### Preproduction Stage A Deliverables: Garden of Math and Evil

#### Is the user’s input platform-specific?

* No → both web and mobile can tap objects onto a grid and click Submit button. The best interaction is to activate an available item, and allow clicking on an open space to plant that item. This is slightly less intuitive than dragging to plant at first, but avoids the fatigue that quickly sets in when dragging and dropping many items throughout the session.
* Keep screen size for mobile in mind for level layouts

Is the user’s input scalable/skinnable at all difficulty levels?

* This is a non-issue, as the interaction will be the same at low and high levels
* We have discussed 12 items to be the maximum onscreen at any one point so Art knows how to size items in the playable screenspace

Are the game’s feedback loops complete/comprehensive, concise and clear?

* What happens when users get answers wrong?
  + Visual Feedback - showing users a “✓” or “✗” is consistent with our other games as a clear measure of success
  + 2nd Attempt - Mimic Organic Order and other games (Fuse Clues, Continuum). Users are given a 2nd chance:
    - Scoring: reduced score for getting it right 2nd chance
    - Levelling: stay where they are if correct, go down by 1 if wrong still
  + Hints:
    - ✓ (TEST 2/14) Show an icon that it is available on all trials where it applies
    - ✓ (TEST 2/14) Show it before the first level they would play where it applies. Prepared player for new challenge, signified improvement, smooth transition into new material.
    - ✓ Hint icon “jumps” when they answer incorrectly to let them know an hint is available
    - ✗ ~~Show it after they get the answer wrong. Too aggressive.~~
    - ✗ ~~Show it at the beginning of the session.~~
  + There will be no Review section at the end of the session. Unlike other games where words can be reviewed and learned, this game is not about the particular trials but the application of core concepts, so Review would not be valuable enough to include.
  + Showing the Correct Answer after 2 Incorrect Submissions
    - Show the simplest correct answer that incorporates the appropriate level of complexity for the level
      * Scaffolded algorithm w/ weighted
    - ~~Incorporate the user’s answer? No, too many edge cases for the effort~~

Are the feedback mechanisms in those loops intuitive? Do they convey all info/imagery needed to promote an understanding and ease of use (minimizing text)?

* \*\* see above

Is the game’s content dynamically created or based on a repeatable mechanic, as opposed to having fixed content or levels?

* (Similar to Star Search) Level components are listed in [this spreadsheet](https://docs.google.com/a/lumoslabs.com/spreadsheets/d/1wNPYZ6zog5rNU__sJLdqbvDwRDdT5Adx7JEa71BzLbE/edit?usp=sharing), but variety will be added by changing up the:
  + Items / attributes included in the level
  + Items / attributes in the prompt
  + Different coefficients as scalars
  + Different ratios as stated ratio goals
  + Alternative TWICE / THEN prompts *at higher levels*

How will the game’s difficulty and/or mechanics adapt as players succeed and progress?

* \*\* see above spreadsheet

What parameters will be exposed to configure the game’s difficulty per-level?

* \*\* see above

How will we ensure there will be enough content to satisfy players after hundreds or thousands of gameplays?

* Affirmative capability to produce very difficult and very easy levels
* Feel confident in the evidence that changing values, items, and attributes in prompt produce acceptably different trials as perceived by the user

How will the game measure and score a player’s performance in a positively reinforcing way (w/o any possibility to cheat or avoid training and still get good scores)?

* The game will mirror other games by scoring trials based on level and any 2nd tries/hints used
* Levelling may incorporate some momentum element that will help put users in the correct difficulty category faster and accommodate not having TIME as a variable, which we are doing away with because of how negatively people reacted to it in playtests.

If scoring is dependent on timing, does this conflict with the way we expect the target audience to engage with the game? If so what other solutions are there.

* \*\* see above

How granularly can this game be scored? Will most users always have an opportunity to continue improving little-by-little?

* Rankings of dials to tune difficulty (steve hypothesis):
  + ✓(major) Math concept complexity (or linear combination)
  + ✓(minor) Complexity of available items (overlapping attributes/distractors)
  + ✓(very minor) Complexity of locked items
  + ~~✗ Complexity of ratio~~
    - Q: How difficult to users find it to identify and ignore distractors?
      * Distractors make it only very slightly more difficult (excluding relative probability problems)
    - Q: How difficult does it make a trial to lock relevant items?
      * Locked relevant items make the problem much more difficult, and seem to scale with the difficulty of the problem

Does this design have low variance/luck factors in scoring?

* There is low luck involved in getting the correct answer, and it gets harder to be lucky at higher levels. Variance could be different depending on if they understand/don’t understand certain probability techniques and get those techniques more/less.

How and when will new concepts be introduced?

* Tooltips before new material indicate to users that they are advancing, are coming upon new material, and give them an anchor to an example where they can see the tool being used

Are there any additional Science team concerns/requirements to keep in mind as we move forward with full production of the game?

* What does a session look like
  + Continue to playtest number of trials, but plan to mimic Organic Order (6-8 trials)

How will we architect the game’s source code and files to allow for full production to run efficiently, allowing for possible future changes/iteration, and making it easy for cross-game consistency changes to be integrated painlessly?

* Unity shared/components still under discussion but see no problem particularly since prototype is in Unity
* All ratio and correctness calculations are currently completely line portable
* View interactions are very simple

Has QA reviewed the game design and provided a list of required debug functionality?

From @DavidBeavers (2/24/2017)

* Show level
* Show any hidden levelling/scoring that’s happening
* Show FPS (if able)
* Allow increase level
* Allow decrease difficulty
* Allow short game (2 trials)
* Allow always correct