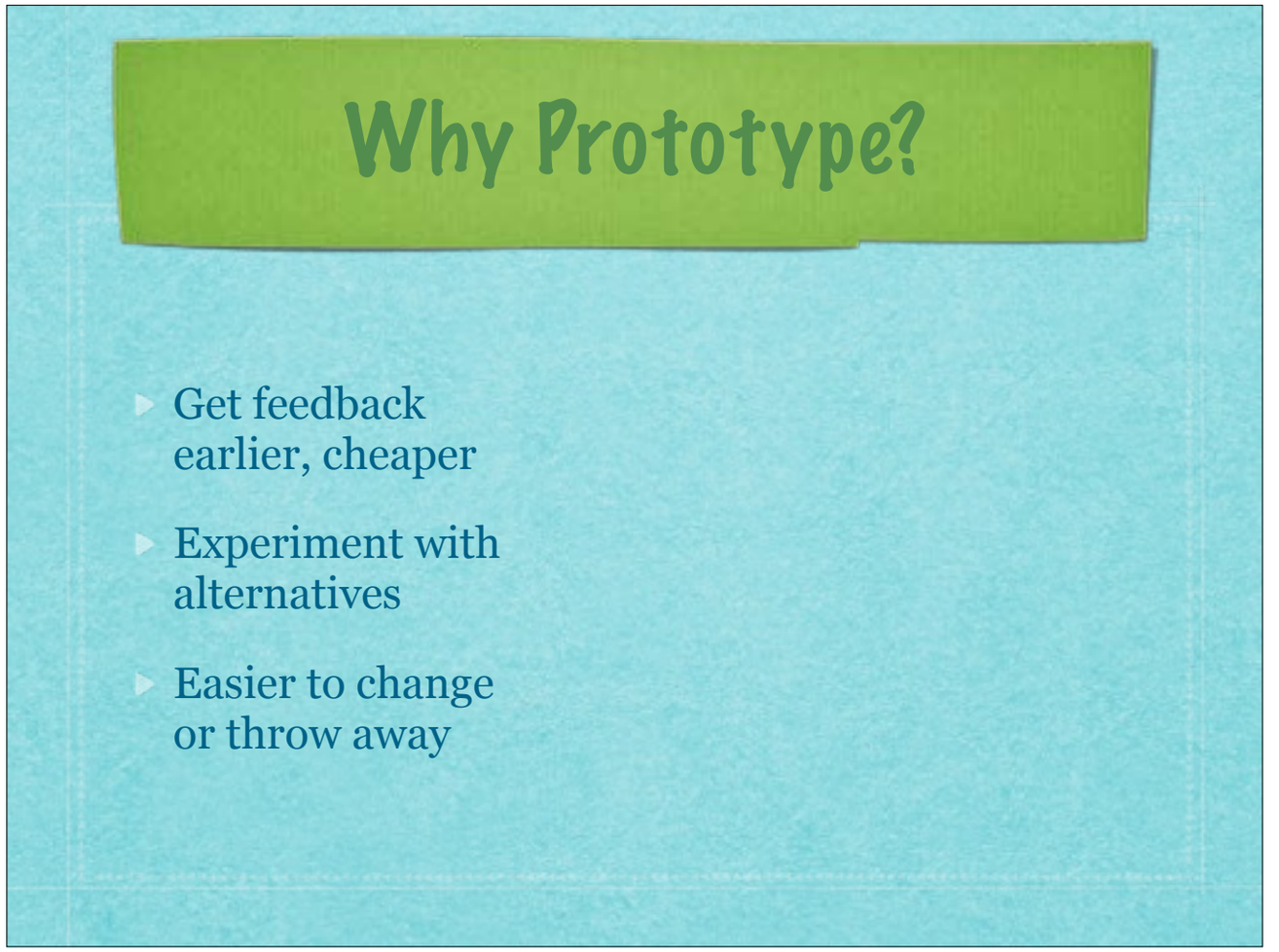
Paper Prototyping

MIT Game Lab

~~Phillip~~ Philip Tan



Prototypes are much **faster to build** than finished implementations, so we can evaluate them sooner and get early feedback about the good and bad points of a design.

If we have a design decision that is hard to resolve, we can **build multiple prototypes embodying the different alternatives** of the decision.

A prototype can be **changed more easily**. Paper is easy to change. You can even change it in the middle of a test. If part of the prototype was a problem for one user, you can scratch it out or replace it before the next user arrives. Paper is more malleable than digital bits in many ways.

If the design flaws are serious, a prototype can be **thrown away.** It's important not to commit strongly to design ideas in the early stages of design. Unfortunately, writing and debugging a lot of code creates a psychological sense of commitment which is hard to break. You don't want to throw away something you've worked hard on, so you're tempted to keep some of the code around, even if it really should be scrapped.

In fact, this exercise should end up with a prototype that is not immediately useful for your digital project!

This is a good mindset to have in early iterations, since it maximizes your creative freedom.

“A series of interesting choices” - Sid Meier

Prototyping lies at the heart of good game design. All you have are the fundamental mechanics to keep you engaged, and **if these mechanics can sustain the interest** of testers, then you know that you’re onto something.

The main advantage of prototyping is that it forces you to **define game mechanics in their purest form**.

Focus on the small handful of choices that the player must make to play the prototype.
Model a system with a few basic rules that creates interesting choices for the player.
Always ask a question about your game and make sure it’s falsifiable.

Physical prototyping **does not give insight to everything**, e.g. the sensory experience created by a digital game, the feasibility of implementing a digital feature.

Imagine getting in a room with developers who know nothing about the project and you are trying to describe to them the game you have in your head.

If you want to create gameplay that people have never seen before, it may be impossible. A written treatment or design document is good, but when you need to communicate how a complex system works, it can’t compare to a prototype that someone can actually play. A physical prototype that they can sit down and play helps them grasp your vision of the game. They have **something solid to work from**.

Any game design benefits tremendously in its early stages by building a physical prototype. Physically prototyping allows you to build a structure for the game, think through how the various elements interact, and formulate a systemic approach to how the game will function.

At a minimum, physical prototyping forces you to **think through the design elements and define them**. You can always change them down the road, but this gives you **a framework to build upon**, and that in itself can save you from stumbling around blindly when it comes to preparing and launching a production team.

Bring in players!

No special skills are required for people to play paper prototypes. Artists, designers, producers, coders, QA leads, audio leads, and even players can help create prototypes and operate them.

Unlike a sketch or storyboard, a paper prototype is **interactive.** You can actually user-test it: give users a task to do and watch how they do it.

If you do *several* prototypes and present them to the same player, people tend to be **more ready to criticize** and offer problems, which is exactly what you want in the early stages of design.

Paper prototypes can **reveal many usability problems** that are important to find in early stages of design. Fixing these problems may require large changes in design. If users don't understand the metaphor or conceptual model of the interface, for example, the entire interface may need to be scrapped.

In some cases, **suggestions from players can also be implemented directly into the prototype**, allowing testers to be involved in the design process. This can be very useful in co-design.

Designers should be cautious and vet each suggestion to see if it fits their vision. **Testing is not a replacement for direction.**

Useful tools

- Big sheets of paper
- Index cards
- Dice
- Post-it® glue and notepads
- Pencils, pens, markers, scissors, tape
- Game bits, small toys, kindergarten counters
- Photocopier, mobile phone camera

Index cards can be used as cards (shuffling, hidden information, recording options) or for rules

Dice can be tokens, help keep track of stats. Don’t use two D10s for large numbers! Use a notepad instead.

I find kindergarten counters more useful than game bits. There’s little assumption about what they’re supposed to be, no worries about returning them back to the games they came from.

Don’t get too enamored with the interlocking cubes (or any other particular prototyping tool).

Don’t forget to keep a record of your drafts and setups. Use your mobile phone camera during the session, a photocopier or scanner for your project.

Team roles

- Players
- “Computer” or facilitator
- Observers

The **computer** is the person responsible for making the prototype work. This person moves around the pieces, writes down responses, and generally does everything that a real computer would do. In particular, the computer should *not* do anything that a real computer wouldn’t. Think mechanically, and respond mechanically.

The **facilitator** is the human voice of the design team and the director of the testing session. The facilitator explains the purpose and process of the user study, obtains the user’s informed consent, and presents the user study tasks one by one. While the user is working on a task, the facilitator tries to elicit verbal feedback from the user, particularly encouraging the user to “think aloud” by asking probing (but not leading) questions. The facilitator is responsible for keeping everybody disciplined and the user test on the right track.

Everybody else in the room (aside from the user) is an **observer.** The most important rule about being an observer is to keep your mouth shut and watch. Don’t offer help to the user, even if they’re missing something obvious. Bite your tongue, sit on your hands, and just watch. The observers are the primary note takers, since the computer and the facilitator are usually too busy with their duties. Have a notepad on hand!

Wizard of Oz

- Someone plays the "computer"
- Very constrained
  - Communication
  - Rules
- Don't let the player know what the "computer" is thinking

Depth refers to how deeply each feature is actually implemented. Is there a backend behind the prototype that's actually implementing the feature? Low-fidelity in depth may mean limited choices (e.g., you can't print double-sided), canned responses (always prints the same text, not what you actually typed), or lack of robustness and error handling (crashes if the printer is offline).

Can be **high-fidelity in depth** at little cost, since a human being is simulating the backend.

Player vs Player

- People play together
  - Cooperative/Competitive
  - Symmetric/Asymmetric
- Very loose
  - Open communication
  - Rule negotiation
- Facilitator helps to explain rules

May still be useful for single-player games

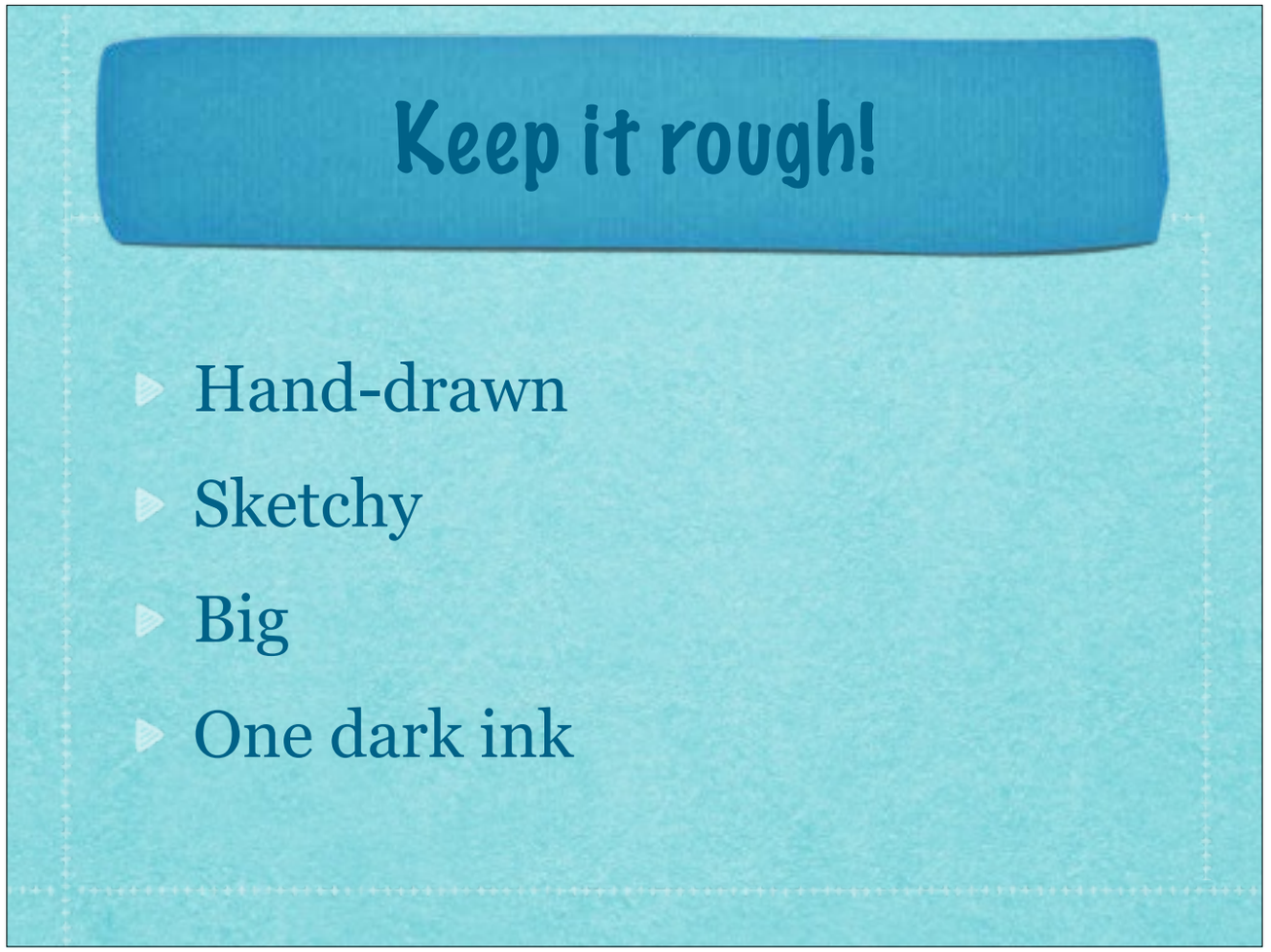
“NPC”s can have different rules from “PC”s

Hard to get consistent results across tests

Still absolutely necessary for multiplayer games

Live Action

- Not for today's exercise!
- Predetermined rules
  - Explained before game
  - Loose interpretation and constraints
- Limited communication
- Try it for "avatar walks around" games



Hand drawing is fast. You can draw many sketches in the same time it would take to draw one nice picture, let alone implement a game with code.

Some of you can draw very nicely. Try to make it look rough and sketchy.

Sketchy prototypes focus attention on the issues that matter in early design without distracting anybody with details. You aren't bothered with details like font, color, alignment, whitespace, etc.

Rough prototypes improve the feedback you get from users. They're less likely to nitpick about details that aren't relevant at this stage. They won't complain about the color scheme if there isn't one.

A hand-sketched design also seems less finished, less set in stone, and more open to suggestions and improvements.

Big is good. A paper prototype should be larger than life-size. Fingers are bigger than a mouse pointer.

Everyone involved needs to see what's going on during testing.

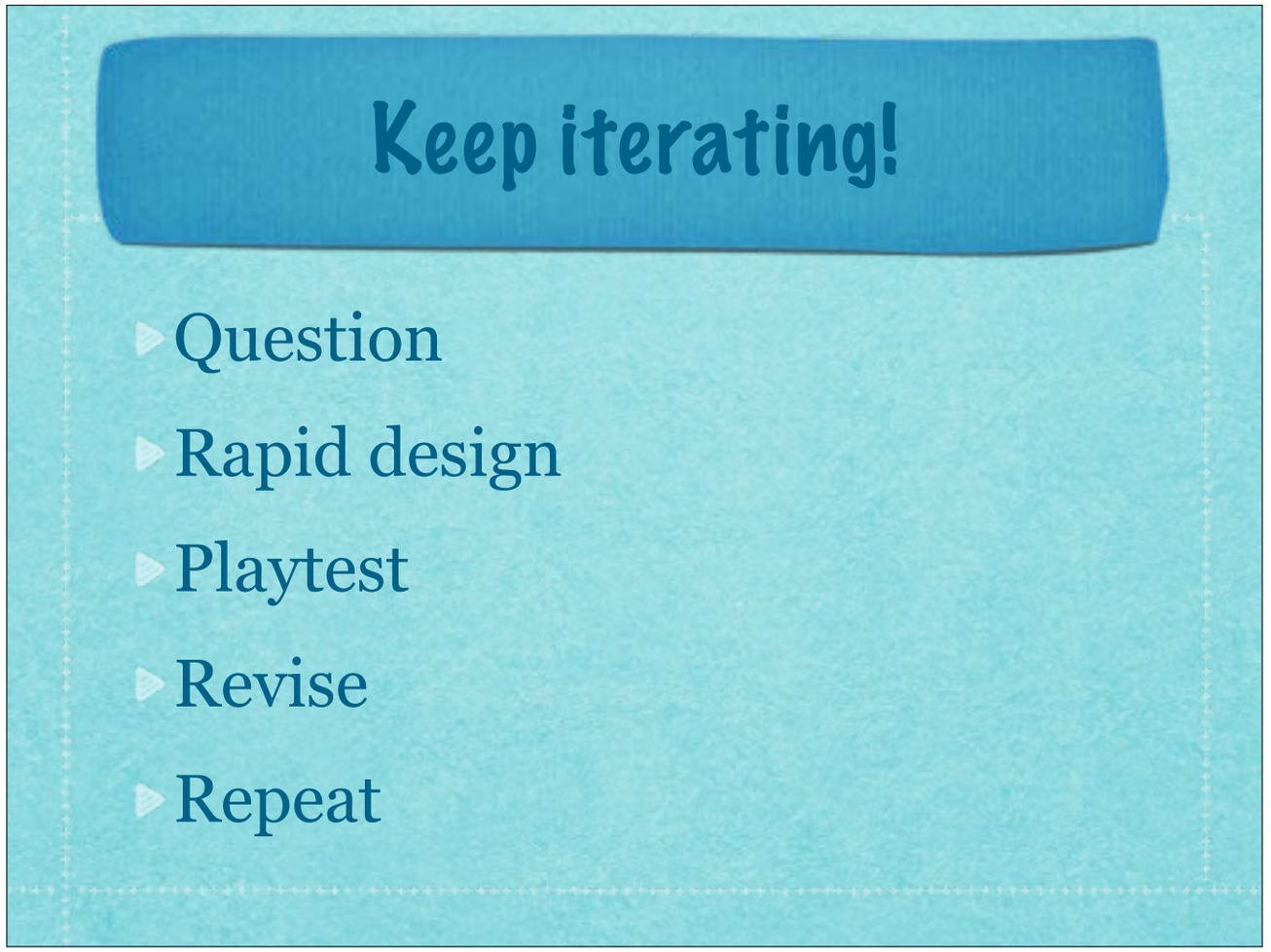
One dark ink. Don't worry too much about color in your prototype. Use a single, dark color for text.

You can use color to differentiate between different types (cards, bits) but anything you draw should be monochrome.

Game paper prototypes should only represent a subset of your feature set.

While it’s easy to cram more features into a paper prototype (just sketch/write more) it is usually detrimental to iteration, since it makes it harder to identify which feature caused problems.

Also, feature creep makes playtesting longer because people have more to read or understand before they can play.



Question: Start with a problem to be solved and an idea to be tested. Make sure it's falsifiable - what does a workable solution need to achieve? You can also look at Axiomatic design: establish Axioms - things that you are taking to be true or baseline that need to be fulfilled by any workable design, then think of solutions.

Rapid design: The key point of prototyping is that it has to be rapid. The more iterations a game can undergo, the higher quality it will be, so spending time discussing issues can be a waste of time. There is a great tendency among designers to debate the outcome of something that could be tested in half the time.

Playtest: A full or partial play session to identify strengths and weaknesses in the design.

Revise: Fortify weaknesses or build upon strengths. Sometimes, changes are small incremental changes incorporated one at a time. Keep and eye out for unexpected play dynamics that cause the game to go in a completely different direction, those might be gold.

Repeat: Play again! Once things seem to be working, grab someone from another prototyping group to playtest.

Keep changing!

- Amend, replace, kill
- Limit/unlimit
- Player interference
- Mess with the play order
- Multiply/divide by 2
- Core and reconstruct
- Throw away

Kill a Rule

Killing a rule is usually a good thing to do in the prototype stage. Identify the core of the game and start by killing every rule that doesn't directly affect the core of the game.

Make a Resource Limited (Or Unlimited)

Interacting with Your Friends (multiplayer, or single-player vs NPCs)

While it's always fun to win a game, it's occasionally as fun to stop someone else from doing the same or otherwise affect their play.

Look at the various actions your players are performing in a game. Next, ask yourself, "How could someone else stop that from happening or make it happen even faster?" You can even take it one step further and ask how players could then protect themselves from aggressive moves or solicit help more directly from their opponents.

Allowing players to affect the play of other players is an effective means of introducing a bit of uncertainty even in the most mundane games. It also forces a minor amount of strategy on players who must consider preparing a counter to that attack or a means to ally with another player.

Mess with the Play Order

We've talked about rearranging the order of the rules. Consider allowing players to mess with them too: Mechanics such as "go again" and "skip a turn" in board games are also common.

Use the "Rule of Two"

If something seems off but you're not sure what, take one of the game's values and either multiply or divide it by two. Making such a drastic change gives you insight into how the game's values interact with each other and what effects they have on play, and you'll often realize things you never would have seen if you made smaller, incremental changes.

Core and reconstruct

The prototype you've created may or may not be very playable. Parts may be out of balance and rules may conflict. Your game may also feel slow or disjointed.

Many designers panic at this point and throw out everything. They feel that their game is hopeless and the only solution is to start from scratch with a new game idea concept.

Before you take such drastic measures, go back to your just your core game mechanics.

CMS.611J / 6.073 Creating Video Games

Fall 2014