# Project 1: War or Peace

Learning Goals

* Follow an interaction pattern
* Write readable code that adheres to Ruby convention
* Write tests
* Create an Object Oriented Solution to a problem (objects should interact to achieve the stated goal)
* Host code on Github and submit a Pull Request

Overview

In this project, you will write a program to simulate a game of [war](https://en.wikipedia.org/wiki/War_(card_game)). A user will be able to create two players, and watch them ‘play’ the game. In order to build good habits, we’ve broken the project into small classes to demonstrate objects that have a single responsibility.

Additional detail is included in the pages below.

* [Setup](https://backend.turing.edu/module1/projects/war_or_peace/setup)
* [Project Requirements](https://backend.turing.edu/module1/projects/war_or_peace/requirements)
* [Evaluation Rubric](https://backend.turing.edu/module1/projects/war_or_peace/rubric)

Setup

* Fork [this repository](https://github.com/turingschool-examples/war_or_peace)
* Clone YOUR fork to your computer
* Make sure you don’t clone the turingschool-examples repository
  + You can confirm that you have cloned the correct repository by running git remote -v. You should see something like this:

origin git@github.com:<YOUR\_USERNAME>/war\_or\_peace.git (fetch)

origin git@github.com:<YOUR\_USERNAME>/war\_or\_peace.git (push)

* Complete the iterations included in the project requirements
* Remember to commit frequently!

Requirements

In order to help you to organize your project, we have broken the requirements into four separate iterations. It is expected that you will complete iterations 1-3, while 4 includes extensions that represent opportunities to further explore.

* [Iteration 1](https://backend.turing.edu/module1/projects/war_or_peace/iteration1)
* [Iteration 2](https://backend.turing.edu/module1/projects/war_or_peace/iteration2)
* [Iteration 3](https://backend.turing.edu/module1/projects/war_or_peace/iteration3)
* [Iteration 4](https://backend.turing.edu/module1/projects/war_or_peace/iteration4)

In addition to the functionality outlined in these iterations, we will expect the you to do the following:

* write tests for each class
* write readable code
* make frequent commits
* use pull requests

For more detailed information about what we expect, please review the [rubric](https://backend.turing.edu/module1/projects/war_or_peace/rubric) before starting this project!

Rubric

Learning Goals

* Follow an interaction pattern
* Write readable code that adheres to Ruby convention
* Write tests
* Create an Object Oriented Solution to a problem (objects should interact to achieve the stated goal)
* Host code on Github

|  | **Exceptional** | **Meets Expectations** | **Below Expectations** | **Well Below Expectations** |
| --- | --- | --- | --- | --- |
| **Functionality** | Application is robust and can handle a variety of inputs including invalid inputs and completes iteration 4 | Application adheres to the flow outlined in the specification and can handle a variety of valid inputs (not just what is outlined in spec). Iteration 3 is complete. | Only up to Iteration 2 is complete or the application does not follow the flow outlined in the specification. | Iteration 2 is not complete. |
| **Ruby Mechanics** | Project includes appropriate uses of hashes and enumerables not covered in class | Appropriately uses Ruby’s built in datatypes and methods, flow control, and enumerables. | Does not appropriately use one or two of the following: Ruby’s built in datatypes and methods, flow control, and enumerables | Does not appropriately use Ruby’s built in datatypes and methods, flow control, and enumerables, or does not build classes |
| **Ruby Conventions** | Classes, methods, and variables are well named so that they clearly communicate their purpose. Code is all properly indented and syntax is consistent. | Code is mostly properly indented, spaced, and lines are not excessively long. Class, method, variable, and file names follow convention | Code demonstrates some proper indenting and spacing. Class, method, variable, and file names inconsistently follow convention | Code is not properly indented and spaced and lines are excessively long. Class, method, variable, and file names do not follow convention |
| **Testing** | All methods that do not require command line input/output are accurately tested. Best use assertions are made. | Each class has its own test file. Every method listed on the specification is tested. Most tests are written to accurately verify expected behavior. | Tests are written for most methods listed on the specification, but the tests may be in an incorrect location or the tests may not accurately verify the expected behavior | Fewer than 7 tests written (in addition to the card tests provided) |

**Version Control**

Code is required to be hosted on a GitHub repository, and will have at least 20 commits and one pull request.

Project Reflections

Throughout your time at Turing (Mods 1 - 4) you will be asked to keep a log of reflections for each of your projects. We suggest creating a gist that you can update at the end of each project so that all of your information exists in one place. This gist will come in handy as a reference when you start your interview prep and thinking about how you can leverage your project experience as great examples in your answers. Below is the template to use when writing out your reflection. Feel free to add additional questions/prompts that you want to reflect on to help you prepare for future interviews.

### **Summary**

In a few sentences, explain what the project was and the scope of your work. Imagine explaining your answer to a technical recruiter or a developer who you’d potentially be interviewing with.

### **Overall (Choose One)**

Solo

* What were your personal learning goals for this project?
* How did you work to achieve these goals?
* What was your process for starting the project?
* What worked well? If something didn’t work well, what was it and how would you approach it differently?

Pair/Group

* How do you think you did working with this group compared with the groups you’ve worked with previously?
* If you were to work with the same group again, what is one change the group could make to work together more effectively?
* What was the greatest challenge you had as a group?
* How would you describe your role in this team?
* How often did you interact with other team members?

### **Technical (Choose One)**

* What was a specific technical challenge your group faced?
* How did you approach this challenge?
* What resources or strategies did you use when you were presented with a technical challenge?

### **Reflection**

* What lessons did you learn during this project?
* How have you applied these lessons in a recent project or how do you plan to apply these lessons?

### **Group Project Reflections**

* What have you learned about the use of agile vs. waterfall in software projects?
* How did you and your group approach project management in this project? (What tools did you use, how did you hold each other accountable, etc.)
* In your team retro, how did you engage in the feedback process?
* What principles of feedback did you use in these conversations?
* How would you describe your ability to communicate feedback?
* How has this experience affected your communication skills?
* How do you want to improve in your ability to communicate feedback?