My idea

Use this to summarize your idea, plan it using sketches, notes and pseudocode as needed

| A shooting gallery style game using caricature creatures, specifically a “Duck”, a “Bear”, and a “Cat”. The idea being to have the user place the appropriate “Hat” on the corresponding creature, specifically to match an indicated “Goal” creature which will be displayed in a section of the UI. |
| --- |

Where will the inventory skills be demonstrated? List every one to be sure you’ve included them.

|  | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Milestone 1** | **Milestone 2** | | **Milestone 3** | | **Milestone 4** | |
| Basic Setup | Refining functions in classes | |  | |  | |
| 1 line, ellipse, rect… | 6 contstrain | | 17 Nested Loop | | 36 ArrayList size, get, remove,… | |
| 2 fill, stroke, strokeWeight… | 7 keyPressed, mousePressed,… | | 18 Break | | 37 PVector use? |  |
| 3 Modes | 11 println, stop, | | 20 Function w/ No Para or Return | | 41 Random 2D Vector | |
| 4 setup and draw | 15 switch | | 21 Function w/ Return | | 42 Normalized Vector? |  |
| 5 background, random, noise | 16 For loop, While loop | | 22 Diff Param and Arguments |  | 43 New Method in PVector | |
| 8 ++, +=, --, -=, \*=, /= | 19 Diff for loop and while loop |  | 23 Function w/ simple Argument | | 44 Timer |  |
| 9 local variable | 25 Diff class and object |  | 24 Function w/ object Argument | | 45 Game States |  |
| 10 global variable | 26 Constructor function use |  | 29 New Object creation | | 46 Button Roll-Over change |  |
| 12 if, else if, else | 27 Processing Tabs |  | 30 Constructor w/ Param | | 47 Drag & Drop |  |
| 13 ==, >=, <=, >, … | 31 Diff Array and ArrayList |  | 32 Backwards list use? |  | 48 Sprite Sheet animation |  |
| 14 &&, || | 39 Basic Physics | | 33 Populate Array | | 49 Collision |  |
| 28 Class & Constructor use | 40 Find Direction and Distance | | 34 Populate ArrayList | | ~~50 Game Control Plus~~ |  |
| 38 PVector |  | | 35 Manage Object in Array or List | |  | |
| Yellow cells are questions  The small box to the right is green when the question is answered. | Need clarification on 39 and 40 | | List populating and lives handling. | | Please Note Grey Boxes are not guaranteed to be implemented | |

Answers:

**19 What’s the difference between a for loop and a while loop?:** November 28th

A for loop uses variables specific to it, which cannot be used outside the loop. The for loop repeats its loop until the input value returns false based on the input conditional. The input is automatically adjusted after each successful loop by a specified amount included in the initial loop statement (an equation typically). A while loop repeats its loop until the input conditional returns as false. The while loop needs predefined global variables and does not automatically adjust any variables automatically, and will continuously run the code within its interior until the conditional returns false. If the input conditional’s variables are not updated inside the loop, the conditional will never change and the loop will never end.

Example:

for (x = 0; x < 20; x+2) {println(x);}

The above will cause the loop to return 11 numbers printed to the output like the table on the right, but will return an error if the user tries to print x after the loop is completed due to X not being a global variable.

| Loop output |
| --- |
| 0 |
| 2 |
| 4 |
| 6 |
| 8 |
| 10 |
| 12 |
| 14 |
| 16 |
| 18 |
| 20 |

int x = 0;

while ( x < 20) {println(x);

x = x+2;}

The above will cause the loop to return the same numbers, and will still return 20 if asked to print x after the loop is over. If however x = x+2; is not included, the loop will always print 0 and will never end.

**22 What’s the difference between parameters and arguments?:** December 13th

Parameters are the values used when a function is defined, they set the names and value types used in the function. Arguments are the values actually being passed when the function is called. Typically they are global or local variables.

**25 What’s the difference between a class and an object?:** November 28th

A class is all the code involved within an object. The class is used to identify any functions or variables involved. An object is a single instance of a class, and must have its own unique identifying name.

Example: A class of animals will include all of their Biological identifiers such as number of feet, and if they are amphibious. An object would be a red-tree frog, which is a single instance of the animal class, and has the identifier that it is amphibious.

**26 What is a constructor function? What does it do any when?:** November 28th

A constructor function is used inside a class to set up the default variable states for an object from the class when it is made. The constructor function is called automatically when a new object of the class is made.

In the example to the right the object sparked uses the constructor function within the Sparkles class to define its colour as red, and its position on the screen as (2, 5). Without the constructor function on the 5th to 9, processing would have no way of knowing what the colour or location of the sparkle is, just that the sparked object exists.

| Sparkles sparked = new Sparkles(red, 2, 5);  class Sparkles{  color colour;  PVector position;  Sparkles (color c, float x, float y){  colour = c;  position.x = x;  position.y = y  }  } |
| --- |

**27 Why should each class have its own tab in processing?:** November 28th

Using separate tabs for each class in processing makes it easier to identify which code belongs to which class, as well as making the main tab of the program easier to read. Processing will still consider any tabs to be part of the program, simply functioning as code that would be added on to the end of anything in the main tab.

**31 What’s the difference between an array and an ArrayList?:** November 28th

Arrays are a single list that tracks simple types such as char, int, float, or even color. Arrays can be modified in length in processing using built in functions such as append(), shorten(), and expand(). ArrayLists are arrays that can be modified in length, and are used for objects. Entries in an arraylist can be easily added or deleted using the built-in functions add() and remove() respectively, but to access objects in an array the developer must use get() on the list to return the specific object.

**32 Why would you want to go through a list backwards, decrementing the index?:** December 6th

Looping through a list backwards lets the list be updated with new values without losing the previous value. I.E. if you want to track where a shape has been you would change [5] to become [4]’s values (2, 3) [4] becomes [3] (2,2) and so on and so forth until the value 0 is hit, which is then updated with the current location before the shape moves. If looped through forwards the values would all just become the same with [1] taking (0,0) which then feeds it to [2] which feeds to [3] and so on making the list essentially wipe itself.

**37 When should you use PVector instead of float variables?** November 28th

PVector is good for when a location, direction, or speed needs to be recorded. PVectors record 2 - 3 floats together as a sort of simple object. This results in those floats always being associated with that object, and results in less variables needing to be declared.

**42 What is a normalized vector, why is it useful?:** November 28th

First brief explanation of magnitude: magnitude is the length of a vector, or the float distance of the vector from the origin point (also known as (0,0)) to the vector location.

Normalizing a vector is to reduce a vector to a length or magnitude of 1 while still keeping its direction. This allows the vector to be scaled easily.