Halfway Home

A GAM 450 Postmortem

Overview

This document contains post-semester reflections on Makeshift Monocle's stat-management visual novel game *Halfway Home*. The project team was 5 strong, with some external technical assistance. My co-director Jesse Lozano and I partnered up to handle all development, design, and writing, while the other three team members provided much-needed art and audio assets to support our vision.

I'll discuss what our goals were for the project, the major challenges we faced throughout development, the general outcome, and finally the lessons we learned and want to pass on to anyone looking to create a project in a similar vein.

Everything stated below comes with the disclaimer that Halfway Home's development period has not ended. This postmortem is for the *demo* which shipped at the end of the semester.

Goals

When Jesse and I set out to make this project we decide on a set of pillars to guide development.

1) Ensure the player accomplishes something with each failure.

We decided early on that we wanted to allow the player to fail because consequences create meaning. That doesn't mean, however, that we wanted the player to feel like they've wasted their time. We wanted the player to be able to learn from their failures in a mechanically tangible way, that way failure can even be a reward.

2) Teach all the rules in the introduction.

This later expanded to include character introductions as well. This is a pretty obvious principle for almost any game, but it bears repeating.

3) Reduce repeated content.

To facilitate forgiving the player's failures while also not ballooning our narrative scope and keeping to our intended plot, we decided on a time loop mechanic to allow the player to take on the same challenges again and again until they succeeded. However, with this solution came the danger of boring the player with repetitive content. We ended up cutting loop content out of the demo, but I still kept to this principle when writing the Room Defaults, tiny generic scenes that play in each location when no special scenes are available. Each of these defaults has randomly generated descriptions that build unique-micro scenes for the player to avoid them feeling overly repetitive.

4) Focus on narrative.

Jesse and I love narrative and DigiPen rarely gave us academic opportunities to practice our craft. We had to create our own opportunity and made sure to make it a pillar.

5) Encourage emotional thinking.

There are a lot of games that ask you to think and a lot of games that make you feel. We love both and wanted to make a game that makes you think about feeling.

6) Keep to a focused setting and characters with no "wasted space".

We recognized early on that it would be easy for us to get excited and bloat the narrative, so we created this pillar to keep us vigilant for any scope creep that would make for a less effective (and more expensive) experience.

Even as the project evolved and shrunk from its original scope (more on this in later sections), we kept to these core values. In fact, by the end of the project, we added two more:

7) Create engaging and empathetic characters.

As the project developed it became increasingly clear that much of the engagement hinged on the strength of the characters, and that the mechanics and setting acted to support them.

8) Combine narrative and systems design.

Jesse and I are passionate about both areas and wanted to hone our skills in these disciplines as well as demonstrate them to potential employers.

Challenges

The challenges we faced on Halfway Home fall into three categories: Subject, Production, and Genre.

Subject

The earliest obstacle we faced was the gravity of the subject matter itself. Mental Health is a serious topic and failing to capture it properly could not only have made the team look incompetent, but even malicious. We aspired to tackle a challenging topic because we believe in the need to normalize these issues and present them in an honest and human light. A lot of research was needed to ensure that our portrayals were accurate and insightful. We drew upon our own experiences and those around us. We spoke to friends and family who struggled with these issues. We studied mental disorders and clinical practices to get a the clearest possible perspective and concepted our characters and setting around the facts rather than our preconceived notions.

At the end of the day, we knew that we could never put in the time required to be prescriptive in our writing, so rather than focus our narrative on how to deal with each mental issue clinically, we decided to emphasize the importance of the universal medicine of friendship, openness, and self-acceptance.

Production

Production hurdles were all too common. For one, despite the deliberately low technical scope of the project we lacked an art or audio team capable of keeping up with our narrative ambitions, a difficulty we fortunately managed to mitigate early on with recruitment. In retrospect, we got very luck to attract the talent that we did *when* we did, as bringing in art and audio any later would have been disastrous. This is an example of something that we should have caught in pre-production, which brings me to my next point.

We left some things out in pre-production that *definitely* should have been developed further. We didn't spend nearly enough time defining our character arcs, which created a lack of clarity that slowed production until the second semester when we finally recognized the problem. Furthermore, while we dedicated time to developing a distinct voice for each character, the player character's voice was left vague, a problem which took us until mere weeks before demo launch to realize.

Finally, production suffered from time issues. As full-time students, the team lost a lot of time to outside projects and assignments. In particular, the audio team was crippled for months and only picked up when we hired a guest composer to help out. Additionally, we lost a lot of time debugging as a result of our project having two separate technical frameworks. Jesse's framework could be debugged quickly, but the framework that interfaced with Ink (the scripting language used for our story segments) was developed by an external consultant, meaning any issues with it took much longer to resolve.

Jesse and I were often bogged down by tasks that were necessary to the project, but professionally unproductive. As the more proficient with UI/UX of the two of us, it was decided that I would handle the game's user interface. This was the best decision we could have made without hiring someone to focus on it, but it took a ton of time away from me that I could have spent on writing and systems design. In hindsight, we should have put far more effort into hiring someone to handle this aspect of the project for us.

Genre

Visual novels are a niche genre, which presented a few issues for us in testing. For one, the professors were largely unfamiliar with the genre. That lack of literacy limited the speed and utility of their feedback. Likewise, the majority of our playtesters had never played a game like ours, meaning that we had to determine what was a response to the game versus what was a response to the *type* of game.

Visual novels are also a longer-form medium, meaning that testing took a long time. This content scope also naturally made for more editing work and detracted from the polish of the final product.

Finally, this was our first time working on this type of project. We love the genre but had never worked in it before. This meant that while we had a good understanding of what the final project should look like, we had a very poor understanding of the pipeline for creating it. We made a lot of "rookie mistakes" that would be easy to avoid if we did it all again. For example, when requesting poses for each of the six characters from our artist, we asked for a standardized list of emotional primitives, e.g. happy, sad, angry, etc. This turned out to be problematic for two reasons: First, each character had different emotional tendencies. Some characters never looked afraid while others were always happy. Second, while emotional primitives are common to all, they also aren't very interesting. We asked for six standard poses for each character when we should have asked for four unique poses that conveyed more personality.

Outcomes

The fruits of our labor. How did we succeed? How did we fail?

As a student project, we lack the ability to quote sales figures, downloads, or review scores, but we are confident that overall the project was a success. It's difficult to tell how successful the game is due to a lack of players and our position as students. That being said, the team agrees that Halfway Home is the most polished product any of us have made yet. In the end, while we were unable to ship the entire game in April as we initially intended, we shipped a core minimum that has inspired the team to keep working on it and push for a Steam release over the summer.

We met our project goals handily, though on occasion not without consequence. For example, we introduced everything in the intro, but this made the game's opening feel slow to players. Our most significant engagement spike is an hour (or more) into the game and most testers never got that far.

Our greatest achievement has been fostering a genuine connection between our players and the characters. Even among testers who don't care for visual novels, we have gotten a strong indication that players identify with the core emotions and perspectives presented to them. We successfully systemized mechanical metaphors for mental health and self-improvement. These systems complement and support the narrative and should effectively demonstrate Jesse and my ability to blend said aspects of design.

The Takeaway

What did we learn? What can others gain from our experience?

Despite Jesse and my recognition that we needed a polished portfolio piece to show to employers, learning remained my first priority on Halfway Home, as it has with all of my student projects. I challenged myself to work with difficult subject matter, which exposed me to a lot of information and personal stories about mental health. I took on the most writing work for a single project in my life thus far and was instrumental in determining the structure of the entire narrative.

Jesse and I got a lot of things right, looking back on it. We were unafraid to make cuts and always prioritized the whole experience over any part. Furthermore, whenever we got into a creative dispute, which was often, we learned to shift our perspectives, allowing us to find an innovative solution by identifying the problems we were trying to solve rather than arguing over the details of our proposed solutions. Coming from larger teams in our previous classes, we experienced firsthand the benefits of a smaller team. We were nimble, ready to adapt, and always had a solid understanding of the project's status and trajectory. This awareness allowed our scope to shrink naturally over time without any hurt feelings. Every time I found myself thinking it was about time to cut a scene or a feature, Jesse was on board and already thinking the same.

Jesse and I learned a lot of things the hard way, too. For example, as designers we neglected to solidify technical standards for the project in the same way that we determined narrative and design practices, which slowed us down. As discussed earlier in the challenges section, we made a lot of mistakes born of a lack of familiarity with the pipeline. Working closely with audio was also a first for both of us it shows, as the audio pipeline was the weakest by far.

Overall, I learned an amazing amount on this project and I am very glad I chose to be a part of this experiment.

Jesse and I created a list of advice for anyone working on a visual novel at DigiPen in the future that we intend to flesh out in more detail, but I've gone ahead and put the abbreviated version here.

10 Things You Should Know If You're Making A Visual Novel At DigiPen

1. You need artists.
2. You need sound designers.
3. Allocate a lot of pre-production time.
4. Playtesting will be a pain. (lots of content, niche genre)
5. Know your tech. Are you making it yourself (better quality, longer dev time) or using a pre-made tool (less quality, shorter dev time)
6. Know what is E10 and what isn't.
7. Your game is UI.
8. Know how to write a story. EX: How do you start?
9. Scope small.
10. DigiPen will still want you to make a game.

