

PROTOTYPING PERSONAL EXPERIENCE

Jenny Jiao Hsia

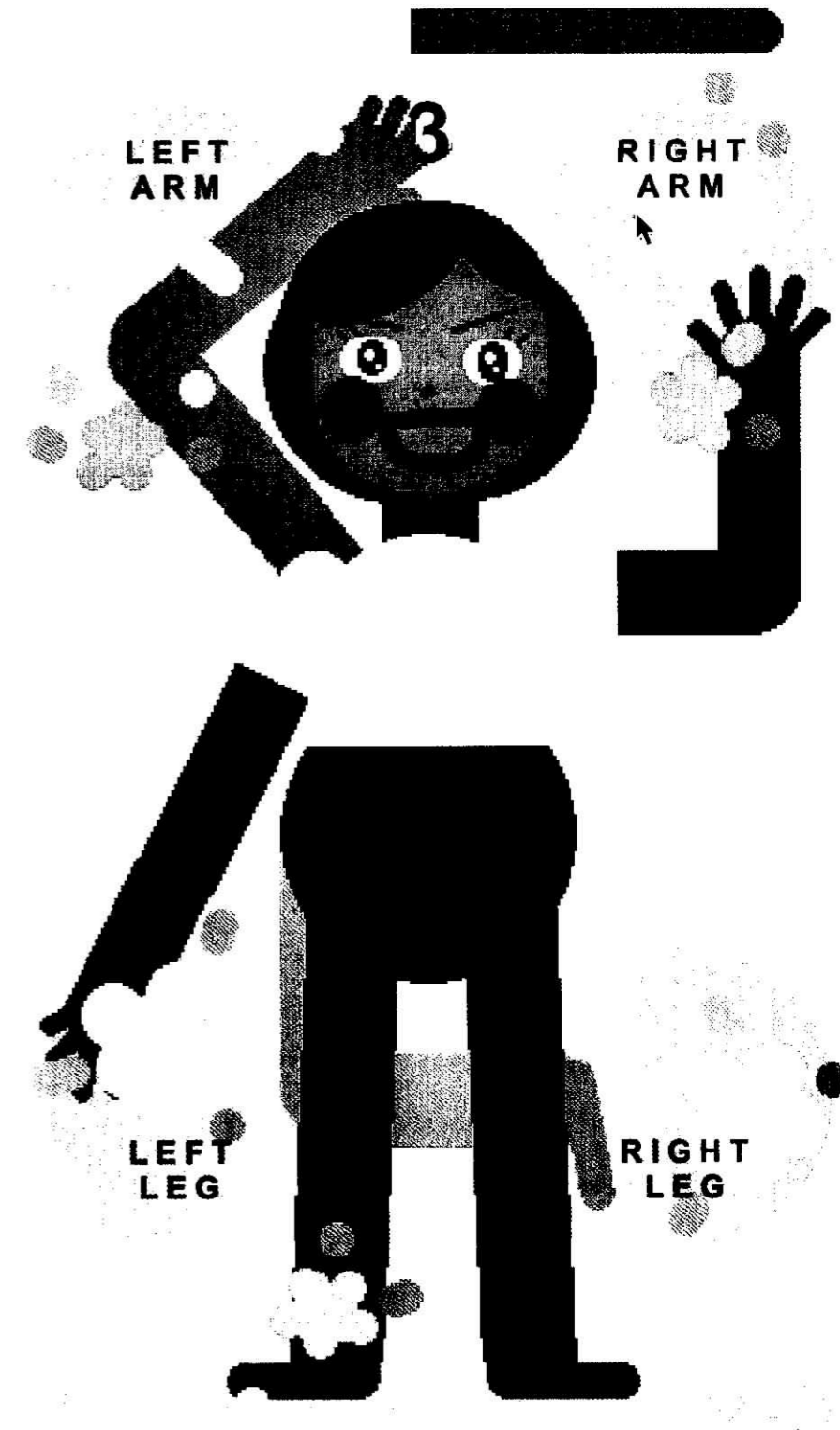
Growing up in a Chinese–American household in New York City, I would often watch my dad and younger brother playing computer games. I had no interest in joining them and believed that videogames were just a huge waste of time. In college, I wanted to become a doctor, but when my plans didn't work out, I started learning how to program. I had always loved to draw so it made sense to combine the two interests. I enrolled in a game course on a whim and ended up making my first game at a game jam that semester. It was a small game about a llama collecting balls of yarn and using them to climb different platforms, but it eventually led to a BFA in game design from the NYU Game Center.

I've continued making games ever since. In 2014, I collaborated with Jane Friedhoff on *Slam City Oracles*, created for NYU Game Center's annual *No Quarter* exhibition.¹ I also worked with AP Thomson during my time at the NYU Game Center and together we created *Stellar Smooch*

(2014) and *Beglitched* (2016). I've made several small games that deal with personal aspects of my life on my own. Some of these are about everyday routines, such as slapping on makeup in 10 seconds or less (*Morning Makeup Madness*, 2016) and putting a noodle-limbed avatar through a series of challenging yoga poses (*Wobble Yoga*, 2016). My other work has dealt with darker topics such as blacking out on New Year's Eve (*and i made sure to hold your head sideways*, 2017) and having a Skype conversation where it's apparent that a long-distance relationship is not really working out (*Chat With Me*, 2016).

Most of my games share a playful and silly aesthetic. Some of them attempt to approach serious subjects with a sense of humour. I think videogames are a really cool medium; but there is still a lot to explore with them. They possess a degree of interactivity, which other media, including literature and film, do not have. Unfortunately, a lot of videogame culture can also be quite toxic and hostile to folks who do not identify with the stories and characters that the majority of these games feature. I believe it's time for videogames to incorporate new voices and perspectives.

The game I am currently working on, *Consume Me* (TBA), is a darkly humorous personal game about



85 *Consume Me*, game development screenshot

TOTAL
CALORIES
0



86 *Consume Me*, game development screenshot

my relationship with food. These prototypes are part of a larger experience that draws on my past with disordered eating. By making you feel obsessed with calories, forcing you to look at food strangely and mechanically, and fret over losing weight, *Consume Me* puts the player directly into the mind of the dieter. These prototypes explore a three-way dynamic between the player, the character in the game and the fact that this character is based on me, the author. 'Fit', 'slap' and 'control' are just some of the ways the player experiences this dieting mindset, as expressed through mechanics. What does it mean to push and prod the character into certain eating behaviours when the player doesn't get full control of the character's thoughts and internal state? The player is put in the awkward position of performing as the character, but only in a limited sense.

These prototypes are intended to be intimate and confessional, and present a goal-oriented relationship with food using simple, but distressing mechanics. Cramming *Tetris*-shaped pieces of food on a plate to hit a calorie target, putting a flopping avatar through a fat-burning workout, and showing the protagonist's distress as she tries on a crop top, place powerful feelings of self-consciousness and anxiety front and centre, with a potentially discomforting undercurrent of humour. Is it okay to 'play' – or have fun – with someone else's pain? Am I giving you permission to poke fun at my own suffering?

My approach to game design is different from what many people might expect because I'm primarily working on my own. I don't have a huge budget or co-workers that specialize in different areas of production. Most games made by AAA companies tend to be built by many people. These games are split into parts and each part is handled by a different group. There are teams of artists, programmers, designers, producers, musicians, testers, business people, lawyers – all contributing. As a result, those games cost more to make and tend to be a lot more prescriptive in their design. There is little room for risk taking when there's a bottom line.

Since I'm working by myself, I get to experiment. I can build things quickly and I get to see the game through – from its initial conception to the finished product. Maybe it doesn't result in something that is nearly as polished, but there is a little bit of me in every part of the art, code and design of the game.

The freedom I have to make all the decisions can feel daunting at times. Sometimes it's easier for someone to tell you what to do with your time instead of making those decisions yourself. Working by myself has made me more self-

sufficient, but it has also made me extremely aware of my own limitations.

I don't have to spend time on meetings or discuss my ideas with other people in order for them to get implemented. I'm my own boss; if I want a change to happen in my game, I have to do it myself – whereas in a larger studio, there's probably a lot more back-and-forth going on between different teams in order to implement a particular feature.

It's definitely lonely at times. Working on my own has made me realize that I'm ultimately the person who cares the most about my projects and that I'm the one who decides whether or not the game gets made. Over the past couple of years, I've developed my own design process, which I've used to prototype different aspects of *Consume Me*.

Being able to articulate why you like something is as important as all the other parts of making a game. Knowing why you consider something to be good will help you to be critical of your own work. For me, I spend a lot of time browsing Instagram and Tumblr. I download a lot of games and I play them. I ask myself why they were compelling. I'm inspired not only by other games, but also by other media as well. I create folders that I fill with screenshots and images that I can reference and incorporate into my own work. Two questions I ask myself when I put something in one of these folders are: 'Why did this catch my attention?' and 'How did it do that?' I don't always have solid answers for why I like something – but it's good to put into practice. These habits have become a part of my routine – doing a little bit of this everyday has trained me to see, play and consume in a more critical way.

Before I even open my computer, I ask myself: 'What do I want to make?' If I don't immediately know the answer to that question, I usually start off with a prompt. Prompts provide enough of a constraint to focus the prototype on a specific idea. Here are some prompts if you have no clue what you want to make: one button; 10 seconds; trying; repeating pattern.

During this phase, I usually scribble down ideas in my notebook. It's helpful to draw out 'scenes' that help me visualize the sort of game I want to prototype. I usually start small – for example, I narrow down the prototype to one action that I want the player to perform. Then I ask myself how the game will end and what the goal might be. Keeping these responses simple is good for a start, because it will help me come up with a prototype quickly. I usually also ask myself how I plan to interact with this game: am I going to use a keyboard, mouse, touch controls? There are a lot of questions, but I think these are the most important:



MY PROCESS BREAKDOWN



1. BRAINSTORM

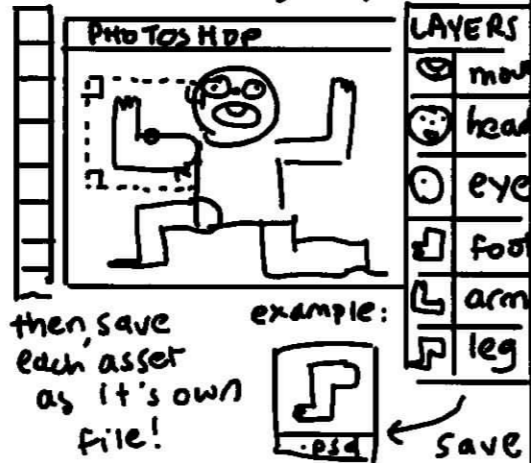
don't go to the computer just yet...



Start with a prompt!

2. DRAW A SCREENSHOT.

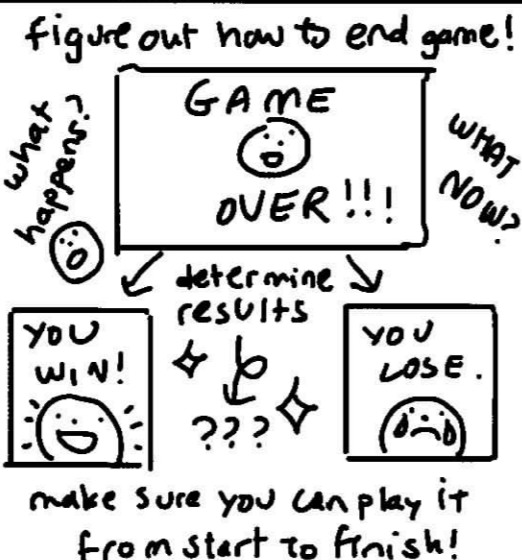
draw on individual layers (GET DOWN BIG PICTURE)



3. CODE THE INTERACTION.



4. CLOSE GAME LOOP.



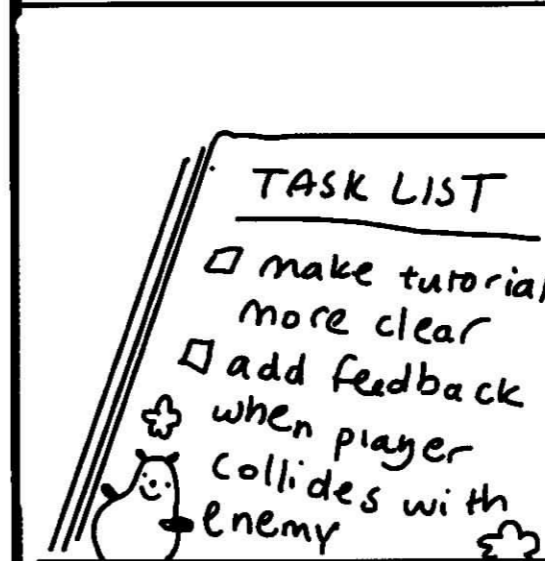
5. ADD POLISH



6. PLAYTEST!



7. INCORPORATE THAT FEEDBACK



8. ITERATE!! (REPEAT STEPS 6 + 7!)



- 1 What is the scenario or prompt?
- 2 What are the actions that the player will perform in this game? Come up with a few 'verbs' (start with just one though!).
- 3 What is the goal?
- 4 How will the game end?
- 5 How do I interact with the game?

Once I have answered these 5 questions, I move to working on my computer. Here is an example of me answering these questions using one of my prototype games Wobble Yoga:

- 1 The prompt is to 'shape-shift'.
- 2 The verb or action is to perform yoga.
- 3 The goal is to complete all the yoga poses.
- 4 The game ends once all the yoga poses are completed.
- 5 I interact with the game using the keyboard and hold down certain keys to control certain limbs.

I almost always start with making all of the art assets first. I draw everything from the background, the environment, the player, the enemies (if there are any) to even the user interaction (UI) in this one Photoshop file and I make sure every element is on its own separate layer so it is easy to separate them later into their own files.

After I've saved each asset as its own file, I import all of the art assets into Unity. Then I begin to code in the interaction. I usually start with one big script where I program all the interaction. Then I split the code up after if I can't keep it all in one big script. I tend to avoid optimizing my code too early because the point of small prototypes is to get the idea across, not to make everything general and all-purpose.

Once I've finished programming the basic interaction, I create some sort of win or lose condition. Usually I put in a timer and the player has to perform a certain action (like match up yoga poses or feed the character) before the countdown ends. Putting something in to end the game will help 'close the game loop', which will give you a sense of this mechanic you've developed.

After I've finished closing my game loop (making sure I'm able to play the game from beginning to end), I start adding sound effects and polish to the game.

Around this point, the shape of the prototype should be pretty obvious and I will play-test the game with other people and get feedback from them. Finally, I will incorporate their feedback by making changes to the game and iterating on it. These are the last couple of steps before I feel like the prototype is finished. This process is pretty loose – the answer I came up with in my notebook may change once I start making the game on my computer and that's OK – the initial ideas are just there to provide something for me to swap out later.

I ended up developing this process because it plays to my strengths. It's easier for me to think in pictures so that's why I start with a 'screenshot' of the game before I even write any code. Over time, I started adapting my art style so that I could quickly produce art for my games. I reduced my colour palette and drew everything in the required dimensions I knew I'd need for the game. Letting visuals dictate design worked for me, because it allows me to quickly get my ideas across and explore simple designs without programming stopping me from thinking. It was easy to follow this formula and I could churn out prototypes relatively quickly. As I continued to develop more prototypes in this manner, they began to all share a common aesthetic both in terms of form and content. The prompts forced me to get my point across with each prototype, and I can have fun being creative within the constraints of this template.

Despite all the positives, I hit a wall with this design process. I started encountering larger problems that could not be solved with the template I had created. Design problems began to emerge when I realized that all these small prototypes lacked any sort of 'depth' – I didn't know what to do with all these parts. I needed to figure out some sort of way to combine them. I had to go back and experiment with different design processes in order to solve these different problems.

What I hadn't realized was that when something wasn't working out with the game design, it was crucial to examine the method I was using, not just the work I produced using the method. Creating a prototyping method is useful: figure out what you're good at and what you like and play to those strengths. However, you also need to be aware of the limitations your method imposes. As a designer, your process for solving your problems always creates new ones.

¹
No Quarter is an annual exhibition of specially commissioned multiplayer games held by NYU Game Center.



88 Consume Me, game development screenshot



89 Jenny Jiao Hsia in her workspace, 2017