We have a few questions for you that will help us continue with the design of the game:

* What will the format of the game look like?
  + Will there be any rounds? If so, how many rounds?
    - Yes, the default number of rounds should be 25 but I’d like the number of rounds to be an experimental factor that I can change.
  + How much time per round?
    - 30 seconds.
  + How will the user receive feedback on when they have found the maximum score?
    - They won’t! At least in human-oriented search problems, we don’t ever know if we’re at a maximum and one of the questions is how we behave around the max if we don’t know it’s the max.
* Are the shapes in the PowerPoint the ones that you want in the final game? Or do you have other files in mind?
  + In spite of using common shapes in the mockup. I don’t actually want common shapes because participates might have association with, for example, the shapes in the demo you’ve done that might bias their willingness to choose them. I’ll get a collection of distinct-but-semantically-uninteresting SVG paths to you soon. For now, any 10 paths should be fine.
* How do you want the Art Display to look? Should it be a mirror of the Selector display or some random scattering of shapes?
  + Randomly placed within an element like 60% the size of and centered above the selector element.
* What data do you want to collect from each player? Presumably, a timestamped (both universal time and within-round timer) log of each shape combination submitted?
  + Yes, we need timestamped submissions. Universal time would work, but we’d be converting it to time since the start of the game so getting that directly would be even better. If possible, I’d like to log if the window ever becomes out of focus. I think a timestamped events log should be pretty straightforward, but please let me know if not.
* Do you have the underlying model for determining the points for each combination yet?
  + In a specific sense, yes, because we’re starting with some validation on the canonical model. But in the bigger picture, what we need is a way to load a static “fitness” file on the server so that we can send the submission as an array (e.g. [1,0,0,1,1,1,0,1,0,1] where a 1 indicates the ith element was included in the “art”) to the server and get back the corresponding fitness (e.g. 875.34). The relationship between the fitness and the Hamming distances between nodes in the primary variable of interest so I’ll be stochastically generating the landscapes and having multiple participants search each one. For that reason, we shouldn’t be coding the landscape generating into the platform and just loading the mapping between locations (i.e. “art” submissions) and fitnesses.

Other things I’ve thought of:

* An important design element is that the selected shapes should remain selected *after* submission. That way the participant is still at that location as they continue the search.
* In order to replicate existing results, the search will start at the lowest fitness value on the landscape. Therefore, the platform needs to be initialized with a specific location already selected. I’ll want switch to randomizing it after replication.
* Point values for the selections will be scaled between 0 and 1000. However, that won’t be made clear to the participants.