

# Assignment 1

**Due Date: Friday, January 22nd, ~~5:00 PM~~ 11:59 PM**

## Contents

- [Introduction](#)
- [Tasks for this Assignment](#)
- [Turning In](#)
- [Grading Rubric](#)
- [Tips and Suggestions](#)
- [Extra Credit](#)

## Introduction

For Assignment 1, your team is tasked with creating a proof of concept demo of a card game called Aces Up using ninja framework.

The rules for Aces Up are here: Note: Aces are high.

- Deal four cards in a row face up.
- If there are two or more cards of the same suit, discard all but the highest-ranked card of that suit.
- Repeat step 2 until there are no more pairs of cards with the same suit.
- Whenever there are any empty spaces, you may choose the top card of another pile to be put into the empty space. After you do this, go to Step 2.
- When there are no more cards to move or remove, deal out the next four cards from the deck face-up onto each pile.
- Repeat Step 2, using only the visible (top) cards on each of the four piles.
- When the last four cards have been dealt out and any moves made, the game is over. The fewer cards left in the tableau, the better. To win is to have only the four aces left.

When the game ends, the number of discarded cards is your score. The maximum score (and thus the score necessary to win) is 48, which means all cards have been discarded except for the four aces, thus the name of the game.

Rules from: [https://en.wikipedia.org/wiki/Aces\\_Up](https://en.wikipedia.org/wiki/Aces_Up)

You can also play online here: [http://justsolitaire.com/Aces\\_Up\\_Solitaire/](http://justsolitaire.com/Aces_Up_Solitaire/)

Your boss wants to know if it is possible to implement this game using the ninja framework. You will have two weeks to implement this game. Since this is just a proof of concept, we will not be developing an extensive UI at this time.

# Tasks for this Assignment

To complete this assignment, each group needs to do the following:

- **Fork [cs361-W16/https://github.com/cs361-W16/Assignment-1](https://github.com/cs361-W16/Assignment-1)** This will be the repo for your project. This repo will be the basis for your grade in this assignment.
- **Turn the assignment into User Stories.** From the Introduction Section, each team should turn the assignment into a series of user stories. We will talk about them in class, but a good introduction to User Stories can be found here: <http://www.subcide.com/articles/how-to-write-meaningful-user-stories/> Once each team has developed their user stories, they will create a page on their Github Project wiki called "User Stories", where they will record all their user stories.

Each User Story must have the following format

Card: ("As a [user], I want [function], so that [value]")

Conversation: (Details that clarify the user story)

Confirmation: (Tests to know when the User Story is completed)

Your user stories will be evaluated if they follow [INVEST](#)

- **Create Tasks from User Stories.** From the user stories, each team will create a series of features that need to be implemented in the code. These will be added to the Github issue tracking system. Each of these must be assigned to a specific team member, as well as have a time estimate given in the body of the issue.  
HINT: Good tasks are SMART. Read more about SMART goals [here](#).
- **Implement Code in Feature Branches.** Each Team member will create a branch for each feature they are going to implement. When the feature is completed, the team member will create a pull request to the group repo. They should use that pull request to close the specific feature. If at any time, a bug is found, it should be reported using the Github issue tracker.

## Turning In and Late Policy

To turn in this assignment, someone on your team should make an commit with the string "Assignment 1" in the commit message before, January 22nd, 5:00pm.

You will get your grade during the Grading meeting you will schedule with the TA or the Instructor the next Monday or Tuesday.

**LATE POLICY:**

Each team will have three late days to use for any milestone during the term. These late days will not be given automatically. If you intend to use your late day, you must email the instructor BEFORE the deadline.

## Grading Rubric

Here is how this assignment will be graded. Notice that there will be both individual and group scores

**Group Grading**

%	Item
10	INVEST User Stories in Github Wiki
10	Feature requests in Github Issue Tracker
5	Features developed in feature branches
10	Code follows MVC
5	Code is stateless
10	Model And Controller Code developed using TDD (>90% Test Coverage)
10	Tests for each User Story
<b>60%</b>	<b>Total:</b>

**Individual Grading**

%	Item	Team Member 1	Team Member 2
---	------	---------------	---------------

10	Implemented at least one User Story		
10	Code merged into Project from Pull Request		
10	Constructive Comments on another team member's Pull Request		
10	Attend Grading meeting		
40%	Total		

## Tips and Suggestions

Here are some suggestions that might make your life easier.

Manage your tasks with <https://waffle.io/>. You can use your Github username and it makes the task management a lot easier.

Use <https://slack.com/> for communication. If you use #Slack, join the OregonState group, and I will be happy to answer questions on Slack.

## Extra Credit

Your entire team will earn an extra 5% extra credit for this assignment for sharing a meal off campus somewhere. It can be a working meal or just a social meal.