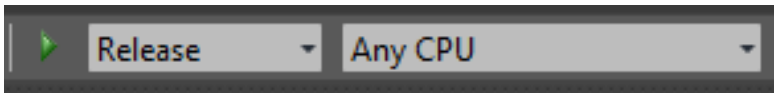
Siftables Emulator
Deployment and Usage
Singularity Software
April 27, 2012

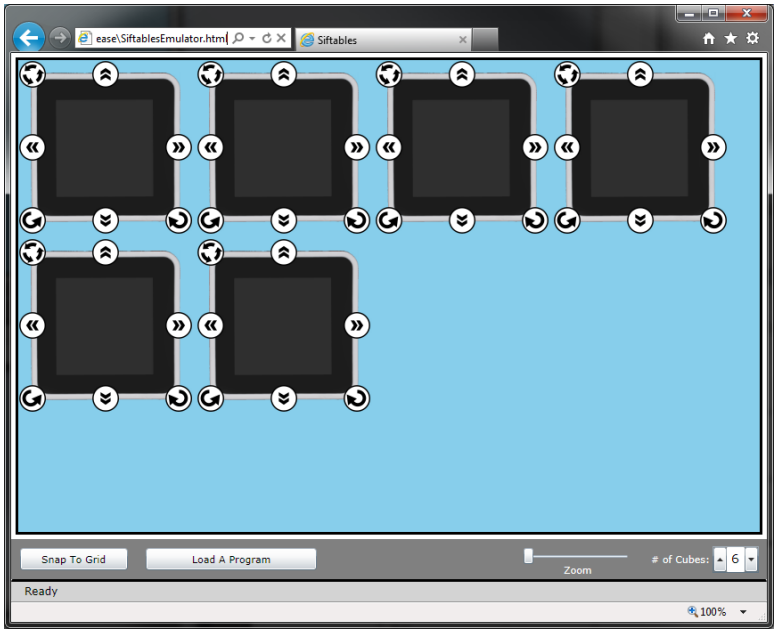
1 Install the emulator

Siftables Emulator is a cross-platform Silverlight application. To install it for the first time:

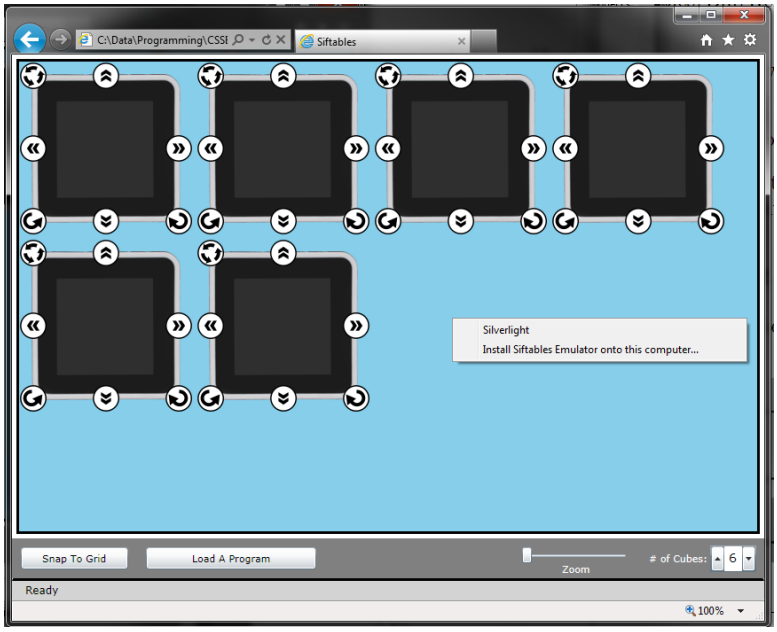
1. Build the project using the Release configuration in Visual Studio.



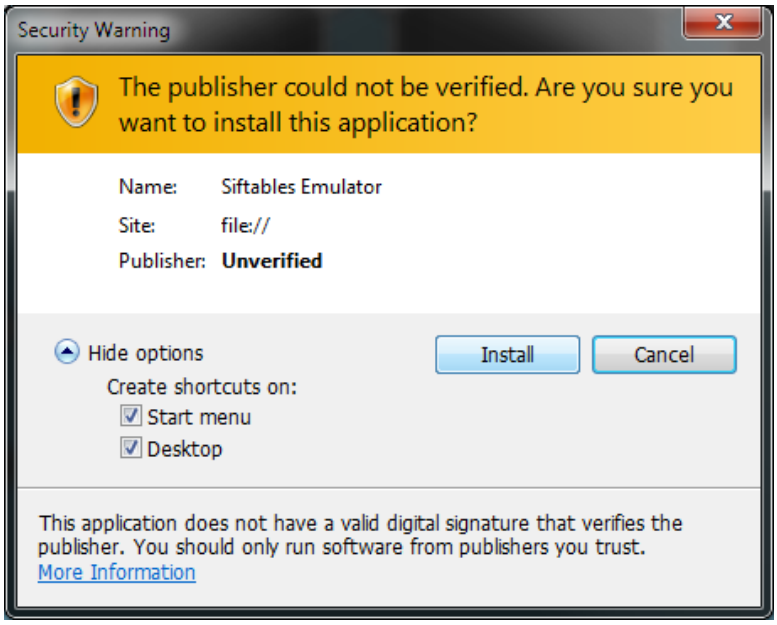
2. Open [project root]/Siftables/Bin/Release/SiftablesEmulator.html in a Silverlight-compatible web browser.



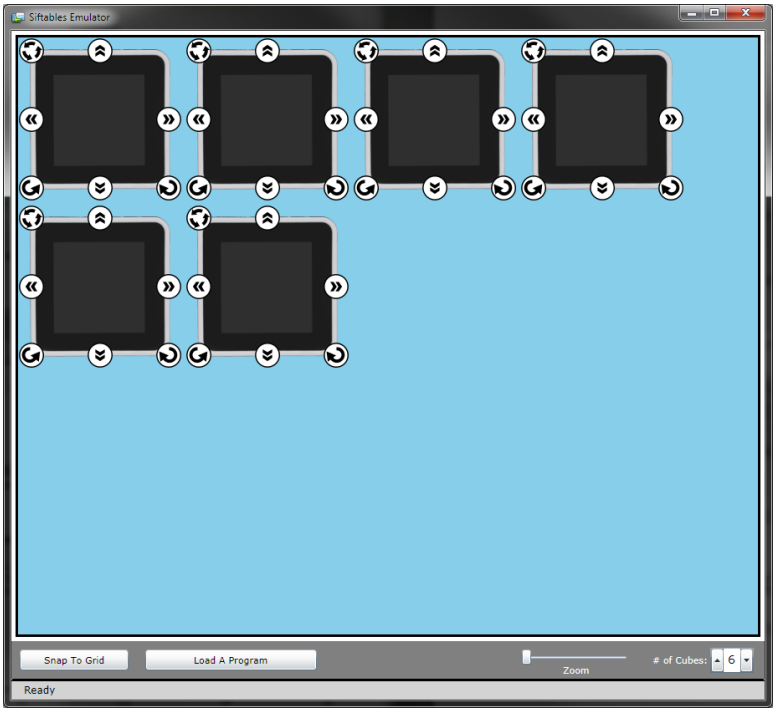
3. When the app loads, right click anywhere and choose "Install Siftables Emulator onto this computer..."



4. Follow the install wizard, choosing your preferred shortcut locations.

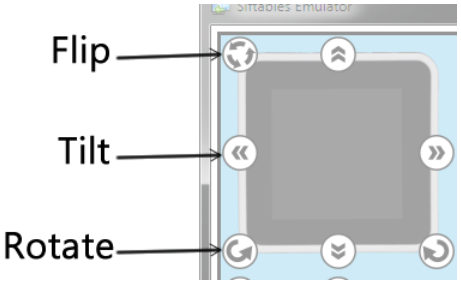


5. Siftables Emulator should launch automatically. If not, it can be launched from wherever you opted to install shortcuts in the previous step.



2 Walkthrough: Interact with the emulator

Once you've launched the emulator, play around! If you get stuck, consult the following mapping of actions on the physical Sifteo Cubes to their digital equivalents.



If you want to...	
Flip the cube,	click the flip button.
Tilt the cube,	click the tilt button for the direction to tilt [left, up, right, down].
Rotate the cube,	click the rotate button for the direction to rotate [counterclockwise, clockwise].
Shake the cube,	drag the cube horizontally back and forth rapidly.
Press the cube,	click the virtual screen.

If you click...	
Snap to Grid,	the emulator rearranges the cubes into a 4-cube wide grid based on order of cube creation.
Load a Program,	the emulator opens the Load dialog for running a Siftables application DLL.
Zoom,	the emulator zooms the canvas in (right) or out (left) to a maximum of 2x zoom.
# of Cubes,	the emulator changes the number of cubes available on the emulator "screen" to a [maximum of 9, minimum of 1] cubes.

3 Program for the emulator

3.1 Application Programming Interface

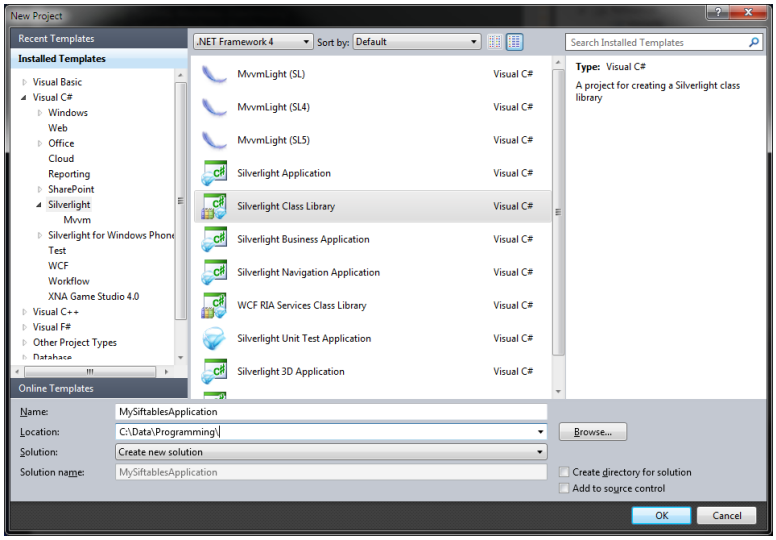
Siftables Emulator can be programmed using the official Sifteo API available at http://developer.sifteo.com/. The team believes that our implementation of the API outlined there is complete and is functionally on par with the native Sifteo.dll provided for use with the physical Sifteo cubes. This implementation is a combination of work done by the team specifically for Silverlight and the Siftables project and work done by the Sifteo team. The latter part comes in the form of SifteoExtensions.dll, a partial version of Sifteo.dll decompiled and retargeted to Silverlight with Sifteo's permission.

3.2 Target the emulator with an existing application

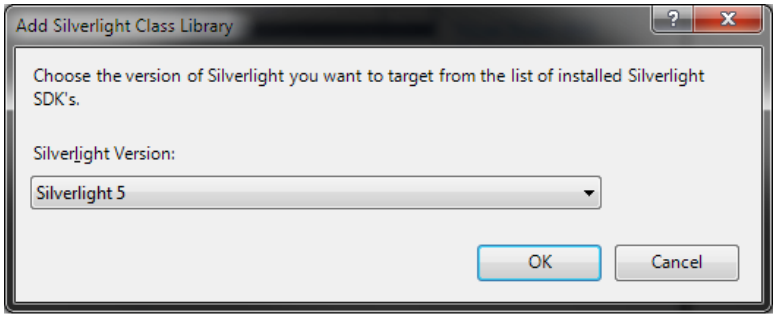
To target an existing application targeted for Sifteo Cubes to run in Siftables Emulator, follow the instructions in step 1 of the following walkthrough. When you've created the new project, add your existing application files to it.

3.3 Walkthrough: Prepare and run a new application

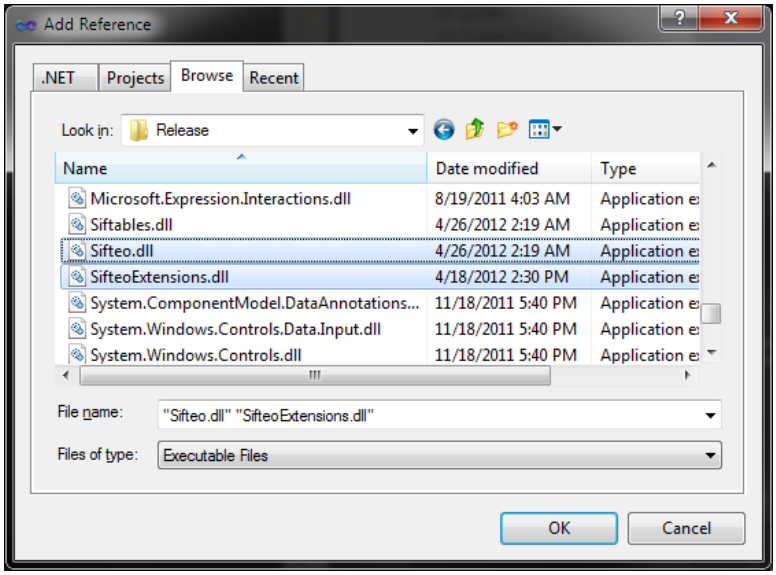
1. Create a new project for the application.
   (a) Select the Silverlight template group in the left pane, then select Silverlight Application in the center.
   (b) Give the application a name and a location. We recommend not creating a directory for the solution.



(c) Ensure that you use Silverlight version 5.



(d) Add Sifteo.dll and SifteoExtensions.dll to the project references. Note that those DLLs will only exist if you have already built the Release configuration of the emulator as specified in the installation instructions.



2. Build a blank runnable application.

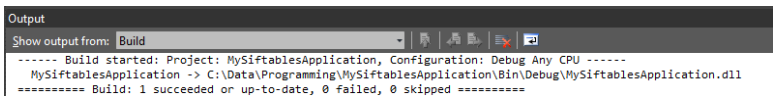
(a) Create a new class.

(b) Have it extend BaseApp.

```
13  namespace MySiftablesApplication
14  {
15      public class MySiftablesApplication : BaseApp
16      {
17
18      }
19  }
```

(c) Have it use the Sifteo namespace (Visual Studio should offer to do this for you).

(d) Build the solution. Note the location of the DLL in Output.



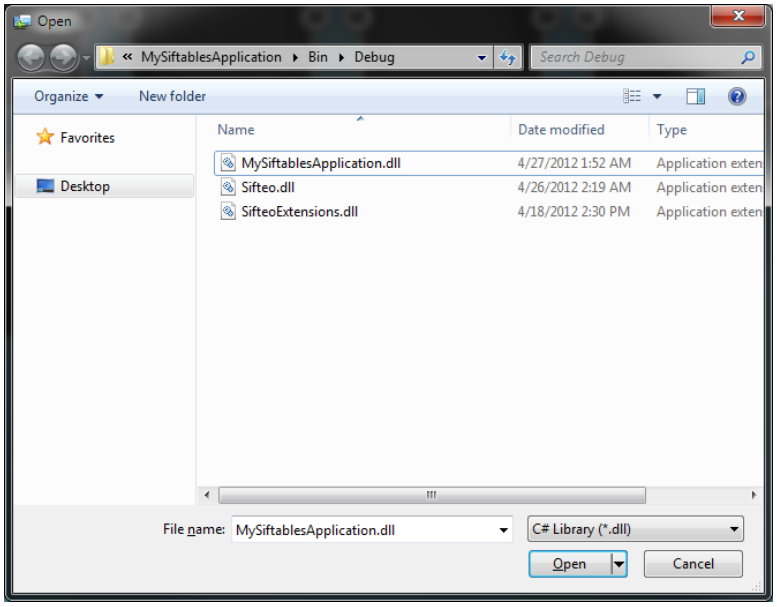
3. Run the blank application in the emulator.

(a) Launch the emulator.

(b) Click "Load A Program".

(c) Select the DLL built in the previous step.

(d) Click Open.



At this point, you have a fully runnable Siftables application. It doesn't do anything... but that part is up to you!

3.4 Example: Respond to cube events

All cube interactions appear to your program as events. If you want to do something when one of those interactions occurs, simply add a handler to the appropriate event.

If you want to respond when the user...	
Flips the cube,	bind to cube.NotifyCubeFlip.
Tilts the cube,	bind to cube.NotifyCubeTilt.
Rotates the cube,	you can't - at least, you can't directly. The orientation of the cube only affects operations that use neighboring to make multiple cubes interact.
Shakes the cube,	bind to cube.NotifyShakeStarted and/or cube.NotifyShakeStopped.
Presses the cube,	bind to cube.NotifyButtonPressed.
Has neighbored cubes,	access cube.Neighbors.

3.5 Example applications

Example applications are located in [project root]/Applications.

- CubeTestApp has examples of how to respond to most cube interactions and is a good place to start.
- ChangingColorsApp is an example of how to work with neighbored cubes.
- FractionOrderingApp uses neighboring and images in a simple ordering game.