# 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-blum-ma/Blum-ISP/blob/456332013a5e5b7bfffd8712b1130fe187d5e668/CS:GO%20Map%20layout/GameScene.swift#L20-L74>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene.swift#L94-L117>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene2.swift#L28-L112> |
|  |

**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-blum-ma/Blum-ISP/blob/456332013a5e5b7bfffd8712b1130fe187d5e668/CS:GO%20Map%20layout/GameScene.swift#L16>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/456332013a5e5b7bfffd8712b1130fe187d5e668/CS:GO%20Map%20layout/GameScene.swift#L16>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/456332013a5e5b7bfffd8712b1130fe187d5e668/CS:GO%20Map%20layout/GameScene.swift#L103>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene2.swift#L50>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene2.swift#L70-L74>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene2.swift#L93-L95>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene2.swift#L41>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene.swift#L94-L104> |
|  |

**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 |
| --- |
| n/a |
|  |

**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-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene.swift#L90-L94>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene2.swift#L68-L74> |
|  |

**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 |
| --- |
| n/a |
| / |

**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-blum-ma/Blum-ISP/blob/456332013a5e5b7bfffd8712b1130fe187d5e668/CS:GO%20Map%20layout/GameScene.swift#L100-L119>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene2.swift#L68-L117> |
|  |

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

**A2.2** use sequence, selection, and repetition 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-blum-ma/Blum-ISP/blob/456332013a5e5b7bfffd8712b1130fe187d5e668/CS:GO%20Map%20layout/GameScene.swift#L103-L107>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/456332013a5e5b7bfffd8712b1130fe187d5e668/CS:GO%20Map%20layout/GameScene.swift#L13-L16>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene2.swift#L95-L110> |
|  |

**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 |
| --- |
| n/a |
|  |

**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 |
| --- |
| <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene2.swift#L111>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene2.swift#L74-L84> |
|  |

**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 |
| --- |
| <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene2.swift#L20>  <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene.swift#L17-Lundefined> |
|  |

**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 |
| --- |
| Used comments to make code more readable and organized  Ex: <https://github.com/rsgc-blum-ma/Blum-ISP/blob/master/CS:GO%20Map%20layout/GameScene.swift#L90-L117>  Comments make it easier to understand what certain code does if the user doesn’t remember. |
|  |

**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 |
| --- |
| Learned how to properly use the debugger in XCode and understand why you would want to use it. Also learned more about breakpoints and why they exist. |
|  |

**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 |
| --- |
| The debugger was one of the key tools in which I developed a larger understanding of. It allowed me to run through brackets in my code and easily determine where my code wasn’t being read properly or even read at all. |
|  |

**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 |
| --- |
| n/a |
|  |

**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 |
| --- |
| 1st: <https://github.com/rsgc-blum-ma/Blum-ISP/blob/456332013a5e5b7bfffd8712b1130fe187d5e668/CS:GO%20Map%20layout/GameScene.swift#L88-L111>  2nd: <https://github.com/rsgc-blum-ma/Blum-ISP/blob/413b0be36d4cec220b7a8917fd438dd0be904af9/CS:GO%20Map%20layout/GameScene.swift#L94-L117>  final: <https://github.com/rsgc-blum-ma/Blum-ISP/blob/b0c954ba3b341479a22bcc879205a7be1bc034e0/CS:GO%20Map%20layout/GameScene.swift#L90-L117>  This was a section of my code which developed more and more after every commit. This occurred due to the several issues which occurred in my code kept leading back to this area. Solving each issue was accomplished through strategies as simple as trial and error and more complex strategies like working backwards or stepwise refinement. Although at the end, I was successful in creating an efficient well-structured section of code. |
|  |

**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 |
| --- |
| Before: <https://github.com/rsgc-blum-ma/Blum-ISP/blob/456332013a5e5b7bfffd8712b1130fe187d5e668/CS:GO%20Map%20layout/GameScene.swift#L9-L119>  After: <https://github.com/rsgc-blum-ma/Blum-ISP/blob/44a3881c737bc3e4423bff3ee7f92c7925f73808/CS:GO%20Map%20layout/GameScene.swift#L94-L112>  As you can see, using previously taught methods like using terminator to print a variable in the terminal, or using the Debugger I managed to independently figure out why my hitboxes weren’t working properly and managed to fix it. |
|  |

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

## Comments and Proposal for Level of Achievement

Understanding that this is a checkpoint 1/2 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?

I believe that despite the actual lack of material, I believe that I have spent a fair amount of time understanding how to use certain tools to my advantage and being able to understand what is necessary for my final product. Therefore, I believe for the final checkpoint, I will be able to accomplish much more due to my gained knowledge on using XCode efficiently.