# Sprint One: Planning and Prototyping

## Deadline One: Tuesday 11/8

* Using your understanding of the base project, create a prototype of your feature that integrates into the base project.

## Milestones:

### Milestone 1: Tu 10/25 -- Choose and understand the base version of Tetris

* Meet, discuss whose tetris file will be the base of the project
* Read through and understand the file.
* Think about how your features will fit into the base project.

### Milestone 2: Tu 11/1 – Plan your pieces and how they fit together

* Create an overview for the data flow of your features. Details will be refined later, for now, just document the logical steps of execution.
* Document and communicate how your features fits into the base project. If the base project needs to be refactored to facilitate a feature’s integration, communicate this and we will adjust the schedule.

### Deadline: Tu 11/8 – Prototype your features

* Using the logical flow you planned, create a bare bones prototype of your features that integrate into the base project as you planned.
  + Don’t worry about code smell or making it good yet, the prototypes just have to “work” in the loosest sense of the word.
  + Try to change the base code as little as possible. Try to make your prototypes as add-ons to the base code, not a direct code injection.

## Plan around:

### Homework 5 – Sat 10/29

# Sprint Two: Testing, Refactoring, Refining, Finalizing

## Deadline Two: Thursday 12/1

* Design a UML diagram for your features. Use this to refactor your prototype into a finalized version of the feature that satisfies your tests.

## Milestones:

### Milestone 1: Fr 11/11 – Create Initial Tests for your features

* Determine the main functionality of your tests. If you think you’ll refactor a specific section, create a test to ensure that the refactored code produces the same output.
* Comment the test’s purpose above the tests, and what possible errors can occur.
* Separate the tests from your module as much as possible, but don’t worry if you have to have an assert statement in the code.

### Milestone 2: Tu 11/15 – Object orient your planning

* Determine the Data Entities within your features.
* Use the Data Entities to create a UML diagram for your features.
* Use this UML Diagram to create and document a data flow.
* Record any Design Patterns you use in the implementation of your features. You’ll need at least 3.

### Milestone 3: We 11/25 – Object orient your programming

* Refactor your code to reflect your UML diagram/Process Flow.
* Focus on getting a working version of your features that reflects your planning, not on removing every piece of code smell.
* Document any code smell you’re able to remove by refactoring to an object-oriented version and the refactoring pattern you used to do so.
* Update/add any tests needed.

### Deadline: Tr 12/1 – Sniff and finalize your features

* Using your tests, refactor your features to remove as much code smell as possible.
* Update all variable names, method names, class names etc. to have descriptive names that describe their functionality.
* Finalize your tests, update comments where necessary.
* Update your UML diagram if any changes are made.

## Plan around:

### Homework 6 – Sat 11/19

### Midterm 2 – Tu 11/22 (No Class)

# Sprint Three: Sniffing, Documenting and Presenting

## Deadline: Saturday 12/3

* This is a hard deadline, it cannot be moved no matter how nicely we ask.
* All sections of the presentation, documentation and code smells will be submitted as they are at the deadline.
* Finalize your documents on Code smell, Refactoring and Design Patterns.
* Create your section of the Manual and Presentation.