Prepare for testing!

* Only include components that emerged from testing commentary
* Solve problems
* “fix things you’ve put in already

This means every critique is based around testing 🡪 what did the users say? What are your testing results?

There will be “push back” 🡪 why did you change this?

Take testing results and refine into an issues list (e.g. GitHub Issues)

Testing 🡪 Findings 🡪 Key Issues 🡪 Volunteer to fix issues

Looking at scores for each question in a spreadsheet 🡪 findings made through charts etc

Test approximately 5 **naïve** people **minimum** per week (6 for us = 2 full testing sessions) 🡪 Ask some back as “deep testers”

Report each week becomes “we discovered this, here’s what we’re going to do to fix it”.

Get participants 🡪 make appointments 🡪 manage results 🡪 process issues

Two team members

* Watch player reactions
  + Video recording?
* Watch game play
  + Screen capture

Follow a script to make reproducible results 🡪 can make a video recording of it OR be very dead-pan serious while reading the script.

Keep video/audio files for reference (useful for problem understanding)

Both team members make COPIOUS notes!

* Record as a “stream of consciousness” and edit later
* Note EVERYTHING

Testing Processes Testing Session Outputs

* Boil down findings from testing log into a prioritied issues list
* Track issues 🡪 deal with game breakers first (GitHub)
* You will NOT be able to fix all bugs 🡪 choose wisely
* First report is next week 🡪 get testing started now!

Weeks to come

* Week 5 🡪 Crawl 🡪 plans, who, what, where, when, how you will test
* Week 6 🡪 Walk 🡪 simulated naïve playtest by Ross (David Bromley)
* Week 7 🡪 Run 🡪 actual naïve tester observed by Matt and Ross
* Show results of playtesting: findings, surveys, what you did in response

Prioritise naïve playtesters over deep, fix tester feedback, report to Ross. (Weekly Report should be modified for tester feedback)