);

    unsigned int numClusters;
    numClusters = HASH_COUNT(clusters);

#ifdef DEBUG
    printf("There are %u clusters\n", numClusters);
#endif

    return cluster->id;
}

// Returns pointer to cluster with ID = id if exists, else
// return null, basically a wrapper around HASH_FIND_INT()
cluster_t * getCluster(int id) {
    cluster_t *p;
    HASH_FIND_INT(clusters, &id, p);
    return p;
}

void setDraw(int id, bool b) { cluster_t *c = getCluster(id); if(c) { c->draw = b; } }
void setX(int id, int x) { cluster_t *c = getCluster(id); if(c) { c->x = x; } }
void setY(int id, int y) { cluster_t *c = getCluster(id); if(c) { c->y = y; } }
void setDX(int id, int dx) { cluster_t *c = getCluster(id); if(c) { c->dx = dx; } }
void setDY(int id, int dy) { cluster_t *c = getCluster(id); if(c) { c->dy = dy; } }

```

```

void setColor(int id, struct color *clr) { cluster_t *c = getCluster(id); if(c) { c->color =
*clr; } }

int getDraw(int id)    { cluster_t *c = getCluster(id); if (c) { return c->draw; } else {
return -1; } }
int getX(int id)      { cluster_t *c = getCluster(id); if (c) { return c->x; } else {
return -1; } }
int getY(int id)      { cluster_t *c = getCluster(id); if (c) { return c->y; } else {
return -1; } }
int getDX(int id)     { cluster_t *c = getCluster(id); if (c) { return c->dx; } else {
return 0; } }
int getDY(int id)     { cluster_t *c = getCluster(id); if (c) { return c->dy; } else {
return 0; } }
int getHeight(int id) { cluster_t *c = getCluster(id); if (c) { return c->height; } else {
return -1; } }
int getWidth(int id)  { cluster_t *c = getCluster(id); if (c) { return c->width; } else {
return -1; } }
color * getColor(int id) { cluster_t *c = getCluster(id); if (c) { return &(c->color); } else
{ return NULL; } }

void deleteCluster(int id){
    cluster_t *c = getCluster(id);
    if(c){ HASH_DEL(clusters, c); }
}

bool detectCollision(int id1, int id2){
    cluster_t *c1 = getCluster(id1);
    cluster_t *c2 = getCluster(id2);

    if (c1 && c2) {
        SDL_Rect r1, r2;
        r1.x = c1->x; r1.y = c1->y;
        r1.w = c1->width; r1.h = c1->height;
        r2.x = c2->x; r2.y = c2->y;
        r2.w = c2->width; r2.h = c2->height;

        const SDL_Rect *r3 = &r1;
        const SDL_Rect *r4 = &r2;

        SDL_Rect res;
        SDL_bool ans;
        ans = SDL_IntersectRect(r3, r4, &res);

        if(ans == SDL_TRUE){
            return 1;
        }

        return 0;
    }

    return 0;
}

```

```

#ifdef SKIP_MAIN
int main(int argc, char* args[]){
    struct color col;
    col.r = 0;
    col.g = 0;
    col.b = 0;

    struct color col2;
    col2.r = 0xFF;
    col2.g = 0;
    col2.b = 0;
    //Make new screen
    cluster_t *c = NULL;
    c = malloc(sizeof(cluster_t));
    c->height = 50;
    c->width = 50;
    c->x = 50;
    c->y = 100;
    c->dx = 0;
    c->dy = 0;
    c->color = col2;
    c->id = 0;

    HASH_ADD_INT(clusters,id,c);
    struct color *colptr = &col;
    startGame(colptr, 640, 480);
}
#endif

```

7.) genesis.h

```

#ifdef GENESIS_H
#define GENESIS_H

#include "uthash/include/uthash.h"
#include "uthash/include/utlist.h"

#include <stdbool.h>

typedef struct color {
    int r;
    int g;
    int b;
} color;

typedef struct position {
    int x;
    int y;
} position_t;

//what is our implementation here?

```

```

typedef struct cluster{
    int x;
    int y;
    int dx;
    int dy;
    color color;
    int height;
    int width;
    const char *name;
    int id;
    bool draw;
    UT_hash_handle hh;
} cluster_t;

typedef struct board {
    const char* name;
    color color;
    int height;
    int width;
    cluster_t *clusters;
    UT_hash_handle hh;

} board_t;

void add_Cluster(int length, int width, int x, int y, int dx, int dy, color *color);
void remove_Cluster(int id);
int create_id();
#endif

```