# 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-Longwell-J/Fighter/blob/dc3b016984e53e816fbee2b4ad554ea28e1e8d23/Fighter/GameOverScreen.swift#L16-L26>  <https://github.com/RSGC-Longwell-J/Fighter/blob/9dfa1f88d7951cfd0590066d97623cb7f2d14e36/Fighter/GameScene.swift#L21-Lundefined> |
|  |

**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-Longwell-J/Fighter/blob/1d38e9248f97dc4db185a5fe3bc0a16deab19b75/Fighter/GameScene.swift#L28-L33>  https://github.com/RSGC-Longwell-J/Fighter/blob/9dfa1f88d7951cfd0590066d97623cb7f2d14e36/Fighter/GameScene.swift#L21-Lundefined |
|  |

**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-Longwell-J/Fighter/blob/9dfa1f88d7951cfd0590066d97623cb7f2d14e36/Fighter/GameScene.swift#L111-L112>  <https://github.com/RSGC-Longwell-J/Fighter/blob/9dfa1f88d7951cfd0590066d97623cb7f2d14e36/Fighter/GameScene.swift#L97-L98>  <https://github.com/RSGC-Longwell-J/Fighter/blob/9dfa1f88d7951cfd0590066d97623cb7f2d14e36/Fighter/GameScene.swift#L148> |
|  |

**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-Longwell-J/Fighter/blob/9dfa1f88d7951cfd0590066d97623cb7f2d14e36/Fighter/GameScene.swift#L146>  <https://github.com/RSGC-Longwell-J/Fighter/blob/9dfa1f88d7951cfd0590066d97623cb7f2d14e36/Fighter/GameScene.swift#L129>  <https://github.com/RSGC-Longwell-J/Fighter/blob/9dfa1f88d7951cfd0590066d97623cb7f2d14e36/Fighter/GameScene.swift#L54-L68> |
|  |

**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 |
| --- |
| As already stated I haven’t used arrays yet but I intend to. |
|  |

**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-Longwell-J/Fighter/blob/9dfa1f88d7951cfd0590066d97623cb7f2d14e36/Fighter/GameScene.swift#L84-L90>  Click on the screen to move around the fighter |
|  |

**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 |
| --- |
| Not yet, but I intend to add program solutions such as, having a game over screen when you run out of lives, and some powerups taking over other ones. |
|  |

**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 |
| --- |
| Not quite yet but they will be parts of code I’ve already talked about. |
|  |

**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-Longwell-J/Fighter/commit/1d38e9248f97dc4db185a5fe3bc0a16deab19b75#diff-ba1a2d1936d693d8da8f3bab94dd5ce6R77 |
|  |

**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-Longwell-J/Fighter/commit/1d38e9248f97dc4db185a5fe3bc0a16deab19b75#diff-ba1a2d1936d693d8da8f3bab94dd5ce6R74>  https://github.com/RSGC-Longwell-J/Fighter/commit/1d38e9248f97dc4db185a5fe3bc0a16deab19b75#diff-ba1a2d1936d693d8da8f3bab94dd5ce6R65 |
|  |

**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 |
| --- |
| I have done this many times throughout my program, but I’m not sure how to show it |
|  |

**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 |
| --- |
| <https://github.com/RSGC-Longwell-J/Fighter/commit/1d38e9248f97dc4db185a5fe3bc0a16deab19b75#diff-ba1a2d1936d693d8da8f3bab94dd5ce6R75>  <https://github.com/RSGC-Longwell-J/Fighter/commit/1d38e9248f97dc4db185a5fe3bc0a16deab19b75#diff-ba1a2d1936d693d8da8f3bab94dd5ce6R89>  I have good naming of my variables and functions, but I haven’t done any commenting yet, which I intend to do |
|  |

**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 |
| --- |
| Once again, I have completed this problems on multiple occasions, but don’t know how to show this |
|  |

**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 |
| --- |
| I haven’t done this yet, but I’m sure it will pop up eventually. |
|  |

**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 |
| --- |
| I don’t really have “test cases” per say, but I test the program to make sure that the code works the way I intend it to at the end. |
|  |

**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 |
| --- |
| I haven’t had to work around any major problems so far, but I will use these strategies when 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 |
| --- |
| I haven’t worked with anyone yet, but eventually, I will use a lot of peer feedback to better my ISP. |
|  |

**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?

**I haven’t a clue of what my level of achievement is since I don’t know where the rest of my classmates are at, and what I should be comping my work too. I do know that I have put a pretty large amount of my time into my ISP already, and I have made some serious headway into my final product.**

***Check-in Number 2***

**I’ve added a ton of new features since my last check-in, including adding a game over screen, power ups, a new background, and a lives counter. With all of these new additions to my program, I’ve had to either brush up on some previous coding skills, or learn new ones altogether. I’ve achieved much more over the shorter period of time between the last check-in then I did when we were first assigned the project. I believe this is due to my program finally evolving into something much more unique, and I feel that now that I have a starting base to be proud of, I can start to really make it flourish into a real achievement.**