# Independent Study Project – Checkpoint 1

## Purpose

To create a product that engages you and that you would be proud to share to a public audience.

Along the way, you will develop your ability to problem-solve using a variety of strategies, to implement a solution in code, to manage source code using accepted industry practices, and to plan and meet commitments for project milestones.

## Evaluation

As described in January, I am now taking a standards-based approach to evaluating your progress in the course.

What does that mean?

It means that I value the *process* of your work on this ISP as much as your *product.*

It means that I am looking, quite simply, for you to provide evidence of having met the expectations listed.

To that end: using your commits on GitHub, and your posts on Sesame, how would *you* evaluate your progress so far?

You probably will not have yet demonstrated *all* of the expectations, but have you hit some? How often?

For each expectation shown on the following pages:

1. Provide links(s), optionally with brief explanatory text to specific parts of a commit in your source control history
2. Give yourself a 1 to 5 star rating

## Curriculum Expectations

### A1. Data Types and Expressions Demonstrate the ability to use different data types, including one-dimensional arrays, in computer programs;

**A1.1** use constants and variables, including integers, floating points, strings, and Boolean values, correctly in computer programs;

ASCII, Unicode) to internally represent data and store information;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/RSGC-Sheeres-A/Pipes-Game/blob/6f44fc69c1e3e942fe00633839b868953e0a5dca/Pipes/Level_1.swift#L15-L31>  This shows that I’m able to use changing variables because there’s a variable that’s value increases as the for-loop runs. Also, it shows I can assign values to integers because of how I assigned specific characteristics to sprite nodes. |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A1.3** use assignment statements correctly with both arithmetic and string expressions in computer programs;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/RSGC-Sheeres-A/Pipes-Game/blob/5cceb9b3677703ab4a44517a5194323b0f543e81/Pipes/LevelSelect.swift#L76-L80>  This example shows I am able to use let statements correctly because I assigned a transition using a let statement as well as changed the scene using a let statement. |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A1.4** demonstrate the ability to use Boolean operators (e.g., AND, OR, NOT), comparison operators (i.e., equal to, not equal to, greater than, less than, greater than or equal to, less than or equal to), arithmetic operators (e.g., addition, subtraction, multiplication, division, exponentiation, parentheses), and order of operations correctly in computer programs;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/RSGC-Sheeres-A/Pipes-Game/blob/e78431004352397f949f21b77b56202bb9aeb55b/Pipes/LevelSelect.swift#L71-L75>  This is evidence of how I was able to use arithmetic operators by using a “+”. Also, it is an example of how I used a comparison operator; “==”. |
| <https://github.com/RSGC-Sheeres-A/Pipes-Game/blob/5cceb9b3677703ab4a44517a5194323b0f543e81/Pipes/TitlePage.swift#L51-L62>  This is an example of how I was able to use an AND Boolean operator. There are three separate instance of the AND Boolean operator in this link. |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A1.5** describe the structure of one-dimensional arrays and related concepts, including elements, indexes, and bounds;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/RSGC-Sheeres-A/Pipes-Game/blob/e78431004352397f949f21b77b56202bb9aeb55b/Pipes/LevelSelect.swift#L71-L73>  This is an example of how I used an array. In this instance, I’m calling the second cell of the array to get the number that is in it. The first cell, cell 0, is used to store the word that identifies the object (a box). |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A1.6** write programs that declare, initialize, modify, and access one-dimensional arrays.

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/RSGC-Sheeres-A/Pipes-Game/blob/8200cc139cdb4aecd432ca2523cea038db501713/Pipes/Level_1.swift#L18-L21> I used an array here to set the values of the X and Y coordinates of the individual circles. |
| <https://github.com/RSGC-Sheeres-A/Pipes-Game/blob/8200cc139cdb4aecd432ca2523cea038db501713/Pipes/Level_1.swift#L50-L139> For all 10 of these circles, I was able to access the array from the previous link to get the X and Y coordinates to add the circles to those locations. |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

### A2. Control Structures and Simple Algorithms Demonstrate the ability to use control structures and simple algorithms in computer programs;

**A2.1** write programs that incorporate user input, processing, and screen output;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/RSGC-Sheeres-A/Pipes-Game/blob/e78431004352397f949f21b77b56202bb9aeb55b/Pipes/LevelSelect.swift#L56-L77>  This link shows that I am able to incorporate user input, because I have the user click on something and check what the user has clicked on. It also is an example of screen output, because it changes the scene to a different one using a transition. However, it doesn’t process user data and output the processed data because my game is not complete yet. That is actually one of the aspects im struggling with right now. |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A2.2** use sequence (order in code), selection, and repetition (loops) control structures to create programming solutions;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/RSGC-Sheeres-A/Pipes-Game/blob/e78431004352397f949f21b77b56202bb9aeb55b/Pipes/LevelSelect.swift#L34-L35>  This is an example of how I used sequence, because I wanted the grid these for loops create to start at 1 in the top left corner and finish in the bottom right. By using this order of the Y value as the first for loop and the X as the second, it allowed me to create a grid in the order I wanted it to be in. |
| <https://github.com/RSGC-Sheeres-A/Pipes-Game/blob/e78431004352397f949f21b77b56202bb9aeb55b/Pipes/LevelSelect.swift#L57-L76>  this is evidence of how I am able to use repetition loops because this is a large block of loops. |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A2.3** write algorithms with nested structures (e.g., to count elements in an array, calculate a total, find highest or lowest value, or perform a linear search).

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
| <https://github.com/RSGC-Sheeres-A/Pipes-Game/blob/e78431004352397f949f21b77b56202bb9aeb55b/Pipes/LevelSelect.swift#L57-L76>  This algorithm allows the computer to check what block the user has clicked on. It is important because it brings the user to that level. |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

### A3. Subprograms Demonstrate the ability to use subprograms within computer programs;

**A3.1** demonstrate the ability to use existing sub-programs (e.g., random number generator, substring, absolute value) within computer programs;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
|  |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A3.2** write subprograms (e.g., functions, procedures) that use parameter passing and appropriate variable scope (e.g., local, global), to perform tasks within programs.

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
|  |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

### A4. Code Maintenance Use proper code maintenance techniques and conventions when creating computer programs.

**A4.1** demonstrate the ability to identify and correct syntax, logic, and run-time errors in computer programs;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
|  |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A4.2** use workplace and professional conventions (e.g., naming, indenting, commenting) correctly to write programs and internal documentation;   
 (also includes use of source control)

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
|  |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A4.3** demonstrate the ability to interpret error messages displayed by programming tools (e.g., compiler, debugging tool), at different times during the software development process (e.g., writing, compilation, testing);

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
|  |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A4.4** use a tracing technique to understand program flow and to identify and correct logic and run-time errors in computer programs;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
|  |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**A4.5** demonstrate the ability to validate a program using a full range of test cases.

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
|  |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

### B1. Problem-solving Strategies Use a variety of problem-solving strategies to solve different types of problems independently and as part of a team;

**B1.1** use various problem-solving strategies (e.g., stepwise refinement, divide and conquer, working backwards, examples, extreme cases, tables and charts, trial and error) when solving different types of problems;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
|  |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

**B1.2** demonstrate the ability to solve problems independently and as part of a team;

| Evidence: provide link(s) where possible, optionally provide brief explanatory text, add rows as needed |
| --- |
|  |
|  |

**Overall rating on this standard**: ✩ ✩ ✩ ✩ ✩

## Comments and Proposal for Level of Achievement

Understanding that this is a checkpoint 1/3 of the way into the ISP, and that mastery of all standards is not expected at this point in time, what do you suggest as your current level of achievement? Why?