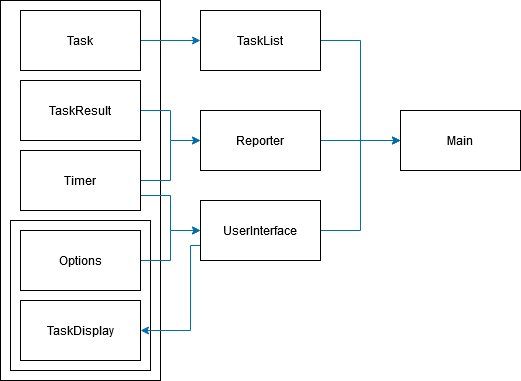
# Spatial Abilities Study Program Architecture

The purpose of this document is to provide an overview of the intended architecture of the program. It will not cover implementation details such as class and function definitions and will instead focus on the relationships between objects and their roles in the system.

First, a simple diagram will lay out the relations between the system components, which will then be explained in detail. Then at the end there will be a small diagram detailing the mapping between a task and the UI which may provide further clarification.

## Component Overview



### Main

This is the entry point of the program. It will initialize the User Interface, Task List, and Reporter components, then it will execute every task in the list until the list has been exhausted.

### Task List

The task list is just a list of tasks that will be performed. The only purpose of the task list is to provide the next task to be completed to main. When all the tasks in the list have been completed then the program is finished.

### Reporter

Reporters report task results to our system so we can store them. This will likely manifest as an http request to a back-end web server.

### User Interface

The User Interface component provides an abstraction for interacting with key components on the webpage. Examples would be setting the title and instructions for a task or getting the containers for a test’s images that are being compared.

### Task

A task is any step the user is going to go through during the test. This can be an instruction page, a web form, or a test itself. Tasks will provide Options, a Timer, a Task Display, and a Task Result to the other program components.

* Options and the Task Display are provided at the beginning of the task to initialize the UI
* The Task Results are returned when the task is completed.
* Timers are used throughout the task to notify the user of their remaining time and report the user’s task completion time.

The criteria for the completion of a task is determined by the task itself. Be sure to include event bindings in the task’s template (provided by the Task Display) which will ensure the task completes.

### Option

Options are simply the choices a user can make during any task. They are provided to Main from the current Task. When a task begins, its options are read and displayed on the interface from outside of the Task itself. A web form may only have one option which is a submit button, whereas a test may have many more options. Options provide their html template to the User Interface object to be displayed.

### Timer

A Timer is initialized when a task starts and is set to time from 0 to the task’s specified duration. The timer is used for metrics and displaying the timer bar at the bottom of the screen. There will likely only ever be two implementations of a timer:

* A limited duration timer which will force task completion when it’s done.
* An infinite duration timer which will track the completion time of a task but will not force completion at any point.

### Task Display

A Task Display is responsible for filling in the UI with a task-specific template. A task comparing images will utilize a Task Display that loads the two images onto the screen, whereas a task utilizing pie-charts will return an implementation that generates the charts and displays them.

Task displays are called at the beginning of a Task to display the necessary components for the user to complete the Task. The global User Interface object is passed to the Task Display so it can appropriately modify the interface. This relationship is inverse to the one between Options and the User Interface. Where the option provides the html template to a User Interface, the Task Display is given access to the entire User Interface abstraction. This is because the graphics for a Task is more complex than a simple button to choose an option, and it may modify multiple elements on the interface.

Currently, Options provide the template with which they display on the screen.

### Task Result

Task Results are returned from a task on completion and are passed to Reporters. These objects don’t have any functionality as they are only containers for test result data.

## Task-UI Correlation

The following diagram displays how the properties of interest in a Task bind to the User Interface, which may help understand how the components interact.

Note that the layer between the properties of the Task and the page itself is the User Interface component. No components except the User Interface component directly query the DOM, with an exception for elements that belong exclusively to a template produced by a Task Display.

