To Do:

* ~~Refactor controller~~
* ~~Redo pan animator~~
* ~~Figure out what doesn't work in safari~~
* ~~Animation timing~~
* Standardize use of "loaded", "initialized", "configure", etc;
  + Use "constructed"

To Do Later:

* Refactor event dispatcher
* Modify menu css so it works with and without app modifications
* Make site cluster code work with sphere coordinates
* Figure out scaling bug – why 256.5 instead of 256?

Stuff I'm inconsistent on:

* Semi-colons after function expressions
* Single vs double quotation marks
* Capitalizing constants?
* Use of let / const

Notes:

* sharing data happens only thru controllers, not by passing parameters
* models and views should only broadcast events, not listen for them
* controllers can both listen and broadcast
* components interact with each other in controllers

Initial:

* Fade or show only when all images have loaded

Zoom In:

* Load new frame;
* Scale images up; no problem b/c all necessary images will already be in the dom (can animate this)
* when all images have completed loading, fade in new frame

Zoom (In and) To:

* The to-location should be somewhere on the screen; we just need half a viewport buffer and it should be fine

Zoom Out:

* Load new frame;
* Scale images down; no problem b/c all necessary images will already be in the dom (as long as the zoom out increment is 1);
* when all images have completed loading, fade in new frame

Zoom Home:

* this one is the most challenging due to the fact that there might be big negative z changes and big x/y changes.
* The strategy is to pre-load the frames, and then run the animation; it would be possible either to wait until everything has loaded (and use some busy animation) or else just be OK with possible white spots;

Panning:

* Translate the image container; calculate top, bottom, right, left buffer distance. If less than a certain threshold, then add more images to the container; make the translation contingent on there being enough room
* hope that buffer is large enough that the new area won't become visible before it loads;