Tracey and the Magic Brush - EECS 494 Project 3 Postmortem

Tracey Lynes is a young aspiring artist who was on her way to her first ever art showcase when suddenly a petty witch cursed her into her own paintings! Use a mouse or a Nintendo Wii remote to draw her path to freedom in this art themed 2D platformer!

Disclaimer:

While this is a postmortem of Tracey and the Magic Brush (abbreviated as TatMB), this is only a postmortem of the development process while it was my CS capstone project for EECS 494 at the University of Michigan (Ann Arbor).

The EECS 494 Showcase

The EECS 494 end-of-the-semester Showcase went amazingly well. I went into it without much expectations of success since we did not have any time to playtest the final level/the showcase build due to technical issues with the submission, but to my surprise the only bugs that were present were not only minor, but also able to be patched out in a hotfix I was able to push during the showcase. Having three total setups worked wonders for the presentation of the game, as two keyboard and mouse setups surrounded the Wii remote setup in the center, which caught people’s attention. Almost no one said anything negative about the game that didn’t have to do with the aforementioned bugs that were patched out. In the end, we not only showcased an amazingly fun and appealing product, but ultimately won the showcase vote, with the professor himself saying it’s uncommon for a single player game to win the showcase.

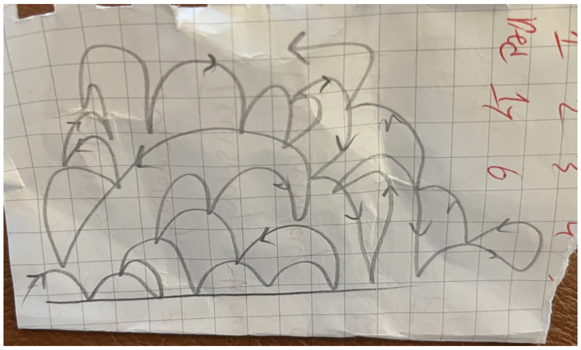
When observing the players (specifically people in my age range), the Wii remote control scheme felt natural to use. One of the players stated something along the lines of “it feels as though I have played this game before, like it was natural to just pick up the controller and point at the screen like it was an actual Wii game back in the day.” This felt amazing as those feelings of nostalgia and familiarity were part of the intended impact I wanted my game to have on players, with its simple themes, homemade art-style, and older control scheme. I also noticed that the guidance was on point, as I (from my own point of view), never had to help people understand a core mechanic of the game, as although some sections may be more difficult to pass through using the Wii remote alone, players never directly asked me what to do. Even in more difficult sections, players often figured out how to get through on their own, usually through trial and error in a way akin to games like Celeste and Dark souls: frustrating, but fair.

Creative Process (as an individual working on Project 2)

The following is paraphrased from my P2 postmortem since this was my original project:

The initial idea was to have a platformer where the mouse/cursor can draw walkable trails with varying properties depending on its color.

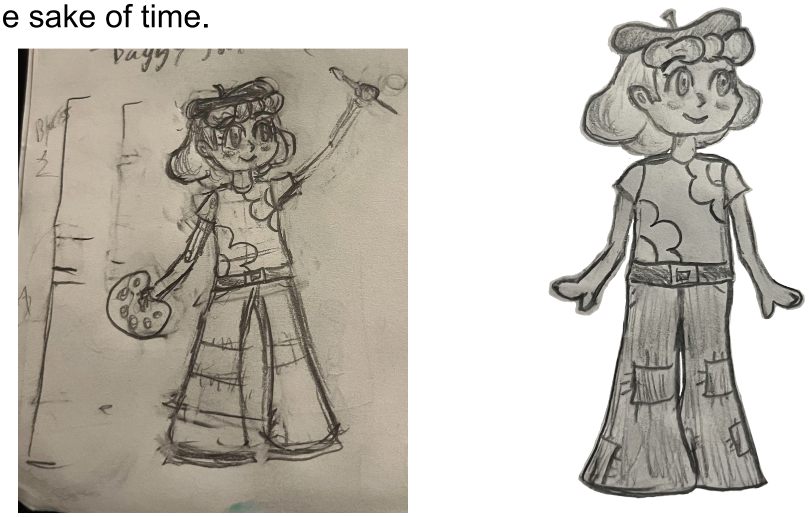
The idea was partially inspired by a “game” I would play to pass the time back in elementary school. I would use the tip of my pencil as the character, and everytime I moved the pencil on the page, the trail drawn is physical and can be jumped on (An example is shown below).



This idea was adapted to better fit a platformer style game by having the trails independent of the character's location, as it would likely be frustrating to code and/or play since one could easily get stuck. Another aspect of the game was the ability for trails to have differing properties based on the color selected, such as an icy blue trail being slippery but fast, and a green trail being bouncy. This was partially inspired by the 2012 iOS game *Nihilumbra* by *Beautifun Games* which has a similar mechanic involving painting surfaces rather than in the air.

Since the trails start to disappear after a while and the amount of trace you can use per attempt at the level is limited, it can lead to the player both needing to plan ahead before they start drawing and needing to think fast when the trails start to disappear and/or when running low on paint. Another aspect is the fact that the amount of paint consumed per trail is dependent on its length, thus the player can choose to use long single lines or choose to draw a bunch of smaller lines and risk hitting obstacles or messing up platforming.

The character you play as was a simple design, with the name, *Tracey Lynes*, which is a play on "tracing lines" being the starting point for the whole character's concept. Originally, I was going to make more drawings for different poses, but I decided against it for the sake of time.



(Left: Original concept art. Right: GoldSpike Sprite.)

Creative Process (leading a team working on Project 3)

Unable to brainstorm a game concept we were satisfied with, one of the team members suggested we continue work on my project 2, since all 3 of my teammates spoke highly of it and said it had a lot of potential. I was secretly hoping for someone to suggest that as I already had lots of ideas of ways to build upon my original design, along with essentially skipping the need for a mechanical "goldspike" since it was already implemented in my project 2.

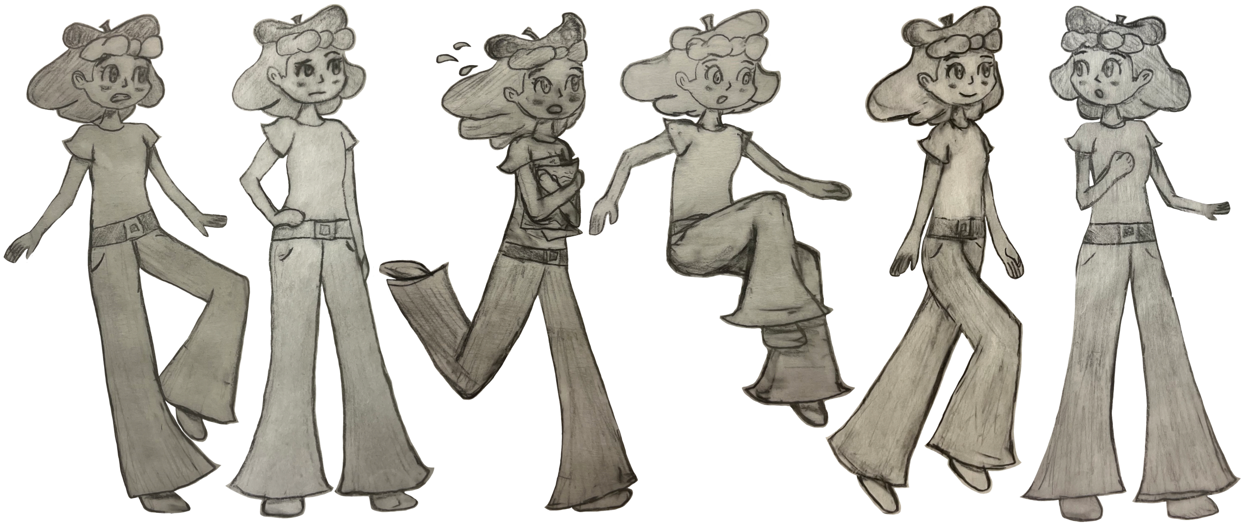
The main two goals for improving the game were 1) improving the level structure and gameplay flow, and 2) expanding aesthetic presentation; making the game not feel like a demo and more like a real and complete product/experience. The intended impact (as mentioned before) was to maintain the mechanics of the project 2 version while instilling a greater sense of adventure and engagement with the players.

This was accomplished through themed levels around each color, a (nearly) complete visual presentation including animated player sprites and a full soundtrack, and an intro cutscene to set the stage for themes of the game. Not to mention the implementation of the Wii remote, or what I call the "wavy pointy light stick", which added even more engagement. I really wanted to make this game feel more like a real game than just a demo, so to speak.

(Right: P3 GoldSpike Tracey Sprite; used as the main reference for all other sprites)

Development and Iterative Process

With the concept agreed on, we now had to answer the question of how to build off of the original concept. Since the core concept of the game had already been made, we brainstormed two main focuses for our development: Level structure and aesthetic presentation/juice. For our goldspike submission The former came through reimagining the level structure/gameplay loop, with the main focus being connecting rooms with each other and having camera transitions between them to add a greater sense of adventure/connectivity to the world of the game, and the later coming through an animation state machine which was a hybrid of Unity's built in Animation controller and and a C# script to manually change the animation state depending on player gameobject status such as being airborne or walking.



(Various Tracey sprites: one of primary means of adding more personality to the game. Each one takes about an hour to draw, and then another just to crop it out in Paint.NET)

The first milestone was spent trying to finalize new game mechanics such as camera zones, escape sequences, and the pub/sub event system, while the second milestone was spent attempting to finalize backend systems such as level transitions and music/sound systems, along with developing the first level (known as the Black level since that is the paint color introduced. All other levels will be referred to by their color or their color for consistency). While the other team members tried their hand at level design, some were not as good as others, which was helpful in the sense that it brought about the idea of specialization, with myself working mostly on level design/art/juice, and the others working mostly on backend things, new features/mechanics, and building level geometry from provided designs.

By Alpha, the second level was finished in its entirety, with a substantial improvement from the previous deliverable, along with the start of the next 2 levels, which later were finalized in the Beta and Gold deliverables respectively. During the time in between Alpha and Beta, I had spent an unreasonable amount of time working on an intro cutscene, which required storyboarding, designing, drawing, cropping, animating, and implementing the entire thing, with each of those steps taking about a whole 14 hour day to complete, but in the end, the final product was 100% worth the effort as it added that necessary cohesion to the game’s story/presentation.

By Gold (the final deliverable), all of the levels were up to our expectations, each with their own unique theming and level gimmicks to experience. The team was greatly satisfied with the final product.

Playtesting Methods

Playtesting was performed in two main ways: organized class playtest sessions with all the other class studios and course staff, and various ad hoc playtesting done by friends, family members, and colleagues of the team members. Most of the feedback was positive, and mostly had to do with the satisfying mechanics, intuitive controls, and pleasant presentation. Although the most constructive criticism came from the course staff playtest recordings provided to us in a timely manner after each deliverable. Said feedback provided advice for how to fix more complex design problems and technical issues, although a lot of said advice also was not relevant as they pertained to depreciated features such as the three-star ranking system/efficiency ranking and placeholder assets/proof-of-concept levels despite clearly being marked as such. All other feedback provided by the course staff was invaluable.

We utilized the Analyze -> Design -> Implement -> Playtest design pattern in the following ways:

1) Analyze: This step was mostly about deciding *what* needs to be done, whether through looking at the remaining Jira tasks, analyzing player feedback, or addressing a recently found bug.
2) Design: This step was about figuring out how to get the what accomplished, usually through concept art/theory crafting on paper or on the team's general notes google doc. A lot of time was spent discussing different approaches and designs while on a Discord call, often because I was the de-facto design lead in

terms of the game’s aesthetics and game mechanics, especially during the first 2 weeks of development.

3) Implement:Hours upon hours of coding, asset creation, fixing errors, etc. As team lead, I would schedule times often to meet as a team to discuss what we are all working on and what progress has and has not been made on them.
4) Testing: Actively during the development of a new feature, new level, new mechanic, etc. the members of the team would ask each other to test them to make sure they worked as intended, and comment on any feedback in terms of development progress/bugs. The other aspect of this comes in the form of the playtest sessions, which I will comment on in the following section.

The biggest change brought about from the result of playtest feedback was the addition and subsequent removal of a penalty for using small lines that barely used any paint. This was an “issue” that persisted from the original Project 2 design, and I personally had a bone to pick with it as I thought the strategy, while being “efficient” was just wasting time/slowing down the pace of the gameplay as not only was the efficiency ranking dropped, but every level, even test ones, gave plentiful amounts of ink/checkpoints to be able to finish each level without such strategy. Because of this, I Decided to implement a system for penalizing the overreliance of tiny lines within a shorter period of time, making the strategy unusable. However, from later feedback, the real solution was revealed through better level design, with rooms having lots more ground to stand on rather than every room having a large pit to cross and a limited amount of paint to do so with. By Milestone 2, we had finally decided to remove this penalty, as not only was it unnecessary, but the game/levels were more about adventure/exploration, rather than purely about efficiency. (Additionally it was removed because using the Wii remote was inherently more chaotic/jittery, thus its use favored longer lines anyway.

Management/Miscellaneous

Management was done exclusively through Jira. In terms of the task time estimates, they were mostly arbitrary. By the end of the project, the high aspirations did not fit well with the time remaining, thus we had not much of a plan other than "Grant finishes the levels and everyone else does various juice/QoL improvements since they have 3 other classes to work on but Grant doesn't". Additionally, a lot of features were cut from development such as the creation of the tutorial level, the 3rd level (centered around the cut green color), and the final level/boss fight with the witch antagonist. The Jira helped when team members didn't know what to do next, and would consult the Jira to see which specific tasks needed to be done, whereas near the end the team members knew what they wanted to work on for juice/QoL, but weren't listed in the Jira due to diminishing returns in doing so. If this was a more professional team of game developers, one of us would be designated as the main Jira manager to add tasks and keep track of progress.

We were able to keep our code very DRY (don't repeat yourself) through extensive use of the publish/subscribe observer design pattern known as the "Event Bus." Most code was able to be connected via events that enabled a much greater level of code complexity rather than having to reimplement functions or create getters/setters when unneeded. This did come at the cost of having a large number of events made by various coders, and being able to tell which events were being used by which class/script, causing for slightly confusing code at times. The solution for this (which I have partially done in my own time since finishing the class), is to properly document what events are used in which files, and describing their use cases as well to prevent duplicate events being made that in theory would serve the same purpose. Despite that, use of the event bus was crucial in our development process, and although I was a bit skeptical about its usefulness originally, it proved to be invaluable after I realized how useful and versatile it is.

While I cannot comment on my teammates, I found myself seldom using AI websites to generate code for this project, partially because most of my code was already written from project 2, partially because the AI chat sites had no knowledge of the nature of our eventbus and its implementation, and partially because a sizeable portion of my side of the development process was centered around non-coding aspects such as drawing sprites, building level geometry, reorganizing menus, etc. For the few times that I did use AI, specifically ChatGPT, it was mainly to get code to achieve a simple task when I was stumped, but due to experiences from project 2's design process, I decided against using AI for script writing for the most part. I feel as though the less general your code is, the less applicable it is for AI to write it.

Going Forward

This is without a doubt the greatest creative project I have ever made in my life. Mostly because I start a lot of projects but seldom do I make much progress on them, let alone finish any.

A lot went right with the project, mainly the level designs, gameplay loop, aesthetic/theming, soundtrack, and general presentation of the game. I could not have asked for a better game to work on, nor a better team to do it with.

The main thing that went wrong during this project was a lack of proper specialization. While some people were able to work on general coding tasks and a few were able to do level design/some art, I was the only one who could do large time investment tasks such as the intro cutscene, creating character sprites, making surface and background textures, etc due to this being the primary class I had during the semester whereas some teammates had upwards of 4 other classes to worry about. This lead to a logistical nightmare in terms of the project scope greatly getting out of hand, and as I frequently said during this project: “I was the one who bites, thus I am the one who must chew,” referring to how the burden of the game’s high aspirations were mostly put on myself, leading to a quick descent into sleep schedule disarray as I would find myself nonstop working for up to fourteen hours a day for most days of the week (this was one of only two classes I had during the semester). It was all worth it in the end though as the final product was polished to a sheen.

In my future game development endeavors, I will hopefully have team members who are more appropriately specialized as to ensure productivity and efficiency in accomplishing tasks, along with setting more realistic pairings of goals and deadlines that don't necessitate such a crunched development process. Notably, I am still continuing work on TatMB to this day, and plan to include 2 more levels, a final boss fight, accessibility/assist settings, a complete art gallery unlocked through collecting the various flowers throughout the levels, etc. I hope to have it released on steam by the end of february, but I would like it to be finished before the next semester's showcase so I can possibly have it displayed there too like that other game by that solo dev at this year's showcase. Stay tuned for that!

In terms of how I might better employ AI assistants in future projects, the simple answer is "don't unless it is simple." As I stated at the end of all of my game trailers for TatMB, I firmly believe that AI art and assistants are not required for anything that I do and I take pride in my abstinence from them. The only place that I might find it helpful is for getting a very simple code such as "make a script that can toggle all game objects in a given array off except the one at the given index" or something like that.

EECS 494 Reflection

The largest takeaway I took from this course is that I actually have a bit of talent for creating games. I always was interested in possibly trying it out, but never felt as though I had the time or the means or the skill to do so. Thanks to this course, a whole new world of opportunities have opened up for me as I genuinely consider going into the field of game development, possibly looking into local game studios in michigan and/or continuing TCY-Games (TCY was short for Tracey but also could stand for “Take Care of Yourself”) as an actual independent game studio lead by myself, with TatMB as my debut game (my brother even designed some merch already). A lot of people supported me and my game over the last 3 months, and so I see no reason not to consider going forward with it.

Special thanks to the EECS 494 staff for guiding me through such an amazing course. I hope I can stay in touch with the staff and possibly go to the game test sessions I heard of where Austin Yarger (class staff lead) displays Greek Tragedy to display a more finalized version of TatMB.