**CODE NOTES:**

**Sonic Pi**

When you have a “do” you have to have a matching “end” or program will not know when to stop.

Example:

Get Eggs (number\_eggs)

Function(parameter)

“else” – when you add on “if” remove the “e” like “elsif”on

== - for the comparison (when not a < or >)

SLEEP – just a pause

Has to start with “if” then as many “elsif” ‘s as you like

You are naming spaces in RAM

**Ruby**

IRB: Stands for Interactive Ruby Environment – gives you the ability to play around with your code or to see what the algorithm should be if you’re stuck

Command + or Command – to increase of decrease font size

Need “ “ if you to print a word

Command S is save as

Command / - puts a # in

% - modulus takes the number and divide by number on the right of it and leaves a remainder, i.e., x % 3 == 0 means number is divisible by 3 with no remainder.

If no remainder, Ruby doesn’t need the “()” but most languages do require.

Def = define a function

If

Elsif

Else

Puts will print to screen

Can insert “then” before puts, but in Ruby it is implied.

If you type a “.”, no space!!!

Array - a collection of things; items get pushed to arrays from the back; something that stores info

An object that contains anything: integer, number, letter, etc.

Method and function are interchangeable

Sudo gem install mini test – a way to make code so anyone can read it

Require “minitest/autorun”

Class Test “filename” < Minitest::Test

Anything under class has to start w/test if you want assertions

Class names – upper case

Function names – lower case

Ruby runs as a script top to bottom

|| “or” - also called goal posts; shift \ key

.inspect – puts the file into a human readable form

Object, variable, array, hash – something that holds data

Method - does something to the object: ex. Chomp, push, etc.

Object = NOUN, Method = VERB

always test in all upper case when you type a letter; computer searches an image

.upcase - primarily use as other languages don’t recognize; 3 below are mostly Ruby

.downcase

.captialize

.swapcase

Argument – what we pass in to functions

{ do

}end

array.map do|x| - take array and do something w/each array

array.pop – takes out last element of array

array.clear – clears the array

array.delete\_at(-1) – would take out the last element in the array for example

<< can add to array or puts something in the array

rindex array=[51,20] – first number is starting position; second is how many positions after

Example: hundred\_array = Array(1..100)

hundred\_array = Array[\*(1..100)]

hundred\_array.delete\_if{|x| x,50} - would delete everything less than 50

[ ] returns the position of item in array

array.slice – returns an element at index you specify

! – makes more permanent if placed after a command, as in array.map!

array.unshift – reverses items in array

.at – is similar to the bracket method

multidimensional array – array within an array

array.flatten – makes a multi-dimensional array one array

float – a number with a decimal

%w – before bracket in array as in array=%w[a,b,c,d]

array.insert(position,”word”) – you could insert an object in selected index in an array

array.each\_with\_index.do |value,index\_position]

array << value + index\_position.to\_s end

.each do – will check each object

object or variable => indicates what you want it to be

gets.chomp gets rid of blank space after answer

array = %w [ ] – space between every object; no quotes, no commas

split.string – an array of characters

**HASHES:**

An array is an ordered collection indexed by its placement like first, second, etc.; if order is important, use an array

Hash is an object index instead of an integer index as an array

Integer – number

Hashes – unordered object, indexed collection of objects; when order is not important; indexed by object key-value pairs; refer to by label instead of placement; no particular order

Key – 1 object that references a second object, the value

So, hashes are object indexed instead of integer indexed

Hash is like a dictionary

Key is to the left of => and value is to the right of the =>

Example:

Person = [‘Dolly’, ‘Throckmorton’,’female’,’brown’,brunette’]

Person = {} – hash = curly braces

Person = {‘first\_name’ => ‘Dolly’, ‘last\_name’ => ‘Throckmorton’, ‘gender’ => ‘female’, ‘eyes’ = > ‘brown’, ‘hair’ => ‘brunette’}

Person.index(‘Dolly’) should produce “first\_

Hash syntax:

Me = {“first name”=>”Dolly”, “age” => 51}

Me [“first name”]

* “Dolly”

Me2 = {:firstname => “Dolly”, age => 51}

Me3 = {firstname: “Dolly”, age: 51}

Me2 = [:firstname]

* “Dolly”

Me 3 = {firstname:}

* “Dolly”

mixed.keys – returns the keys

mixed.values – returns the values

mixed.to\_a – turn to arrays

mixed.length or mixed.size – basically the same

Skipping tests: after your function: skip (“work in progress”)

**Other Info**

Gmail – [dollygthrock@gmail.com](mailto:dollygthrock@gmail.com)

DbGt1965$

GitHub = dollygthrock

DbGt1965$

Git commit (only on your computer)

Git remote add origian

Git push (anyone can see in the cloud; local computer to github)

Create a new repository

Open your terminal window and navigate to the folder you wish to push to github

Git init

Git add .

Git commit –m “Initial commit”

Git config –global --edit

Git commit –amend --reset-author

mk directory

Slack dollygthrock our group is Minded-Minds-Training

**CSV File**: comma separated variable file

When you open a new csv file, Ruby will automatically create it; you don’t need to create a new separate file

Can’t write to excel file when it’s open and you run your csv file

**SINATRA** – Ruby gem

Rest Commands:

Get – retrieve data from the server; go to the client the root of the system

Post - send data to the server

Delete

Can only have 1 Get and 1 Post with the same name

***ERB*** – embedded Ruby; function to call from your **Views** folder; template library – get me the template after the **:** (colon); html file; app.rb – your main Ruby file;

Locals – can pass as many key/value pairs as we want

Fonts: h1 is larger heading; h2 is smaller, etc.

If you change your .rb file, you have to restart Sinatra, but no restart needed if you change your .erb file only

Everything is a string in html: % % - in between means evaluate Ruby code

@variable - global variable: can be dangerous

@@ - class variables which get shared through class hierarchy. Requiring Sinatra makes our file a subclass of Sinatra. When we make a Get or a Post call, we get a new instance of the class.

**Sessions** are cookies – they allow us to store the data between html calls (gets, posts, etc)

Helped with our global variable issue = @name

enable :sessions

Example:

get ‘/’ do

session [:name]

end

Class – like a hash: multiple pieces of info that move around together

Object-oriented: opposite is functionality

Erb – call from post/input; send in multiple functions

Web Apps:

Sinatra gem – local host 4567: Sinatra is doing this default port assignment for us

HTML: <% to open ruby code%>

Sudo gem installer

Heroku – to place on the web; Heroku (space) create “ “ – within quotes must be lower case, numbers and dashes only and can only use once

Bundle install – figures out all the versions

Gemfile.lock – open it using sublime text

Once for each: Gemfile, Procfile

Get – gets info from the user or client

Post – once it gets the info from the user or client, it posts it to the server

Minimax algorithm – game states: give each move a numberical value based on the board state: Win = 1, Draw = 0, Lose = -1 giving win the highest rank, lose the lowest, and draw somewhere in between; so you want AI to choose the move with the highest ran