

Rochester Institute of Technology

B. Thomas Golisano College of Computing and Information Sciences

Master of Science in Game Design and Development

Capstone Final Design & Development Approval Form

May 3, 2022

Student Name: Emily Horton (She/Her)

Student Name: Veronica Vitale (They/Them)

Student Name: Hanrui Zhang (He/Him)

Student Name: Rye Ress (Any/All)

Project Title: Mending Hearts

Keywords: Experimental, Cozy, Relationships, LGBTQ, AI, Level Design, Narrative

Elouise Oyzon (She/Her) 

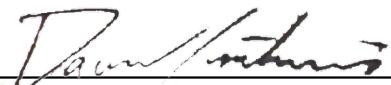
Lead Capstone Advisor

Erika S. Mesh (She/Her) 

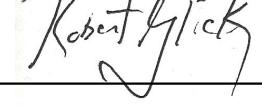
Capstone Process Advisor

Eric Baker (He/Him) 

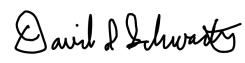
Faculty [Advisor]

David Simkins, Ph.D. (He/Him) 

Faculty [Advisor]

Robert Glick, Ph.D. (He/Him) 

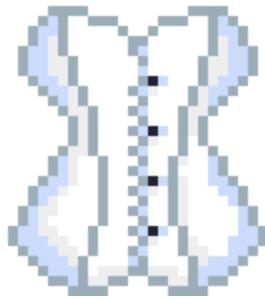
Faculty [Advisor]

David Schwartz, Ph.D. 

Director, School of Interactive Games and Media

Mending Hearts

by Berry-Stained Corset



By

Veronica Vitale

Emily Horton

Hanrui Zhang

Rye Ress

Project submitted in partial fulfillment of the requirements for the degree of
Master of Science in Game Design and Development

Rochester Institute of Technology

School of Interactive Games & Media

B. Thomas Golisano College of Computing and Information Sciences

May 3rd, 2022

Acknowledgments

David Simkins for his wise guidance on game design, narrative design, and relationship dynamics

Robert Glick for his practical advice in the process of writing good interactive fiction, as well as his valuable personal feedback from our writing drafts.

Eric Baker for being a cool dude who gave us practical game development advice, as well as his perspective on relationships.

Erika Mesh and Elouise “Weez” Oyzon for their mentorship, their encouragement, and their butt-kicking.

Jill Bray, for her hard work in coordinating everybody’s busy schedules at the end of each semester.

Executive Summary

This document is a record of the process that our team, Berry-Stained Corset, followed to develop *Mending Relationships*, a cozy, sentimental, and uplifting game about two women learning to live together again after being in a long-distance relationship for two years. From the start of development, we had a clear picture of what the emotional core of the game would be, but how we would deliver that experience to our players was completely unclear. Throughout the development, we made several major pivots on the game's story and mechanics based on both playtester feedback and based on our small team size. Throughout these shifts, we stuck to the same emotional core of the game and we eventually discovered the game structure, mechanics, and story that we wanted to create.

As part of our effort to make *Mending Hearts* a unique and meaningful experience, each team member also worked on their own research. Emily Horton examines the mechanical representations of romance and relationships in games and the considerations that designers should make when setting out to design romance. Rye Ress worked on creating the AI for Juno in order to make her feel like less of a fixture of the house and more like a person. Hanrui Zhang researched level design, including how to create the layout of the games indoor and outdoor scenes in a way that complements the story and immerses the player in the emotional experience. Veronica Vitale's research focused on what the player did inbetween story beats and the design behind that. All of these topics expand on the visual novel, dating simulation, and life-simulation genres combined in an attempt to give the player a unique emotional experience that will stick with them long after they finish playing *Mending Hearts*.

Due to the many shifts in direction, there is still a lot of room for polish and expansion to create the full experience; As-is, *Mending Hearts* is a basic prototype. Although we have designed and programmed several interacting systems to develop on the game's core experience, much of these systems do not fully shine in the game's prototype due to our team's many concept pivots, organizational challenges, and smaller-than-average team size. At the end of our capstone experience, we have created a proof-of-concept for what a fully-developed *Mending Hearts* would look like.

Table of Contents

Acknowledgments	3
List of Figures	9
List of Tables	11
Introduction	12
Audience, Genre, and Market Influences	13
Audience	13
Genre	14
Market Influences Table	14
Conclusions	16
Background Research	16
Production	18
Overview	18
Team Organization	19
Team Formation	19
Roles & Responsibilities	19
Risk Analysis Table	21
Game Development Process	23
Task Management	23
Communication	27
Internal Team Communication	27
Stakeholder Communication	27
Planning & Scope	29
Overall Priorities	29
Minimum Viable Product	29
Timeline	30
Initial Concept & Research	30
First Prototype (Paper Furniture Crafting)	31
Second Prototype (Digital Furniture Crafting)	32
All Faculty Pitch	33
Rockstar Pitch + Documentation Update	33

Third Prototype (Testing New, Basic Systems)	34
Capstone Development Kickoff + Spring First Sprint	35
Spring Sprint 2 + First Playtest	35
GDC Build	35
Final Deliverable	36
Stretch Goals	36
Version Control	37
4.8.1 Source Code	37
4.8.2 Project Artifacts	39
Asset Pipeline	40
Game Design	41
Overview	41
Player Objectives	41
Emotional and Narrative Core	42
Gameplay	43
Aesthetics	43
5.5.1 Influences for Game Art and Feel	44
User Interface	47
Assets	49
Technical Design	52
Summary	52
The Player	52
Interacting with the World	53
Creating Levels	53
Managing the Story	53
Showing Dialogue using Fungus	55
Juno, the Autonomous Girlfriend Agent	57
Tracking the Relationship	59
As a Romantic Partner	60
Playtesting	61
Overview & Approach	61
Initial Playtest - Cardboard Furniture	62
Playtest 2 - Digital Furniture	62
Playtest 3: The MVI of Cottagecore Living	63

Playtest 4 - First of the New Year!	64
Playtest 5	66
Playtests 6(GDC) & 7(Asynch in-house)	66
Playtest 8(Imagine RIT)	67
Future Work	68
Postmortem	69
Overview	69
Reflections	69
Game Design	70
Technical Design	70
Production	71
Playtesting	72
Conclusions	73
Future Work	73
Appendices	75
Individual Research	76
Abstractions of the Heart: How to Design Better Romance	76
By Emily Horton	76
Abstract	76
Problem Statement	76
Background	77
Significance of the Problem	84
Methodology	84
Results	89
Discussion	89
Conclusions	89
Future Work	89
Knowing What You Want But Not How to Get There: A Look into Designing the Mechanics of Mending Hearts	90
Abstract	90
Problem Statement	90
Significance	90
Background	91
Methodology	93
Results	94

Discussion	96
Conclusions	97
Future Work	97
Storytelling Level Design -- Hanrui Zhang	98
Abstract	98
Problem Statement	98
Significance	98
Background	99
Methodology	102
Results	104
Discussion	106
Conclusions	106
Future Work	107
Bringing Autonomous Agents to Life	108
Abstract	108
Problem Statement	108
Significance	108
Background	108
Methodology	116
Results	117
Conclusions	119
Future Work	120

List of Figures

Figure 0	23
Figure 1	24
Figure 2	25
Figure 3	26
Figure 4	31
Figure 5	31
Figure 6	32
Figure 7	32
Figure 8	34
Figure 9	39
Figure 10	40
Figure 11	44
Figure 12	45
Figure 13	45
Figure 14	46
Figure 15	47
Figure 16	48
Figure 17	48
Figure 18	49
Figure 19	49
Figure 20	49
Figure 21	50
Figure 22	50
Figure 23	50
Figure 24	50
Figure 25	51
Figure 26	54
Figure 27	54
Figure 28	55
Figure 29	56
Figure 30	56
Figure 31	57
Figure 32	59
Figure 33	84
Figure 34	116
Figure 35	120

List of Tables

Table 0	14
Table 1	21

1 Introduction

Mending Hearts

What's two years in the face of the rest of our lives?

After two years, Ada is reuniting with her girlfriend, Juno, and they are moving into a new home together. In *Mending Hearts*, you act as Ada and make decisions that will alter the course of the relationship. Unpack boxes, cook meals, and converse with Juno in order to reconnect with her. Can the two of you work together to establish a new normal for the relationship, or have you both grown in different directions?

Mending Hearts is told in a total of four vignettes temporally grounded in one of each of the seasons. These vignettes represent a pivotal period in the couple's relationship. The player navigates the tension and uncertainty of trying to return to "the way things were" when Ada and Juno have each individually grown while dating long-distance. While completing the activities for each vignette, such as cooking breakfast or going on a hike, both their dialogue choices with Juno and the actions they do around the house will affect Juno's feelings towards Ada for better or worse.

The story progresses both by completing each vignette and by unlocking extra memory scenes that help contextualize the relationship between Ada and Juno and what makes them a good couple. The player is encouraged to explore their environment and learn Juno's patterns. They should not be immediately punished. Rather than focusing on a win or fail state, each vignette is just a unique moment of the story tailored to that player's specific playthrough. Though a breakup between Ada and Juno as the game's ending could be considered to be as fail state, the game portrays it as an organic process of growing apart that is healthy to respect and understand.

Mending Hearts is designed to be a cozy, sentimental, and uplifting experience that can be played in the span of one afternoon. Its genre can best be described

as a life simulator game with an emphasis on romance mechanics. *Mending Hearts* is not quite following any one genre convention and is in fact challenging some of the conventions of the genres it's composed of.

The bulk of this document enumerates the major details of our team's journey in creating *Mending Hearts*, covering background research, team organization and processes, and both the game and technical design of *Mending Hearts*. Next, we discuss playtest results and other feedback we received throughout the development of *Mending Hearts*. We then close the main portion of our document with a post-mortem analyzing our team's successes, failures, and opportunities for future work on *Mending Hearts*. Finally in the appendices is each team member's individual areas of research that influenced *Mending Hearts*.

2 Audience, Genre, and Market Influences

2.1 Audience

Our audience is one that enjoys games like they might enjoy a warm cup of tea on a chilly afternoon. They are players that, regardless of gaming experience, enjoy experiencing a relatively short, self-contained story that they may participate in. This game could be referred to as a casual game because of its short total playtime and its low skill barrier¹. However, saying that this game's audience is casual gamers says little about our audience's needs; we must try to be more precise in our definition.

Our characters are part of the LGBTQ+ community in part because we want to add to the relatively small amount of representation the community gets in video games. Because *Mending Hearts* deals with relationships, and more specifically LGBTQ+ relationships, our audience expands to any members of the LGBTQ+ community who want to see their people on-screen in a game.

¹ Baker, Eric. "What Is a Casual Game?" *IGME 450*. Lecture, January 4, 2021.

2.2 Genre

Mending Hearts embodies core mechanics and features from three specific genres: the Life Sims, Dating Sims, and Visual Novels. Mechanics and conventions from each of these genres all come together to serve our goal of delivering a player experience that is cozy and emotionally meaningful.

Life Sim Elements

Core gameplay consists of mundane, real-world tasks that the player might do in their daily life.

Dating Sim Elements

The elements of dating simulators, as well as romance in video games in a broader sense, can be read in Emily's [individual research](#) section.

Visual Novel Elements

The visual novel elements of this game include the dialogue boxes used to deliver dialogue and story points as well as the dialogue options that the player can choose from when interacting with Juno.

2.3 Market Influences Table

Game Title	Influence
<i>Animal Crossing: New Horizons</i>	Player homes feature a broad customization engine with an in-game system for labeling furniture and scoring a player's interior design decisions.
<i>Callico</i>	A game that is designed from the ground up to be a relaxing and vibrant experience. Players decorate and manage their own personal Cat Cafe in a magical world. Players buy, place, and arrange furniture, and also play in a variety of minigames to bake treats and create drinks to sell in their shop.

Game Title	Influence
<i>Façade</i>	An early example of an AI-based conversation system, <i>Façade</i> uses language parsing to interpret the player typing responses to what the NPCs say in real-time and give the player the power to influence how this story about a married couple unfolds.
<i>Gris</i>	A short, impactful game that primarily focuses on visual storytelling through poignant and vivid visual aesthetic.
<i>Mortician's Tale</i>	An incredibly short game experience with a huge emotional impact for its time commitment.
<i>The Sims 4</i>	Sometimes referred to as a “virtual dollhouse”, the gameplay consists primarily of building and decorating homes, creating characters with different personality traits, then watching or directing their lives unfold. Individual Sims behave according to a large number of internal stats that represent their current state as a living person. <i>The Sims 4</i> takes the series further by adding emotional states to sims which both limit their options for interacting with the world and gives them special actions they may only perform while in that emotional state.
<i>Stardew Valley</i>	The bulk of the gameplay in <i>Stardew Valley</i> revolves around real-world farming chores that the player can do. Additionally, <i>Stardew Valley</i> features a friendship and romance system that, while flawed, has a few unique takes on genre conventions.
<i>Undertale</i>	NPCs in the game react to your actions (including meta actions, such as saving and reloading the game) in novel, unexpected ways for the RPG genre.
<i>Signs of the Sojourner</i>	Conversations with NPCs in the game are won or lost based on the players ability to connect with

Game Title	Influence
	them in short, cooperative card games built on an easy-to-learn symbol-matching system that represents the different emotional approaches to having a conversation with another person. Succeeding in a conversation means being able to match your symbols (communication patterns) to those of the other person.
<i>To the Moon</i>	A 2D top-down style story driven game. The story is also about a relationship. The levels are designed to represent and communicate the emotional beats of the story.

Table 00: Chart of games we took reference from with their title and why we referenced them

2.4 Conclusions

The games we looked into had a wide variety of mechanics and told various stories. While many games we looked into did try to portray relationships in one way or another, they did not have the complexity that we wanted to show with *Mending Hearts*. We saw this as a great opportunity to create an experimental game that could be used as a stepping stone for other developers to expand on relationships in video games. We also see *Mending Hearts* as an opportunity to increase the amount of LGBTQ+ representation in video games.

3 Background Research

In addition to basic market and demographic research, each member of Berry-Stained Corset researched an individual topic that backs up their contributions to *Mending Hearts*. Although we are all creative individuals and skilled within our disciplines, *Mending Hearts* does not exist in a vacuum and we each opted to research how designers, programmers, and artists in the field have solved similar problems to ours while documenting design theories and best practices that we should consider following. Here, each team member's research is summarized to explain its relevance and its impact on *Mending Hearts*. Each team member's complete research begins in [section 11 of this document](#).

Designing the Visual and Mechanical Aesthetics

Our artist explored many different art styles that might work well for this game, with the intention of finding a style that both looks appealing in a digital space and serves the cozy, homey design of the game's story and interactions. They have since moved onto looking into other life-sim games to see what mechanics they use to keep the player engaged between story beats. This began by looking into designing the home and how to allow players to express creativity in games while also having it impact gameplay. Though, due to scope constraints this would eventually be scrapped for a simple interaction system that allowed the player to interact with items in the house. Research for this system mostly focused on encouraging the players to explore and interact with the home.

Designing a Romance System

Our writer and design lead has explored various storytelling techniques in video games, and has also researched common limitations with NPCs in video games, along with games that try to break the mold through complex backend systems or alterations to player perception of dialogue.

Designing an Autonomous Girlfriend

Rye explored everything that is autonomous agents in video games. They looked into existing archetypes of autonomous agents, existing workflows for them, and even examples of existing architectures. Using this research, Rye went to work coding the autonomous agent in *Mending Hearts*, Juno. Rye followed the established workflow of Observe, Think, Act where Juno observes the world, thinks about what goal to achieve, and acts on that goal. Rye successfully implemented a Triggering Agent in the form of Juno who can interact with the world and Ada.

Designing Levels that Tell the Story

Our level designer has looked into other 2D story-driven games that tell stories about relationships to see how their level designs work in helping the storytelling and also read some papers about design patterns in level design to find out some design patterns for telling a story about relationships for *Mending Hearts*. He looked into guidance, flow and pacing control, and environment storytelling in

level design to create Mending Hearts' indoor and outdoor scenes in a way that emphasizes the story, gives players cozy and comfortable feelings, and immerses the player in the emotional experience. Based on keeping players immersed in the emotional experience, he tried to add some environment details to those levels to give players some clues about the story that happened in Mending Hearts.

4 Production

4.1 Overview

The main factor that brought each of our team members together was the design goal of eliciting an emotional reaction in players. We wanted to use mechanics to tell a personal story of romance between two characters, inviting the player to direct the story and experience the emotional valences of the relationship they participated in.

Our team's production decisions—especially deciding task capacity for each sprint—prioritized the mental health of our team. We recognized that each member of our team is a busy grad student navigating their final year of a lengthy academic career, and we each had our own responsibilities and life challenges to deal with. On top of it all, we are still dealing with the COVID-19 pandemic, which presents frequent unknowns. All of these factors contributed to our decision to prioritize team member well-being and flexibility over process ceremony and a general sense of urgency.

As we planned a more concrete course for *Mending Hearts'* development at the start of development, it became clear that our initial ultra-low ceremony approach was not working for our team. We began better organizing and tracking project tasks and we established more meeting times than we had anticipated needing. Additionally, we established regular meeting times both for the core team and for communicating with our committee.

4.2 Team Organization

While forming our team and looking into responsibilities, we understood that we are all technically proficient and can help with the coding of the game. That being said, a few members wanted to use this as an opportunity to explore more avenues of game *design*, such as writing and art production.

Initially, we identified one project owner who had a hand in every decision in the games development, one artist who was dedicated to designing our game's aesthetic and creating assets, one programmer who would focus on *implementing* the design of the game's relationship, and a level designer. However, our team's small size forced every member to take on additional responsibilities throughout the entire development cycle and each member's role became much more loosely designed.

4.2.1 Team Formation

The team formed at the start of the 21-22 academic year when Veronica pitched an idea to the rest of the class that Emily loved. The two of them worked together on fleshing the idea out prior to the start of classes in Fall. Later, Hanrui joined, expressing interest in the idea and in working with Veronica and Emily. Rye was the last (though certainly not least) to join the team. Emily, Veronica, and Rye have extensive experience working with one another on the game *Boiling Over* as part of their team from the previous year, Plush Bunch. This familiarity is partially what brought Veronica and Emily together (in addition to their mutual excitement for the initial idea).

Emily and Veronica have continued to be the creative directors on the project, but every team member has both the right and responsibility to provide input. We believe that a collaborative culture in a game studio will produce the best possible game both because everybody has a unique perspective and a unique skill set and because shared ownership of a project drives engagement and motivation within a team.

4.2.2 Roles & Responsibilities

Veronica Vitale

Art & Design, Programmer

Veronica is mainly focused on designing the aesthetics for the game and ensuring they go further the narrative and feel of the game. As well as looking into the interactions the player will have in the game inbetween story beats to really immerse them into the experience.

Emily Horton

Dialogue Designer, Programmer, Writer

Emily is in charge of all things project management. She is responsible for managing the team's Trello board, hosting team meetings, maintaining the git repository, and communicating with external stakeholders. Although this is an important and demanding job, Berry-Stained Corset's frequent meeting schedule and robust online communication make it a matter of just keeping things organized instead of constantly corraling teammates.

Emily's direct contributions to *Mending Hearts* are in the form of design. She is designing the game's relationship system and the ways that the player's actions affect Ada's (the player character) relationship with Juno.

Rye Ress

Programmer, Designer

Rye's main focus on this project is as a developer programming the mechanics of the player and the AI of Juno. Aside from coding new mechanics, patching up bugs, and creating a believable Juno, Rye designs and implements feedback systems.

Hanrui Zhang

Level Designer, Programmer

Hanrui mostly focuses on the level design part for the game to make sure the levels give the players the right feelings about the narrative and the game. Beside level design, Hanrui will also help with coding and debugging for this game.

4.3 Risk Analysis Table

Risk	Likelihood	Severity	Mitigation
Limited synchronous meeting time.	Low	Moderate	Using standups to keep up to date with everyone when synchronous time is limited. Prioritizing synchronous time for things that really can't be done async.
Unexpected sickness or need to be offline.	Med	Low	Communicate often and early with the team so that we can make the necessary adjustments before it gets to be a large issue.
Conflicting Responsibilities	Med	Moderate	Communicate often and early with the team so that we can make the necessary adjustments before it gets to be a large issue.
Intragroup Arguments	Low	High	Communicate with each other and try to find a compromise that will work for everyone.
Burnout + Mental Health	Med	High	Communicate with each other before and take one or two days off maybe.
The Game's Story Does not deliver the emotional impact that the team intends	Med / High	Catastrophic	Playtest often and make sure we are asking the right questions and listening to their feedback.
Narrative Aspect is not related to	High	Moderate	Playtest often and make sure we are asking the right questions and listening to their feedback.

Risk	Likelihood	Severity	Mitigation
Mechanical Aspects			
We lose sight/focus of the vision for the game	High	Moderate	Frequent communication with committee members is key to mitigating this risk. Stepping outside of the team's inevitable production grind and assessing the game from a holistic perspective can act as a reality check for the team. In short, we need to come up for air every now and again.
Loss of Resources	Low	Catastrophic	Continue to treat the Grad Lab with respect and keep it in good condition so we will continue to be able to use the space and its resources for development. If COVID-19 continues to progress and moves RIT to again go full virtual, we will continue to work on our home machines. All team members have personal computers capable of running Unity and continuing development work.
We fall into production grind and ignore design	High	Catastrophic	By losing sight of what our vision for the game is, we would stop asking difficult questions and we would stop designing with intention. The biggest way to mitigate this risk is to meet frequently with stakeholders. Eric Baker is the type of designer who will always ask what the player experience is and ask us what our intention is. He is an excellent resource for design and sanity checks.

Table 01: Chart of the different potential risks, their likely-hood, severity if they do hit, and how to mitigate them

4.4 Game Development Process

4.4.1 Task Management

Sprint Planning

During Capstone Design, we did not follow a sprint structure *at all*. This is because our priority during that semester was to discover what our game wanted to be (and to an extent, what game we wanted to make). Because of the ever-shifting nature of our project's needs and priorities, trying to plan more than a week ahead of time felt fruitless to the team. However, our approach changed when we entered Capstone Development. We decided that we needed to track tasks both visually and asynchronously so that we could coordinate outside of meeting time and have a representation of overall progress and velocity during each sprint.

Sprint 2

Rating System is Roughly Planned
Rough Juno AI (Plan + Basic Implementation)
Write Juno Day-to-Day Dialogue
Continue/Finish Planning Story + Beats
Finish Interior Tileset if Not Done Already
Begin working on new levels (if possible - tileset must be completed)
First Pass at New Interactions
Work on Assets for New Interactable Objects
Revamped the heck out of the grid system

1/27/22 Playtest: Is all of our stuff actually fixed?

- Are players way less confused?
- Do players understand how to pick things up and use things?
- Do players understand the context and not want to murder Juno?

Sprint 3

Juno Can Talk To Ada
Relationship State Exists, Even if It's Very Simple
Prototype of Rating system (Veronica will be breaking this out into task(s))
New Home Interior Should Be Done + Implemented
Write Outdoor Scenes
Begin/Resume Outdoor Level Designs (The Yard, The Surrounding Woods)
Furniture/Home Decor Assets
Sink Assets + Flower Planting Assets

Figure 00: A screenshot of our sprint-planning document. Lines highlighted in green were successfully completed during a sprint. Lines in yellow are ongoing, while lines in red were not worked on. Lastly, extra work that was added and completed during the sprint are highlighted in blue.

At the start of the spring semester, we formed a very rough timeline, splitting the semester up into seven sprints and writing single-line goals to explain what we wanted to accomplish during each sprint. The second Thursday during the second (and final) week of each sprint is a dedicated sprint planning session during which we review the tasks that we completed, note the ones that we did not complete, and revise our plans for the next sprint. Our tasks were defined generally, with typically just the title and a few labels to describe the type of work that needed to be done.

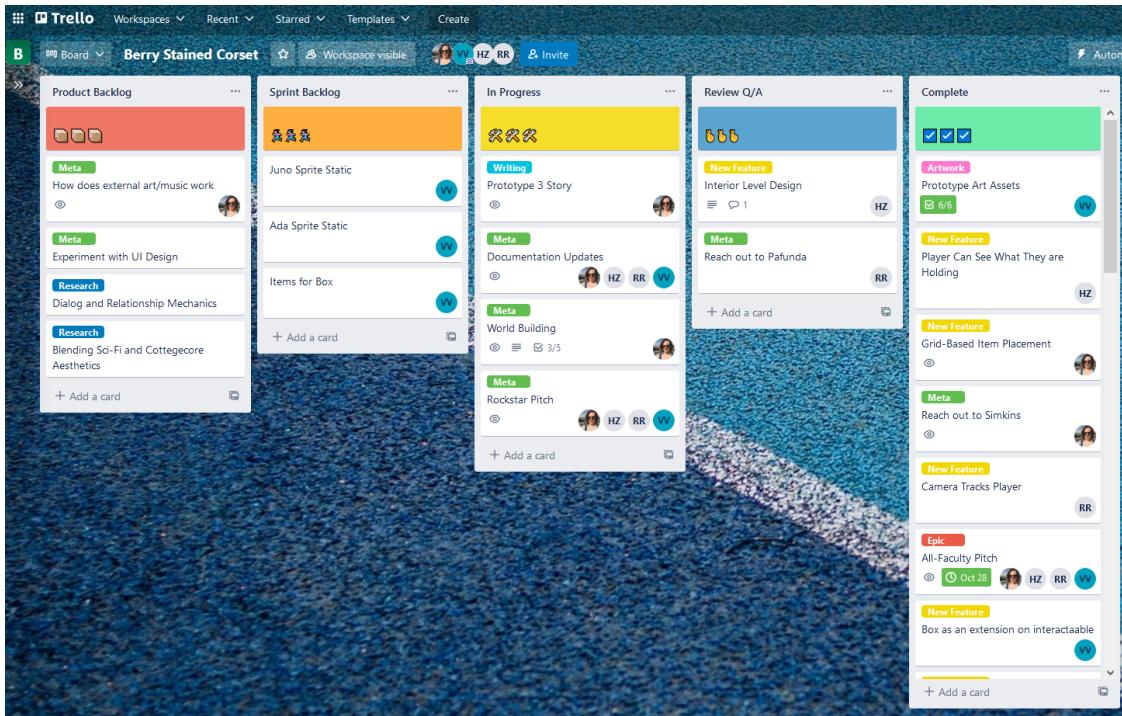


Figure 01: The team's Trello board at a glance.

Organizing Tasks and Tracking Progress

We used Trello to organize tasks and track their progress. It is primarily used by our project manager to know at-a-glance the status of tasks. The most important questions that it answers are “have you *started* working on this?” and “are you *finished* working on this?”. These questions are answered by the cards position in the five lists our board contains, and lets the project manager know when she should review team member work. Our board was broken up into the following lists.

Product Backlog

New tasks that the team *could* work on, but priority hasn't been determined yet and the team has yet to commit to the work in the current sprint.

Sprint Backlog

Tasks get added to the sprint backlog during each planning meeting. These are tasks that the team has committed to, but that are not actively being worked on.

In Progress

These tasks are in-progress; a team member is actively working on them.

Review/QA/Merge

These are tasks that team members have finished and that need to be approved and merged into the project by the project manager.

Complete

These tasks have been reviewed by the project manager and have been merged into the project. Thus, they are completed!

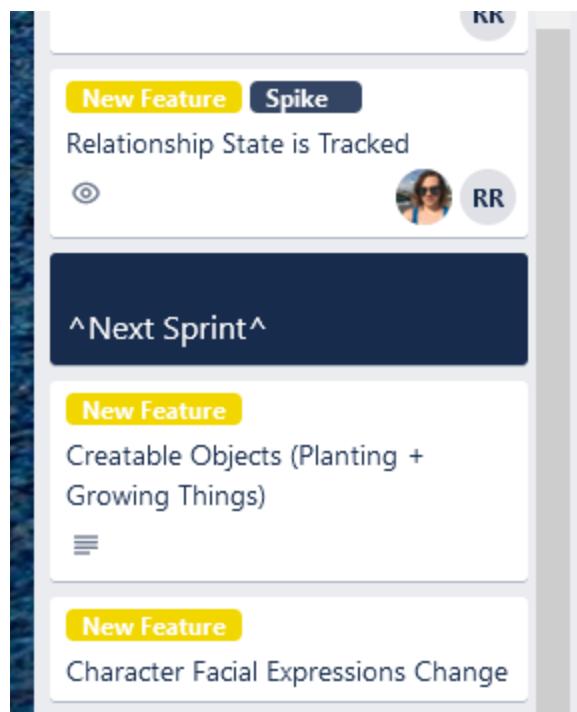


Figure 02: A card is used like a divider to differentiate commitment to different tasks.

Task Creation

Tasks may be added to the sprint backlog as reminders or notes that there is something we care about doing, but not all of these tasks would be added to the next sprint's work. When our team planned each sprint, we were able to revisit tasks that we decided to put off until a later time and reconsider their priority. This was an organic and useful means of weeding-out lower priority tasks without completely ignoring potential work we might need to do.

The screenshot shows a task card titled "UI Design for the Game" in the "In Progress" list. The card has sections for "Members" (with a user icon and a plus sign), "Labels" (with a "Spike" label and a plus sign), and "Description" (with a placeholder "Add a more detailed description..."). Below the card is a comment section:

- Emily Horton** Jan 21 at 1:06 PM: I will be finishing this during sprint 2.
Edit - Delete
- Emily Horton** Jan 21 at 1:06 PM: Memories: <https://docs.google.com/document/d/1asamW4M2vTDqrKhJYuBljwUda61L2QMLa43cuyqu77l/edit>
Edit - Add link as attachment - Delete
- Emily Horton** Jan 13 at 4:47 PM: Romantic Tropes and Expressions in Media: https://docs.google.com/document/d/1Sn3sW8C7sJyC7bphyNdyaqoCkgwrYLL_4uvkf_nkUY/edit
Edit - Add link as attachment - Delete

Figure 03: Cards are devoid of user stories and acceptance criteria/definitions of done. If there are relevant documents or details that need to be added to the card for clarity, we use the comments section.

Tasks are created and defined with relatively low granularity. Because our team is small and meeting very regularly throughout the week, we discuss both implementation details and what a task *means* with one another directly. The main purpose of our task cards (and the board as a whole) is to track progress on tasks rather than to define the tasks.

Conclusions

In some ways, our implementation and interpretation of agile are more akin to development “from the hip”. Though there are definitely drawbacks to this method compared to a higher-ceremony approach, our team never felt the need for additional ceremony. Though we will save the details for our postmortem, it might have been better if we had tried a higher-ceremony approach because we suffered from a lack of process discipline as Capstone Development progressed and our

deadlines became tighter. Taking a casual approach to task definition (and production in general) resulted in miscommunications and trouble merging everyone's individual contributions together.

4.5 Communication

4.5.1 Internal Team Communication

Discussion happened either in person during meeting times or asynchronously through the class discord server in our channel. For Fall semester, we met every Tuesday and Thursday between 1530 and 1700 for group work and to hold standups on Monday's, Wednesday's, and Friday's. For the spring semester we met during class times (Tuesday and Thursday from 1100 to 1330) and on Thursdays from 1500 - 1700. All of our meetings began with an informal standup and check in to assess team member personal well being and capacity to work. Though the bulk of our communication happened synchronously during these meeting times, we also used our team's Discord channel to communicate with one another on task updates or to ask questions. Towards the end of the semester we added a Monday planning meeting from 1100 to 1200 to prioritize and assign tasks for that week.

All of our synchronous communication is hybrid. This allows for more flexibility for our team members who cannot always make it to campus. One unique communication consideration was that those of us working in the labs often wrote a lot of information on the whiteboard for our team. We would always take the initiative to share it with anyone working remotely. We would take a picture of whatever was written and share it in Discord.

4.5.2 Stakeholder Communication

Here we will identify the stakeholders of our project. Because each stakeholder is in a unique position at the university, we needed to adopt a variety of meeting times, formats, and communication styles.

Capstone Advisors

Elouise Oyzon, Erika Mesh

We met informally with our capstone advisors during each capstone class and gave updates as to our progress. Professor Oyzon has also been helpful in providing feedback for art assets. Professor Mesh has helped with any process questions. Both professors have assisted our class in organizing playtests and in setting deadlines for our games' progress. Conversations with each of them tended to be very casual and fostered a safe, candid learning environment for ideas and feedback to flow freely.

Committee

Eric Baker

Because of Professor Baker's packed schedule and its inconsistency due to having small children, we never settled on a synchronous meeting time with him. Instead, our team would message him via Discord, tag him in comments on our project documentation, or visit his office directly when we wanted to ask him for input or update him on our project's status.

David Simkins, PhD

Our team met in-person with Dr. David Simkins on every Thursday at 1230. Discussions with Simkins tended to be long, high-level discussions about game design, narrative structure, and player immersion through the context of roleplaying. Our talks became much more candid over time, as Simkins was always more than willing to share his personal perspective on relationships and the issues that we should consider exploring.

Robert Glick, PhD

Emily (and occasionally, Veronica) met with Dr. Robert Glick over Zoom on Monday afternoons. Emily remained the primary point of contact with Robert because she was the game's writer. As the year progressed, it became much more common for these meetings to be rescheduled. Both Emily and Robert frequently emailed one another in order to be flexible and work with each other's increasingly limited schedule

4.6 Planning & Scope

4.6.1 Overall Priorities

The priorities of *Mending Hearts* have dramatically evolved throughout its development. This is a result of our team spending most of our time in Capstone Development trying to understand what the game *wanted* to be. By the time we finished that first semester, we defined our priorities around the game's main character's *relationship*. Although there is certainly room for innovation within the domain of home decor and life simulator games, the relationship between Ada and Juno *as the game* is both the most unique aspect of the project and the core of the experience. Therefore, our number one priority is to *sell* the relationship of these characters to the player and get them *invested* in helping to determine the outcome.

4.6.2 Minimum Viable Product

Determining the minimum viable product for *Mending Hearts* was difficult because we couldn't anticipate what *gameplay* we needed the player to do in order to evoke the intended emotions. Individual research exploring the emotional experience of relationships and researching the life simulator, dating simulator, and visual novel genres helped us determine the direction we were headed in.

For MVP of *Mending Hearts*, we settled on completing one vignette of the relationship. This complete loop would need to demonstrate that the player's interactions with Juno would change their relationship in a meaningful way. The actual gameplay would involve Ada being able to talk to Juno, Juno being able to decide to talk to Ada, and each of them would have a slice-of-life goal to accomplish. To accomplish these goals, we created a largely on-rails Prologue vignette in order to give the player the context of the relationship. Then, we followed with the Spring vignette, in which Ada and Juno must work together to cook a meal before setting up their respective work spaces in their home. Within this vignette, we would need to allow Ada to talk to Juno at-will, and Juno would respond to Ada based on her current feelings about the relationship. This MVP would showcase every important aspect of *Mending Hearts*: an NPC with feelings and a relationship impression with the player, cozy home-living gameplay, and

long-term tracking of the relationship's progress as it changes between each vignette.

4.7 Timeline

4.7.1 Initial Concept & Research — August 2021

Initially, the idea was you were repairing your house with your girlfriend who just got back from war. The core gameplay loop was talking to your girlfriend to figure out what furniture you should make for your house. Then, you would create a furniture plan, craft every piece of the furniture, and put it all together, presenting it to your girlfriend as the new addition to your house. Throughout these pieces of furniture, the player would have story beats that are conversations with their girlfriend about how their relationship is going after reconnecting.

Our initial research took us about the first month of our project. The goal of this stage was mostly just to get an idea of what other games in our genre looked like and what the overall feel of the game would be. At this point, we also thought that we would have one of the characters be in the military so we did research into that and its effect on relationships.

This stage of development went relatively smoothly. It was a lot about seeing what was out there and what would work for our game. This was also the first time that we were all working together. Even though we have worked together on past projects and in past classes, we had never all explicitly worked together at the same time.

4.7.2 First Prototype (Paper Furniture Crafting) — September 2021

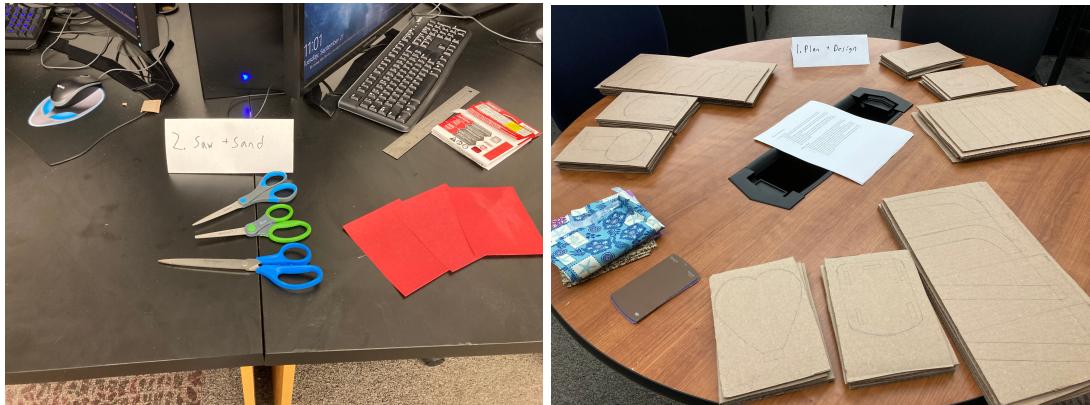


Figure 04 and 05: Materials for Paper Prototype

Shown here is our first prototype: our Paper Prototype. With this, we wanted to test both the process we had outlined for creating furniture as well as how the player would interact with their partner.

Many aspects of this were a success! The players loved physically building and putting together this furniture and they really enjoyed trying to please their partner and create the perfect furniture.

Some aspects however, were a little rougher. The planning for this had some holes. We only had an in-house test two days prior to the open test, which made us realize we had to pivot in a few ways to allow this to succeed.

We really did learn a lot from this playtest, though. We learned that we are on the right path with the interaction with Juno, but there needs to be some balancing between interests. We also learned that the outline for making furniture worked pretty well.

4.7.3 Second Prototype (Digital Furniture Crafting) — October 2021

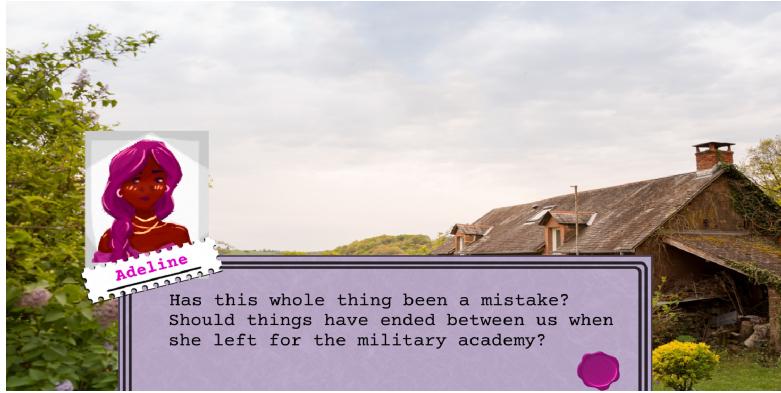


Figure 06 and 07: Narrative UI From First Digital Prototype and Furniture Tracing Minigame

Quickly came our second playtest: our First Digital Prototype.

With this prototype and playtest, we wanted to take what we had on paper and bring it into the virtual world.

This playtest, from our perspective, was a little rough. That being said, there were some successes. For one, players got a good sense of the direction we planned on taking the game in. They also loved the interactions between Ada and Juno and the awesome art we had in.

The issues with this prototype again start with planning. We did not get all the features that we wanted into the game. And we took the morning of the playtest to attach all of our modular parts, which resulted in another chunk being left off. We learned a lot from this playtest. We found out the hard truth about code freezes and the issues that come if they aren't implemented. More importantly, however, was that furniture crafting was not the direction we wanted to take *Mending Hearts* in. Although we as a team had made the connections between restoring old furniture and restoring a relationship, our playtesters did not

understand what the two portions of the game had to do with one another, and they consistently reported that it felt like two different games were happening. We decided to go back to the proverbial drawing board and try a different set of mechanics.

4.7.4 All Faculty Pitch — October 2021

For our all faculty pitch, we were starting to find this new direction for our game. We had a loose concept of where we wanted to take our game. Namely, we knew what we wanted to accomplish and what we wanted to stay away from. We knew we wanted to focus more on the relationship between Ada and Juno than dive too deep into one mechanic or process of building a house. We knew that we should shift Juno away from a former military member so that we can avoid the minefield (if you'll excuse the poor metaphor) of trauma that comes with it. This would take away from what the core of our game was and would take us in directions we didn't want to go. We also knew that we should allow the player to move around in world space and interact with different aspects of their home and with Juno.

Speaking of Juno, we realized that she cannot simply be an NPC who doesn't move and only waits for you to talk to her. She needs to be her own person who acts on her own with her own wants and needs. Based on these wants and needs, she should be able to also move in world space and interact with the house and Ada. We thought that the main conflicts in the relationship would come up in disagreements about what needed to be done around the house and what Juno asked Ada to do and what ended up getting done.

4.7.5 Rockstar Pitch + Documentation Update — November 2021

By the time the Rockstar pitch came around, we had narrowed down our game's new direction and had started to work to make it a reality. We started to get our world built and began the tough task of starting a game from scratch for the second time.

However, we didn't have much to show for this quite yet. So, we used this as an opportunity to improve our all faculty pitch and get some more feedback on the new concept for our game's direction.

4.7.6 Third Prototype with New Systems — November 2021



Figure 08: Grid Based Implementation of *Mending Hearts*

After our lukewarm attempt at creating a digital furniture crafting process, we decided to move in a new direction with *Mending Hearts*.

The goals with this playtest are similar to many first playtests and tech demos; we wanted to showcase our new direction for the game and see if it's worth pursuing further. This includes testing mechanics, interactions, and the overall new look and feel of the game.

From our perspective, this playtest went a lot better than our previous one. Better planning meant that just about everything we wanted in the prototype was there! We took really good notes and our testers were great at playing Twitch Streamer for the day.

This playtest did have its issues however. The main one looking back is the lack of context for the player that would have taken our tech demo to the next level. We also found a lot of mechanical issues to sort out for the future. With these, we found what aspects of our new direction serve our goals well (simple mechanics and focusing on the characters) and what might need to be adjusted slightly (character interactions feel a little more like testing input than talking to someone).

4.7.7 Capstone Development Kickoff — January 2022

We began the semester by evaluating where we were in the game and what we wanted to get out of the game. During our fall semester we spent a lot of time refining the direction and design of the game. In the end we came out with a clear idea of where we wanted to go, but this meant that there was a ton of development work that needed to be done. In order to make sure we stayed on track we broke the semester out into eight (8) two (2) week sprints. We planned out the first four (4) with great detail then left the last few with just some vague goal posts so that we had flexibility as the semester progressed.

Our first sprint was mainly focused on game feel. Sure we have our main interactions in the game, but do they feel good to interact with? This sprint we worked on making it clear to the player which items they were going to pick up as well as giving an introductory scene so the player has context for the game and a small tutorial.

4.7.8 Spring Sprint 2: First Spring Playtest — February 2022

Sprint 2 focused on individual research, prepping for our first playtest of the semester, and some quality of life changes for our game.

For this playtest we wanted to make sure that the improvements to the game were actually making the game more clear, and overall they were. There were still a couple minor issues which we quickly squashed after the playtest. We revamped the grid system so that every item in the home directly interacts with it. This improves consistency in item placement that highlighting items alone could not. We also added tooltip hovers for interactions so players are not guessing how to interact with the world.

Research wise, we have started outlining Juno and her AI, fleshed out some story beats and memories, and began outlining how home decor will be rated.

4.7.9 GDC Build — March 2022

The GDC Build saw a new (and hopefully final) direction for the game. We have removed the grid system from the game and moved away from home decorating, to interacting with the home and your girlfriend. Our story will now take place in four vignettes, representing the seasons of Juno and Ada's first year together.

The outcome of the relationship at the end of one vignette will affect the state of the relationship at the start of the next vignette. We have continued to work on our individual research topics.

This build was a small prototype of the game focused around the prologue. It showcased the memory mechanic, interactions, and changing the state of the game between vignettes. This was also our first build with sound in it.

4.7.10 Final Deliverable — April 2022

The final state of the game is a prototype that has all of the core systems in the build, but they are not all fully represented in the game's content. We were successful in implementing relationship tracking, context-sensitive dialogue responses from Juno, Juno's pathfinding and goal-picking AI, and simple gameplay involving common household tasks, specifically cooking and unpacking furniture. Our final deliverable communicates the context of the protagonists' relationship and introduces each character's unique personality. Finally, we were also able to add music and sound effects to further immerse the player.

In some ways, we were successful in meeting our deliverable, but the prototype lacks significant polish that detracts from the player experience. For example, our UI that indicates interactivity with world objects isn't context sensitive and can mislead the player about what actions they can currently perform. Additionally, Juno's autonomy is implemented, but it is used in one very small and limited context and does not demonstrate the range of tasks she is able to do.

4.7.11 Stretch Goals — The Future?

We wanted to add a scrapbook feature that could collect the player's memories (and other world objects) in one place for review at a later time. This feature would give the player a sense of progression and a means to revisit scenes they had already read in case they had missed something previously.

In order to enhance the story experience, we wanted to create a significant amount of background drawings to show during cutscenes. This way, we would give the player not only a sense of place, but also a sense of the emotions between each character. Though our dialogue can go a long way in conveying

emotions, a fullscreen piece of art showing each character's expression and body language would help the dialogue stick the landing with our players.

Finally, we wanted to implement a second vignette of the story into the game, the Summer vignette. Emily had already written the story for the summer vignette, and Hanrui created levels for this portion of the story, but due to time constraints we were not able to include it. Unfortunately, this content cut severely diminished the impact of *Mending Hearts*' story because the Summer vignette was written to be the first major tension point between the protagonist, and the events that unfolded would be drastically different based on the relationship status from the previous vignette.

4.8 Version Control

4.8.1 Source Code

Due to both familiarity and accessibility, Berry-Stained Corset is using GitHub to manage our code base. Our repository is using a simple branching system in order to avoid common issues with merge conflicts and breaking a build with no means to revert to a working version.

Tags - Creating a Release

At the end of each sprint, we create a new Tag in order to demarcate work that has been accomplished in each sprint. It is both a useful means to revisit project history (ie. compare how the product has evolved) and to have a backup we can revert to in case a sprint goes horribly awry.

Branches - Main

The main branch is not a branch that we ever work off of directly. This branch is only updated at the end of a sprint when all of our work is merged into main. This merging represents our commitment to the work that we have completed; we are affirming that the work has been approved by the team and is working as we expect it to.

Branches - Development

The development branch is updated frequently during a sprint. Similar to the main branch, the team does not work directly on the development branch. Instead, team members create branches based on the development branch. The development branch gets merged into the main branch at the end of a sprint.

Branches - Feature

Team members working on changes to the code base work off of feature branches, which are based on development. Each feature branch is to be named based on the specific task that it is addressing. When a team member has finished their work, they will create a new pull request in order to merge their feature branch into development.

The Merge Process

It is the responsibility of the project manager to approve or deny merge requests. Tasks that are ready to be merged will have been moved into the Review/QA list in the Trello board. The project manager will know that there is a pull request ready to be reviewed. Pull requests will be merged if there are no unresolvable conflicts and no code changes that appear to break things. The project manager is responsible for resolving simple merge conflicts that can arise as developers are working during a sprint and merging code into the development branch– more often than not, the conflict can be fixed directly in GitHub's web editor.

Developer Unity Scenes - Divide and Conquer

Unfortunately, merging Unity scenes using Git rarely ever works because the files are generated by Unity and result in large merge conflicts. Attempting to do so by hand inevitably results in a corrupted scene file that Unity can no longer read. Therefore, each developer works within their own test scene, which is a copy of the game's main scene. By the end of the sprint, all changes that must be made to the main game scene that were done in everyone's test scenes will be done by hand by the project manager. This ensures that functionality added by developers can always be added into the project during a sprint and that development can continue without dealing with conflicting or potentially corrupted files.

4.8.2 Project Artifacts

GDD Capstone > Capstone Teams 2021-22 > Berry-Stained Corset		
Name	Last modified	File size
Art Assets	Oct 5, 2021 Veronica Vitale	—
Brainstorming	Aug 24, 2021 me	—
Builds	Dec 4, 2021 Veronica Vitale	—
Communication Drafts	Jan 6, 2022 me	—
Level Design	Oct 28, 2021 Hanrui Zhang	—
Misc. Meta Documents	Nov 8, 2021 me	—
Moodboards	Aug 24, 2021 me	—
Notes	Sep 28, 2021 Veronica Vitale	—
Playtest	Nov 8, 2021 me	—
Presentations	Sep 2, 2021 me	—
Research	Aug 31, 2021 Ryan Ress	—
Writing	Oct 13, 2021 me	—
Berry-Stained Corset -- Mending Hearts	8:43 PM me	—
RelationshipDataAndLines	Mar 31, 2022 me	—

Figure 09: Google Drive Organization

We used the capstone Google Drive that Weez setup for all of the teams to store our project's artifacts, which includes documentation, meeting notes, artwork, writing documents (including planning documents and actual dialogue scripts) and our presentations. Using the RIT Google Drive was the simplest way to ensure that Weez, Erika, our committee members, and our fellow graduate students can easily access our materials for review. Similar to the team's low-ceremony approach to process, our drive has a loose organization system that we have simply added folders to as it made sense to us at the time. Although we could have invested more time developing a more formal organization system, doing so didn't make sense to us given the scope of our project and the size of our team.

4.9 Asset Pipeline

Item	Priority	Quantity	Specifics	Status	Comments from Review
Juno Sprites	High		12 this should make for a full walking animation in all directions	Done	IMO animation is very low priority. Just get each character facing each way and we can do animation later as polish.
Ada Sprites	High		12 this should make for a full walking animation in all directions	Done	
Ada Emotes	High		>3 At least one for happy, sad, and mad	Done	
Juno Emotes	High		>3 At least one for happy, sad, and mad	Done	I'd try keeping them as neutral for each emotion as possible so that they can be versatile; in the future we will definitely have r
Kitchen Chair	High		4 for each direction	Done	
Bed	High	1		Done	
Lamp	High	1		Done	
Interior Tileset	Medium		Tileset for the interior that allows Hanui to build us a beautiful home. Will need to be scc Done		
Juno's Teddy Bear	Medium	1		Done	
Counter	Medium	1		Done	Divider lines on the surface of the counter seem a little weird. This also doesn't let us have a counter of varying length (ex. tw
Oven	Medium	1		Done	
Ada Portrait	Super High		1 figure out how to make ada ada :)	Done	
Box With Items	Super High	2		Done	
Sink at various levels of dirty	Low		>3 Empty, Filling, Full	Up For Review	Could use some cleanup: lots of random pixels at the top of the sprite (at least on the empty sprite, maybe all of them)
Coffee Table	High	1		Up For Review	
Fridge	High	1		Up For Review	
Desk Chair	Low	1		Up For Review	
Travel Brochure	Medium		1 Something about space travel. A brochure Ada picked up when Juno left at the space po	Up For Review	
Ada's Art Space	Super High		2+? An easel where she paints. Maybe she also has an organizer for art supplies?	Up For Review	
Juno's Science Space	Super High		3+? A desk for science stuff. Work stool. Maybe a laptop.	Up For Review	
Kitchen Table	High	1		Up For Review	
Couch	High	1		Up For Review	
Juno's Bags	Super High		1 Juno's suitcases/bags that she brought home with her	Up For Review	
Bedroom Scenes	High		5 An overhead(?) view of Ada and Juno in bed together. There should be several different pos	In Progress	
Dresser Drawer	Medium		1 For keeping clothes in. This probably Juno's.	In Progress	
Armoire	Medium		1 Ada is definitely That Bitch who would own and use one of these for her fancy clothes	In Progress	
Bookshelf	Medium	1			
Scrap Book UI Mock UP	Super Low		1 This will allow us to figure out what assets are needed for the scrap book		
Flowers	Super Low		>5 * 3 Need various types and stages, seedling -> bud -> flower		Might also need to show that they have been watered, if watering is a mechanic we will be doing.
Dirt Animation/Partical	Super Low		1 Should be scalable to any item		
Broken Down Box	Super Low	1			

Figure 10: Art Backlog

We used a single spreadsheet to organize the production of our art assets. Whenever a task, team meeting, or piece of feedback caused a need for a new art asset, we would add a new row into this spreadsheet, which serves as both an asset backlog and a progress-tracking tool. We adopted this approach due to the way our artist, Veronica, worked on art. Typically, they would want to sit down with their drawing tablet and simply focus on drawing in between other tasks they had to complete. Creating a backlog like this gave them control over organizing production and kept the Trello board less cluttered.

Once Veronica began working on an asset, they would denote that it was in progress by labeling it as such in the status column. Upon completion, the asset will then be marked as “Up for Review,” uploaded to the drive, and team members will be notified in discord that there are new assets to look at. Any changes necessary will be made. Once all team members have agreed the asset is ready to go into the game, it is marked as “Done.” Done assets may be put into the game as team members need them.

5 Game Design

5.1 Overview

We wanted to create a cozy, sentimental, and uplifting experience with a cottage core theme. This meant that our game should be able to be played in an afternoon, curled up in a blanket with a cup of hot chocolate. The player should feel connected to and care about the characters and their stories. Lastly, we knew that there was a potential for the player to “fail” the relationship, but we wanted to make sure that even if that happened the game would still take a positive turn on this outcome.

However, we did not know what the *game* would be. We knew the narrative and we knew the experience we wanted to deliver, but *how* we would deliver and specifically what the player would *do* was unclear to us. Our initial idea was furniture crafting and restoration because we thought that the act of restoring a piece of furniture was symbolic or restoring the relationship between Ada and Juno, and that working on antique furniture was in-line with the cotagecore aesthetic.

This direction did not pan out; though players seemed eager to create furniture (especially in the context of doing so to please their girlfriend), they did not understand how the narrative connected to this gameplay. Rather than try to alter the context or “force” the idea, we instead pivoted to home decorating. The idea of decorating a house and turning it into a home as a collaborative process seemed like a logical step that aligned with our design goals and aesthetics. Additionally, our committee members loved the idea because navigating moving-in and negotiating how to decorate the shared living space was something that rang true to their experiences with marriage.

However, we once again had to change our idea, this time due to scope. Our emphasis on home decor required a lot of assets, but we only had one artist. Additionally, one of our design goals was to allow Juno to intelligently navigate the home and interact with the objects. The grid system we were using to organize where items get placed in the home and how they all interact with each other was too complicated to manage within our time and people constraints. Therefore, we settled on reducing the scope to just interacting with world objects, which would allow us to implement more simple navigation and focus on the interactions between Ada and Juno— which serve our core experience.

5.2 Player Objectives

Players will control Ada, one of the characters in the relationship. Ada and Juno want to get their feelings about each other back and bring their relationship to the next level in their new house. Since we believe that there is no “winner” or “loser” in a relationship, the win condition for *Mending Hearts* would be players exploring the whole story.

The only NPC present is the player’s partner, Juno. Interactions with Juno will be mainly daily conversations between Ada and Juno, but will also include small cutscenes following the two on dates outside the house. However, if players try to interact with Juno too much to even bother her, there will be some dialogues indicating that Juno is getting upset and similar things would happen if players ignore Juno for a very long time. Then, something like the cutscene in the end of the vignette will change to indicate that.

5.3 Emotional and Narrative Core

The emotional core of *Mending Hearts* is the relationship between Ada and Juno, and the player working as Ada to rediscover with Juno what made them such a strong couple in the first place.

Ada is a painter who specializes in landscape painting. She is an artsy woman who sees the world’s beauty through a vibrant and poetic lens. After college, she

had plans to move into a cabin she inherited from a relative, where she could continue her commissions.

Juno is a botanist and recent graduate who loves spending her time outdoors in the natural world. To her, the universe is an intricate and complex system of elegant machinery governed by the laws of physics and math. It is as beautiful as it is mysterious.

Through a chance encounter in college, these two women who are opposites in their fields of study, their world views, and even their color schemes, formed an immediate bond that grew deeper every moment they spent together. Their differences seemed only to complement each other, and they each found the other to be a fascinating and fulfilling source of passion.

Juno was thrilled at the prospect of moving in with Ada in her cabin after college, getting far away from her suffocating family, being surrounded by beautiful natural vistas, and living with a woman she so deeply loves— but all of that would have to wait because Juno was given a now-or-never opportunity to establish herself within her field fresh out of college.

“What’s two years in the face of the rest of our lives?” they asked together, when they last said their tearful goodbyes.

Mending Hearts picks up when Juno returns from her scientific expedition to Ada’s cottage. They are now navigating the intricacies of trying to bring their relationship back to what it was before Juno’ trip mixed tightly with the intricacies of moving into a home with each other for the first time.

Here is where I will link to my writing. Maybe I could also show selected excerpts or a screenshot to show how I formatted things/why?

5.4 Gameplay

Mending Relationships is a single player narrative and life-sim game. The game play is broken up into several vignettes representing different seasons. Each vignette has a set of goals to complete. The player’s performance in the goals as

well as how they interact with their girl friend will determine the outcome of the vignette and affect the next vignette.

Much of the story of the game is given through visual novel-esque dialog scenes. These teach the player about the characters and their struggles. As well as hint at the main goals for each vignette.

The player is able to move through and interact with the home. The main interaction we worked on was cooking. The player can add ingredients to the stove from the fridge following a recipe. Once all of the ingredients are added the player can cook the dish and then place it on the table for eating.

The player can also interact with their girlfriend, bringing up a menu to determine what type of conversation they will have. Interacting with the girlfriend too much or too little will tank the player's relationship. So the player must take care to not pester their girlfriend every second but also make sure they do not forget about her completely.

5.5 Aesthetics

We want the aesthetics of our game to spark a sense of joy in the player. Because of this goal, we identified three keywords that would help guide our decision-making for the aesthetics of *Mending Hearts*: Cozy, Sentimental, and Uplifting.



Figure 11: An early moodboard used to establish the colors and overall vibe of Mending Hearts.

Games that are largely considered casual games typically have very simple artwork that looks cute and colorful, but a wide array of emotions and themes can be explored within those parameters. *Mortician's Tale* uses a lot of purples and cool colors. *Animal Crossing* uses a very wide variety of colors, allowing the player to create their own schemes and themes. *Stardew Valley* uses different color palettes to depict when seasons change. For our game, the palette has a range of earthy tones to fit in with the rural living as well as pastels to give a cutesy vibe. We decided to saturate the colors to give the game vibrance.



Figure 12: Initial color Palette of *Mending Hearts*

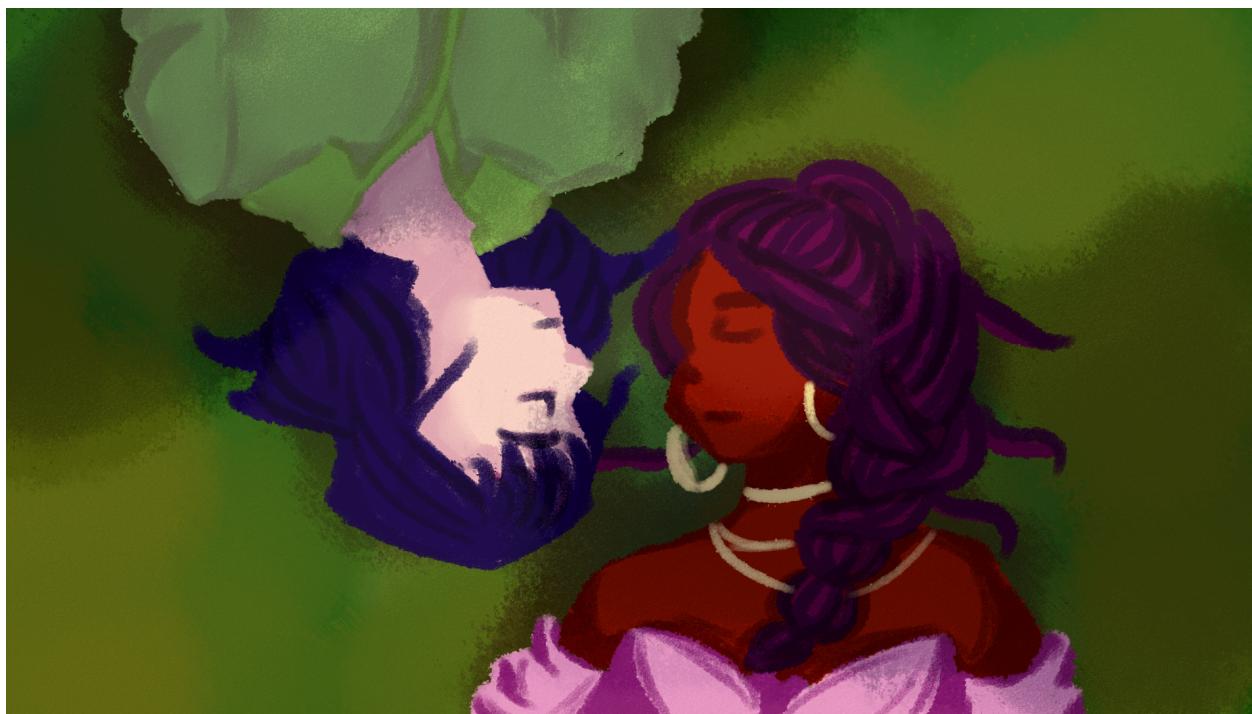


Figure 13: Early Concept Art

We designed *Mending Hearts* around the cottagecore aesthetic, which focuses on living a simple, enjoyable life in the face of capitalism. Oftentimes it takes influences from the past for the aesthetics but with modern values.² Because of

² “Cottagecore.” Aesthetics Wiki. Accessed May 2, 2022. <https://aesthetics.fandom.com/wiki/Cottagecore>.

this combination of aesthetic sensibilities, Veronica considered several historic painting styles and decided to imitate art from the impressionist movement.



3

Figure 14: Impressionism focuses on accurately capturing light to get a specific moment in time and uses short strokes with unblended colors as well as colorful shadows.

³ “Impressionism.” Wikipedia. Wikimedia Foundation, April 20, 2022. <https://en.wikipedia.org/wiki/Impressionism>.

5.6 User Interface

Though there is only one true game mode, the gameplay has two distinct UI states: conversations between Ada and Juno and moving around in and interacting with the game world.



Figure 15: Visual Novel UI for telling the story

Our dialogue is presented to the player in a manner typical of visual novels. Figure 15 shows the final version of our design. A box appears for dialogue to be printed to the screen. The full text is added letter-by-letter, a style that is commonly referred to as typewriter-text. In the upper left corner you can see a portrait of the speaker– in this case, Ada. In the bottom right corner is a button the player can click to advance the dialogue– they may also press space instead.

We designed our dialogue boxes to look like a letter written on parchment paper. The advance button is a wax seal that stands out, indicating the end of the dialogue. The character portraits are bordered and vignetted to give the appearance of a photograph, and the nameplate has a similar pattern-cut edge to that of a stamp.



Figure 16: Options for interacting with Juno

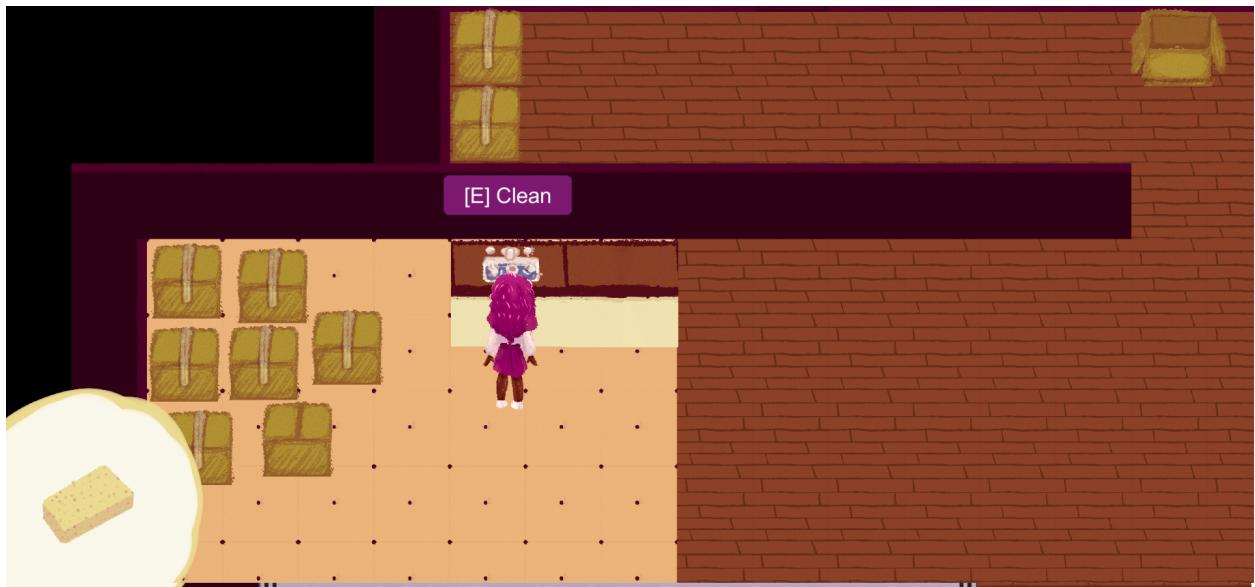


Figure 17: UI pop-up for interacting with items and bottom right corner showing the held item

In the world, the player will be able to see what item they are holding, and UI popups will show the player what they can interact with as well as what interacting with said object will do. Some interactions will open up menus, such as cooking and the fridge. These are all navigable with the keyboard so that the player does

not have to switch between mouse and keyboard. All interactions and menus tell the player what does what. This is to hopefully lead to less confusion.

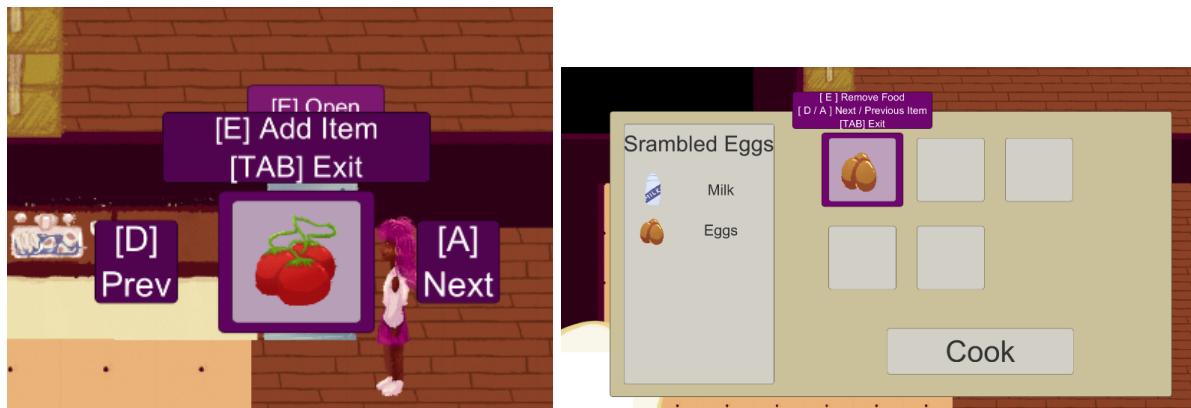


Figure 18 and 19: UI for Cooking and the Fridge, both showing navigation instructions

6 Assets

For information on general aesthetic decisions and why we made them, [see above section 5.5.](#)

Character Design

Ada and Juno as characters have complimentary personalities (i.e. they have different characteristics but fit well together). In order to show this in their design their main colors are complimentary (Ada with pink to show off her femininity and Juno with green to compliment as well as show her interest in plants).



Figure XX and XX: Ada's portraits and overworld sprites.

Ada's design is very feminine, from the use of pink as her main color to the outfit and accessories she wears. Her outfit is based off of Veronica's favorite outfit (and they see themselves as a very feminine and fashionable person so they thought it was fitting). The outfit itself is very simplistic and comfortable reflecting on the cottagecore theme, while also appearing very stylish and put together to show off Ada's personality.



Figure XX and XX: Juno's portraits and overworld sprites.

Juno's design is meant to be laid-back and simple to contrast Ada's feminine and stylish look. Though, it still has its own flair to be on the level of formality of Ada's. Juno's look is inspired a bit by 50's greaser style and more recent alternative styles such as goth and punk. Her hair is seen pulled back since she is used to keeping her hair pulled back when doing her research. Yellow streaks are used in Juno's hair to add more value and contrast in her look since without it, her outfit is completely green.

The Home - Objects



Figure 22 and 23 and 24: Fridge, Ada's Easel, and Juno's Desk

For the home design, the items have a painted and hand-drawn feel to them. This is to add to the cozy feel of the game. The items avoid using bright pinks and greens unless they are specifically meant to be Ada's or Juno's, respectively. This

way the home can feel like it belongs to both characters and not just one. Items that people would normally add a personal touch to, such as fridge magnets that are meant to be shared take equally from Ada and Juno's interests in the embellishments.

The Home - The Tileset

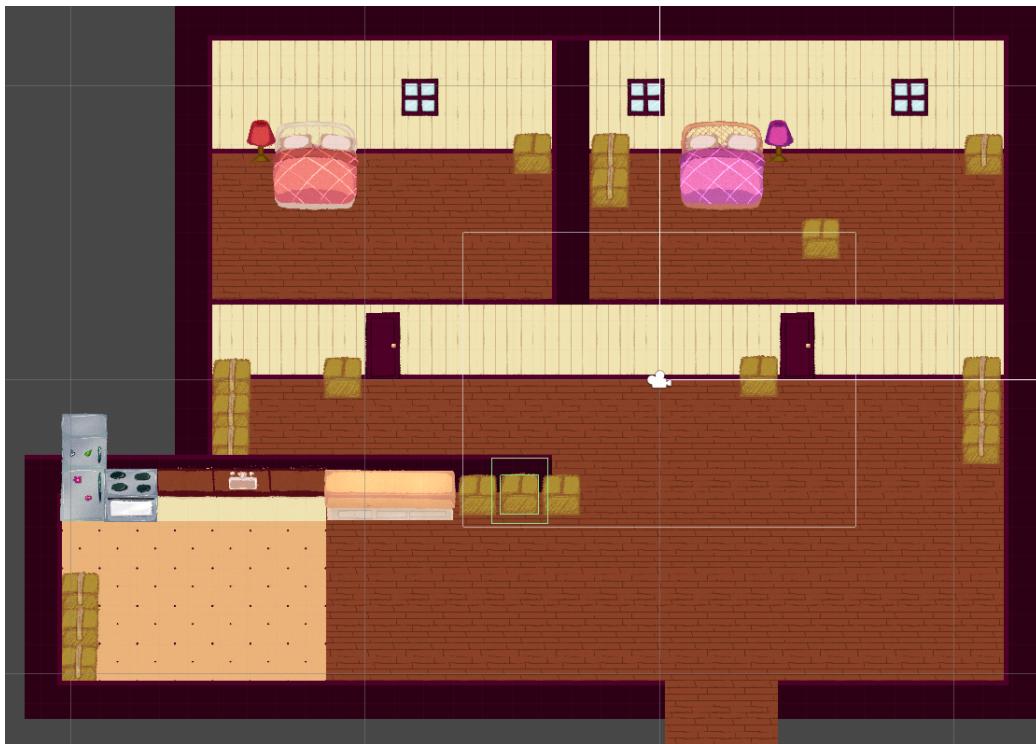


Figure 25: Interior layout using artist-created tileset

For interior design, putting the main entrance to the bottom of the indoor level could make it feel more real when players get into the house from outdoor level than entering from a door on the top of the indoor level. Most of the openable boxes are put in plain sight so that players can notice them easily and the boxes that can not be opened at that time are put at corners of the house to make sure that they will not affect players' experience.

7 Technical Design

7.1 Summary

The major systems in *Mending Hearts* were created largely independently of one another. Although there is an advantage in scripting behaviors and creating prefabs in isolation, our team ended up struggling to combine everyone's work; we failed to pre-plan the interfaces between the different sections of the game's functionality. That said, each individual component, such as the player controller, the story manager, or the Juno movement controller, function fairly well on their own. Where they need to interface, it is common for the scripts to simply find the object in the scene by name or by object type.

We did not envision *Mending Hearts* as a technically challenging project, and we acknowledged partway through development that our team was stretched thin; the strongest developers on the team were also responsible for other major tasks such as asset production or writing. This is why our overall approach was to create the game "in pieces"; get something working as fast as possible so that another team member can use it and you can return to your other priorities. As we will discuss in our postmortem, this resulted in technical debt that our team was later forced to repay as our pieces needed to come together as a cohesive whole.

7.2 The Player

The player character Ada is a simple prefab with a sprite renderer and a rigidbody used to show her in the world and ensure she physically interacts with it as expected. Her behavior script contains a fixed-size array of sprites to represent each of her possible facing directions. Her movement is controlled by the keyboard using Unity's old Input system. Additionally, she is able to interact with objects when the player presses the E key. If she is within the bounds of an interactable object's trigger hit box, she will call the Interact method of that interactable object.

7.3 Interacting with the World

All interactable items in the game inherit from the ItemBase class. This class provides the code for checking if the player is at an interactable distance of the item and the method the player will call to interact with the items.

By providing the code for seeing if an item can be interacted with, this puts less work on the developer for future items. When creating a new item, the developer only needs to assign the correct trigger to the script and adjust it to the item.

All children override the interact method for functionality. This makes it so all interactions are uniform and the player class can call an item's interact method without having to know precisely what type of item it is.

7.4 Creating Levels

We used Tiled to combine the tilesets we have from our artist and free resources from the internet into those indoor and outdoor levels. Tiled is very easy to use when we need multiple layers in 2D levels and add collisions to the walls, trees, etc. After finishing the levels in Tiled, we used a plugin called SuperTiled in Unity to import the levels and the tilesets into Unity. It's also very easy to change the whole levels' scale after the levels are imported into Unity.

7.5 Managing the Story

The needs of managing the game's story and progression can be viewed as two discrete needs: the dialogue that is displayed to the player at any given time and the actual changes to the game world that need to occur to match the story's events. In order to handle both of these needs, we created a Story Manager script that ensure updates occur when they are supposed to.

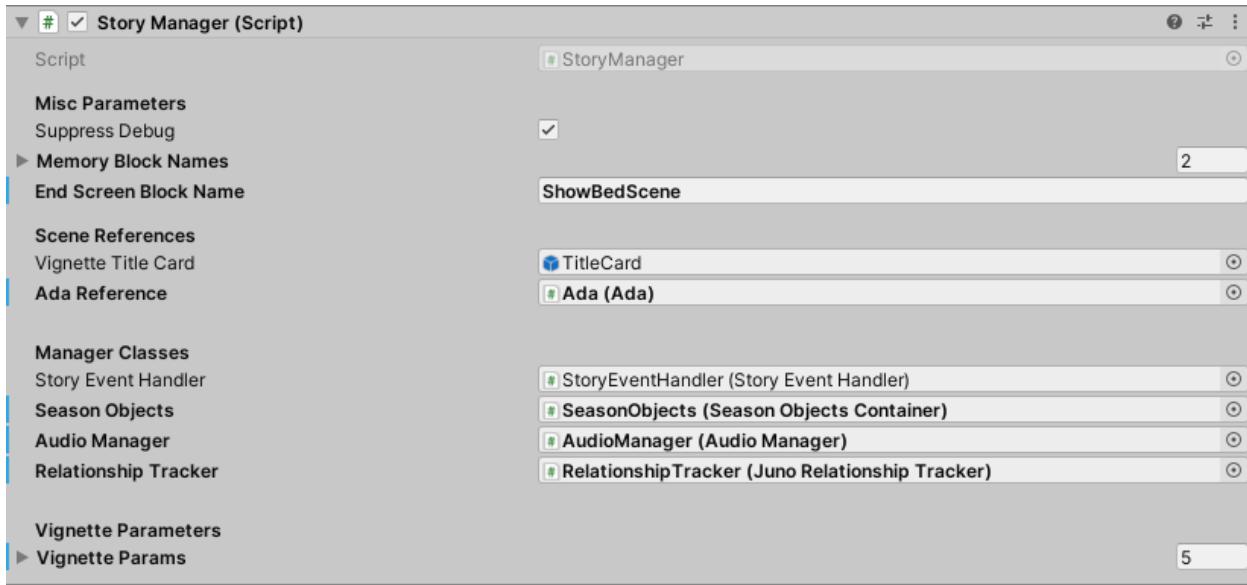


Figure 26: Story Manager script as seen in the Unity Editor

Because our story follows a specific structure of vignettes, we were able to define a procedure for transitioning between vignettes. An end-of-vignette scene is shown, followed by a fade in of a title card. At the end of each vignette, we also change the background music and the ambient sound that is playing to fit the mood of each season and give the player more immersion beyond simply stating that it is now Spring/Summer/Fall/Winter. Additionally, the player and Juno each get relocated at the start of the vignettes depending on where they are supposed to physically be for each portion of the story.

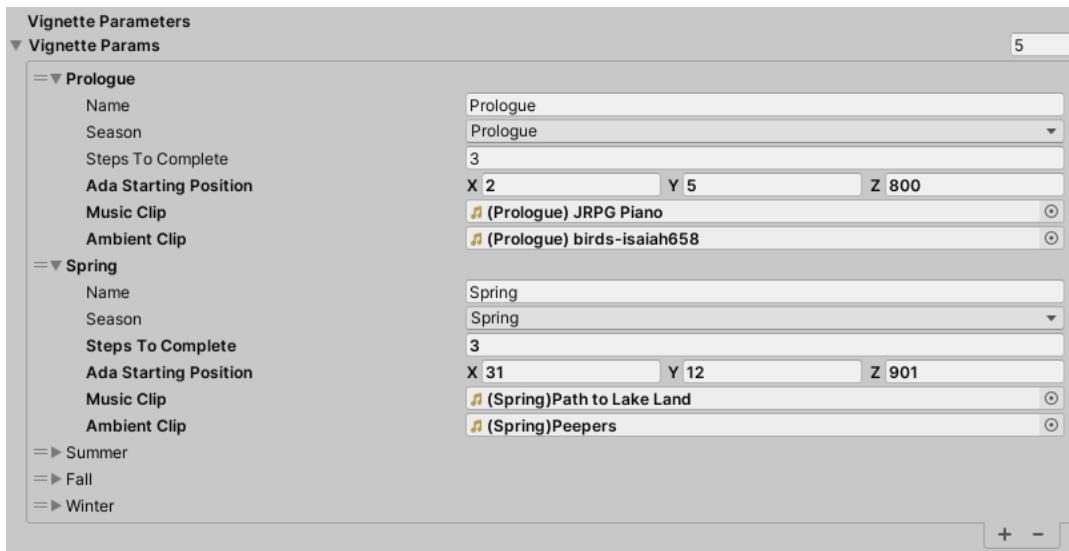


Figure 27: Unity Editor view of the Vignette Parameters struct

The exact parameters of each vignette are defined in a serializable struct called Vignette Parameters. Each instance of the vignette parameter struct specifies the name of the vignette, which sounds to play, the starting positions for Ada and Juno, and the number of steps that occur in each vignette. The steps represent predetermined story beats that happen within a single vignette. When a specific task or goal is completed during a vignette, a function is called on the story manager to advance the story. The current step number is increased, and a helper class called the StoryEventHandler is passed the current vignette and the current step. Any additional logic, such as showing new dialogue, allowing access to another portion of the level, or altering the behavior of Juno is called here.

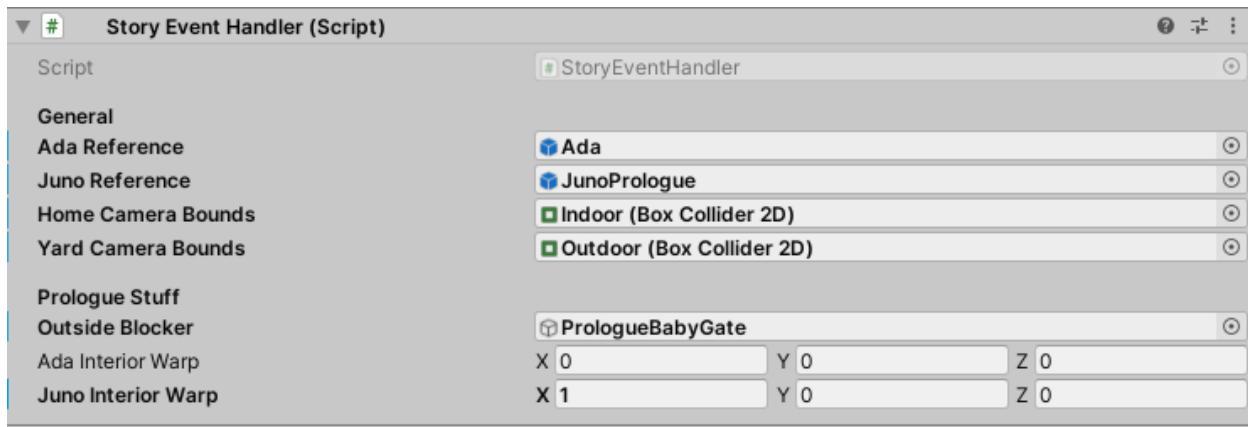


Figure 28: Unity Editor view of the Story Event Handler

7.6 Showing Dialogue using Fungus

Fungus is a free and open source plugin for Unity that provides a suite of features for visual scripting in Unity. Its most unique feature is its dialogue scripting, which has an easy-to-use interface for displaying and controlling the flow of a game's story in dialogue boxes with character names and portraits, in addition to many other scripting hooks to connect with the rest of a Unity project.

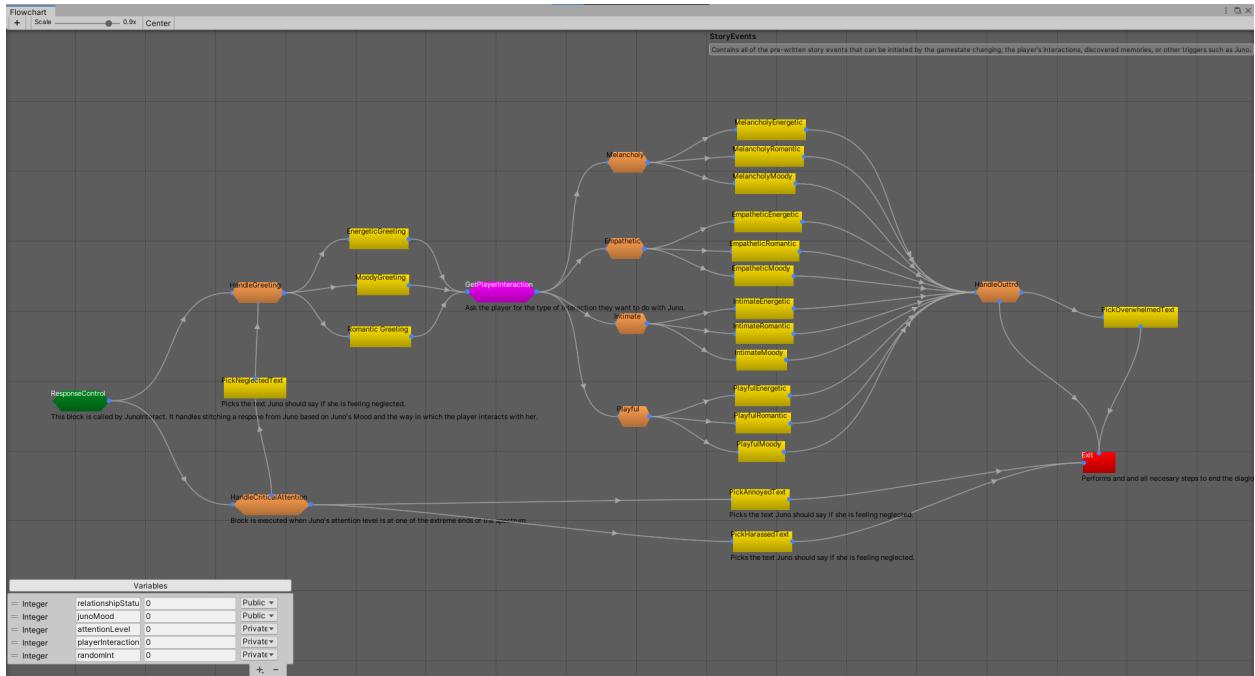


Figure 29: A view of the Fungus flowchart controlling dialogue flow.



Figure 30: Individual commands within a block. A random dialogue line is chosen here.

Our main use of Fungus is to display the in-game dialogues that we have written. Our workflow involves creating a script document for a given dialogue scene using Google Docs so that it is more easily editable and can be checked for spelling

and grammar. In Fungus, we then create a block within the flowchart where the dialogue will be said and create several empty Say commands, which each allow us to pick a character and their facial expression for the dialogue line. Then it is simply a matter of pasting the dialogue text into each Say command.

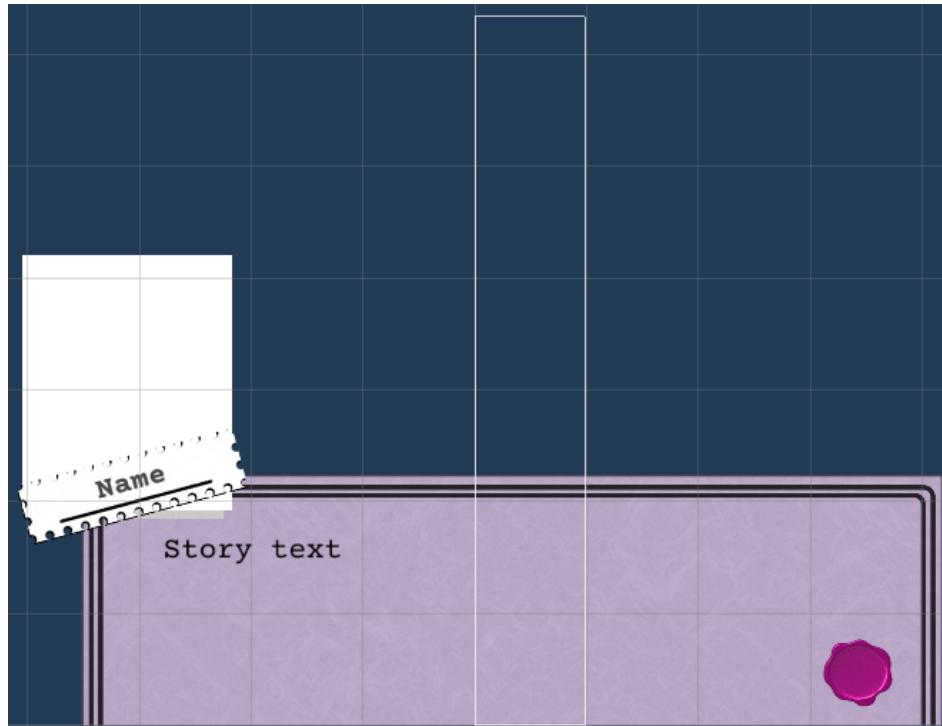


Figure 31: An in-editor view of a character prefab.

Above is a Character prefab, which contains a Character Script and a child SayDialogue. The Character contains display information for any story character, such as their name, the color to be used when their name is displayed, and various portraits of that character. The SayDialogue contains the actual UI that contains the character data and has a Writer script, which is Fungus's way of writing text to the screen.

7.7 Juno, the Autonomous Girlfriend Agent

Juno's code is broken down into separate parts to avoid unnecessary code tangling. The JunoMovement class handles the pathfinding, using A*, and moving Juno through the nodes in that path. When the JunoManager class sends a goal location to JunoMovement, the class performs the A* path finding algorithm ([reference here](#)) to find the most efficient path to that location using the predefined nodes on the level. This is all based on the predefined node system that levels have built in. Once Juno has a path, she is ready to move about in the world. The path is set as a stack and Juno's Move method goes through the path. Each node is labeled with what interactable object is next to it, so Juno can query the node graph to find a node based on its associated object.

The JunoInteractions class handles everything to do with Juno's interactions with the items in the world as well as talking to Ada. How Juno specifically interacts with each item is stored in the item's code. When Juno is inside the item's interaction hitbox, she is able to interact with it. The result of Juno's interactions fall in line closely with the results of the player's interactions with those items.

The JunoManager class is the once central class that interacts with the rest of the game's systems, interpreting the requests given and disseminating it to the appropriate channel.

Our goal was to have Juno be able to feel to the player like she was thinking for herself and acting on motivations brought on by the game-state. To achieve this, we planned to implement the Observe, Think, Act workflow. This means that she would gather information about the game-state, use that information to assess what she should do, and then take small actions to achieve the larger goal. To translate this into code, we used our Goal class. This class represents the overall goal and holds the individual steps required to complete that goal as a queue of C# actions.

What we were able to implement was a more rigid version of this. Passing in the name of the goal as a parameter, the Goal class pre-fills the queue of

actions. This way, we can define what goals Juno should do in the given vignette and pre-define what actions need to be taken to achieve it. Then, all we need to do is store the list of goals in the JunoManager class and allow the story system to call a method to start them.

7.8 Tracking the Relationship

A class appropriately called the RelationshipTracker is used to track the relationship status between Ada and Juno. In order to track the relationship, it has a public method that allows for interaction impressions to be logged from any source. These impressions represent how Juno perceives an interaction from a scale of very negatively to very positively. Each of these impression types are weighted with a point value. When interactions are logged, they are stored until the end of a vignette when their point values are averaged together in order to give a new overall impression of the relationship. The logged interactions are cleared before the start of a new vignette.

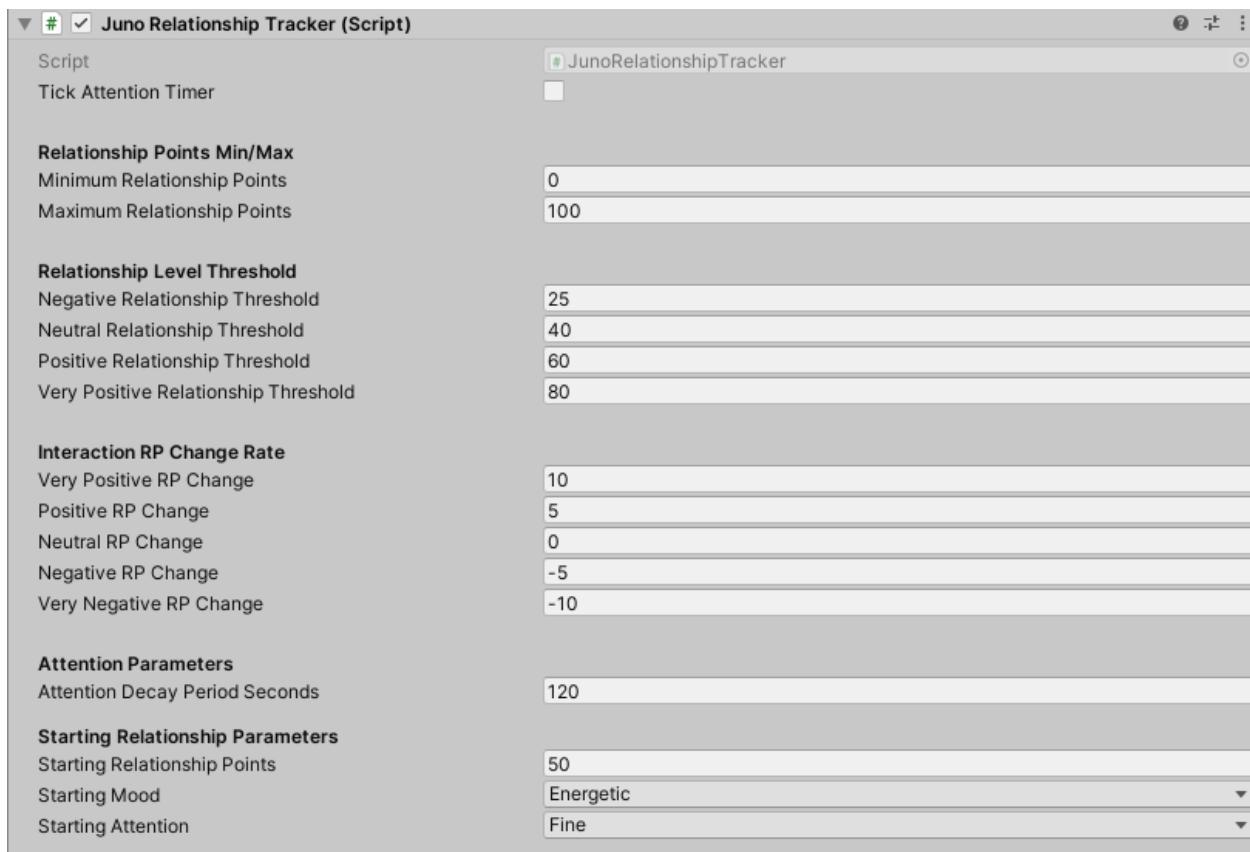


Figure 32: The RelationshipTracker exposes quite a few parameters to the user in-editor so that values can easily be tweaked for balancing purposes.

Originally, this system was set up to interact with our daily schedule system. At the end of each day, the average value of the interaction impressions would be added to a longer-term list of impressions. This longer-term list would be averaged at the end of each day to give the relationship an overall rating. The overall rating would be used to dictate which of the story beats should be played on a given day, if any. An extremely positive relationship or an extremely negative relationship value would cause a relevant story beat to be played on the next day. The goal of the system is to value each interaction and each day so that player interactions would have an effect, but very rarely would any single interaction tank or save the relationship.

8 Playtesting

8.1 Overview & Approach

Testing for Gameplay

There are many aspects to a playtest that can contribute to its success or lack thereof. One of the more front-facing aspects is the gameplay we present for any given playtest. The gameplay has the opportunity to either complement the story or takeaway from the story. If there are loads of issues with the gameplay for any given playtest, this will pull all the attention from players and leave the story hanging. Because gameplay is what players actively interact with, if there are glaring issues, then players will tend to focus heavily on that and almost ignore the dialogue and story elements that make up the other side of the game. Something to mitigate this issue is to have a running list of known and discovered bugs that we can have playtesters see and actively add to. This way, if a player runs into an issue, they can check if it's been reported already, and if it is they can move on knowing we already have that feedback.

There also exist moments where we only want players to focus on the gameplay for a given playtest. If there are new elements of gameplay or if our dialogue is in a more solid state than our experimental gameplay, then we might want to shift our focus to account for that. This can be accomplished by simply telling our playtesters our focus for the given playtest before they begin. Simply telling the testers what to focus on will effectively shift their focus so we can get more feedback on the systems we want to test.

Testing for Dialogue and Narrative Impact

While gameplay is one side of the coin, dialogue is the other side. Both are equally important when it comes to playtests, especially in a game that is story-driven, which relies on dialogue as much as gameplay to get that story across to the players. Similarly to gameplay, if there are glaring issues here, they will detract from the rest of the dialogue as well as the rest of the playtest as a whole. For example, if there are obvious spelling or grammatical issues with sections of dialogue, players are apt to hone in on those and only focus on them. This would take away from their ability to zoom out and give us feedback on the dialogue as a whole and the flow of the overall story. On the same note, issues

with dialogue can take away from gameplay in that when all the attention of the player is on the issues with dialogue, they don't focus as much on the mechanics and interactions in the world.

As with gameplay, there are times where the main focus of our playtest is to find out if our dialogue and story flow works for our game. Again, a simple heads-up to the playtesters before they begin can help shift their focus onto the specifics of the game that we want more feedback on.

8.2 Initial Playtest – Cardboard Furniture

[\[Link to materials\]](#)

This playtest aimed to see what parts of the furniture making process interested and eng[Link to materials]aged players. We did this through a paper prototype where the player created a cardboard chair in one of three styles, as dictated by their girlfriend (our wonderful playtest runners).

The main participants were other capstone students and professors in IGME. Info was collected by observing players as they created their chair and talking with them once the chair was done.

This playtest went smoothly, many players seemed to be concerned about what their girlfriend would think. They noted that they wanted more feedback from the girl friend, besides just one note on whether or not the chair followed the given style. Players also wanted some way of referencing these styles. During the playtest, when players were cutting their pieces out, they had a lot of down time and used this to chat with other playtesters. Many players commented that they loved this social time and would want to see something in the game.

As a result of this playtest, we are working on how to digitize the prototype and not lose out on the social and hands-on aspects of the game, while adding in more feedback.

8.3 Playtest 2 – Digital Furniture

[\[Playtest Materials\]](#)

This playtest was an attempt to digitize the initial paper prototype. We wanted to have a short introduction story and then have the player craft a piece of furniture for their girlfriend. This would hopefully give us an idea of how the players were engaging with the characters, as well as the crafting system.

This prototype included an exposition on who Ada and Juno were and their worries. The player would then take a request from Juno about what she wanted to make, an ornate chair. Pick out pieces of the chair. Cut out the pieces of the chair by dragging the mouse along a line. During the playtest we had a survey for players to take, observed players, and asked them for direct feedback.

We learned a lot from this playtest. While our story did seem to engage players, the mini games and crafting were very lackluster. The players found the minigame tedious and repetitive. The furniture design process did not have enough feedback nor enough options.

We used this playtest as a point to refocus on what we wanted out of the game. We decided that from a coding perspective we had immensely over-sscoped our plans for our team size. Because of this we aimed to create a prototype that will not only include simpler interactions but will have fleshed out mechanics. We also turned our focus away from detailed crafting to a more general approach for housekeeping.

8.4 Playtest 3: The MVI of Cottagecore Living

[\[Link to Materials\]](#)

This playtest was our dive into the new direction we chose for our game. We wanted to test out our new mechanics and game loop. The goal was to see if these new mechanics and in-game goals made sense to the player and compelled them to keep playing. Ideally, this would confirm that our new path was the right path and that we should continue on it. At worst, this would tell us to either find a new route or revert to an old one.

This playtest included very simple, one-button interactions that the player would use to unpack their new home. The player would use a box cutter to open boxes with furniture in them and take that furniture out and place it in their home. Once the furniture was placed, the player could clean off any dirty furniture with a sponge. The player could also check in on Juno and chat with her at any point. We did have a day/night cycle in the game, but it was very rudimentary, just resetting the player's position.

We asked the playtesters to speak their thoughts aloud so that we could write them down. We also had a google form for them to fill out to give us any extra feedback that we didn't get from their thinking aloud.

This playtest assured us that our new path was indeed the path we should stay on. We learned that we should really shift the focus of our game to the story and the emotions we can evoke with it. The very simple mechanics and interactions with the house were the perfect fit to push the story to the front. While we did have some conversation available with Juno, it was mostly a proof of concept that we could have her dialogue vary based on her mood.

Since we confirmed we found the right path for the game, we are now able to press forward and fix the many technical issues that presented themselves in this prototype. Along with the bugs, we planned upgrades to the existing systems in the game to give the players more feedback and make the interactions more understandable. We also started delving deeper and deeper into the story of Ada and Juno as well as allowing Juno to take on a life of her own.

Our goal for the next playtest was to have all the technical issues worked out, upgrading various systems like item pickup, placement, and interactions. We wanted to have onboarding for the mechanics as well as an intro into the story to give players that buy-in that was lacking in this prototype.

8.5 Playtest 4 – First of the New Semester!

Our first playtest of the second semester followed the goals we set for ourselves after finishing our last playtest of the first semester: We wanted to fix up all of our mechanical issues as well as improve on feedback systems and the grid systems in place. The name of the game was to make our prototype more player-friendly. We also added in an intro blurb so the players understood the context of the game as well as a “hidden” memory for players to get to know the backstory and the characters a little better.

For feedback systems we implemented player facing directions, a highlight effect on the object you are about to interact with, and highlighting the cell you are

about to place an item into with green. Pretty much all of these had positive aspects that tell us we are going in the right direction alongside negative aspects that tell us we could improve even more. For improvements to existing systems, we fixed some movement bugs on Ada, we made hitboxes more accurate and player-friendly, adding wider, more forgiving hitboxes for interactions with items to make picking up and interacting with items more player-friendly as well. While the addition of different sprites based on the player's direction helped players to understand where an item would be placed better, it also highlighted our lack of animations. Because the player changed the direction they are facing, some folks found themselves deeper in the uncanny valley without walking animations.

Both highlighting features really helped players to understand the systems we have in our game for picking up, putting down, and interacting with items. However, both also didn't go far enough. The highlight on the object the player is about to interact with wasn't bright enough. It made some items stand out a lot while others just seemed normal. The green highlighting on grid cells was helpful only after players understood what it was indicating. It's also worth noting that picking colors to indicate changes in state is tricky when color-blindness comes into play. To top it off, both of these effects together confused the player in that there are two systems they are trying to keep track of and often got the two mixed up.

The players loved the intro and the memory alike. The characters are shining through more than ever and players can really get a sense of their history and personalities. However, it would be a nice touch to have art depicting the memory as the text appears for the players.

After this playtest, we knew we were on the right path with our upgrades and dialogue as well as how we can elevate them to bring our playtest to the next level. Our goals for the next playtest include a brand new level with our new tileset, more dialogue between Ada and Juno, and the start of Juno being able to walk in the world and interact with the world. We also plan to start keeping track of and showing the player the current status of the relationship as well as get a prototype for "scoring" the decorations of the house.

8.6 Playtest 5

For this playtest we had the project in two parts. One had the story and interacting with items on the grid. The other had the first prototype of Juno's pathfinding algorithm.

This playtest we hoped to have made the grid clearer for the players, but found that it was still pretty buggy. We also added a tool-tip pop-up to hopefully aid with confusion with which button was for interacting and which button was for picking up. Even with this it still seemed unclear to the players.

In this playtest, we had the beginnings of Juno's AI. She was able to walk around, but she moved very slowly and would crash if the player hindered her in her goals.

On a positive note, people did seem to like the characters and wanted to know more about their stories. They were very concerned with what Juno would think about the home. People who got to Mr. Chalmers' memory really liked it, and those who have played the game before thought it went well with the other memory we had in the game.

This playtest solidified a lot of things for us. We needed to step away from the grid system and placing objects to just interacting with the home. Focus more on the tool and memory systems. This should allow us more time to make our story shine and get that amuse-bouche we desire.

8.7 Playtests 6(GDC) & 7(Asynch in-house)

This playtest represented our final directional shift in *Mending Hearts*. This prototype saw the deletion of the grid system. We decided that using the grid to pick up, move around, and place items was both out of scope and detracting from what we wanted to get across without game. As a result, we got rid of the grid system and shifted our focus to the interactions that Ada and Juno have with items and objects in the world.

For this playtest, we were able to implement our first vignette, the Prologue. This introduced players to the context of the story and introduced both Ada and Juno

to the players. The prologue starts with a monologue from Ada introducing much of the context, then allows the player to explore the house until they find the first memory item, the brochure from the day Juno left on her expedition. After this is discovered, the player is prompted to go outside and meet Juno for the first time!

We got a lot of very positive feedback as well as some helpful, yet hard to hear feedback. The players really loved both the art and the dialogue that was present in the prototype. Playtesters used the words “cute” and “cozy” to describe the aesthetics as well as the overall feel of the game, which is a huge success for us as those are some of the key words we wanted to get across with our game. The one critique that most folks had was that many of the interactions were unclear and that the balance between dialogue and gameplay was unbalanced, heavily favoring dialogue and leaving little gameplay to test.

Moving forward from this playtest, we wanted to get in another vignette that had Juno more present. Ideally, she would be moving around in the home and completing goals while the player was free to explore and accomplish their own goals. We wanted to have more interactions that are clear in the form of a cooking interaction that allows the player to make food.

8.8 Playtest 8(Imagine RIT)

Our final playtest saw the more complete realization of our final direction for *Mending Hearts*. We continued with our interactions focused gameplay and were able to get in most of what we wanted to add from our previous prototype. This manifested itself in the spring vignette! In this vignette, we completed our goal of having a cooking system that allows the player to make two different recipes with the ingredients available to them. Additionally, Juno is fully present with the goal of helping Ada make breakfast by taking an ingredient out of the fridge and adding it to the oven. Ada can have conversations with Juno before, during, and after the creation of the breakfast masterpiece.

A lot of the feedback we got from playtesters here echoed or mirrored the feedback we got with our previous playtest. A lot of the testers said that they really enjoyed the direction that both the art and the story were going in. We did

see a lot more of our playtesters start to intuitively understand our interaction systems as well as some of the nuanced hints we give to what can and cannot be interacted with in the world. The biggest point of critique remained the same however: There is a lot more dialogue than there is gameplay and it makes the game feel unbalanced. To some, it felt more than a visual novel than it did a game where your actions truly impacted the story.

8.9 Future Work

Future builds (and therefore, playtests) would need to focus on testing the player's experience of their relationship with Juno. This was one of if not the most experimental aspects of *Mending Hearts*, and it is one that has been left largely untested due to the frequent restarts. In the next iteration of *Mending Hearts*, our team would focus on polishing and expanding upon the interactions between Ada and Juno.

This hypothetical next build would require some fairly large refactors to hone-in on the experience we would want to playtest for. As is, the experience is a tad diluted with how many stop-and-start cutscenes there are in between gameplay portions and all of the interactable objects in the home. Perhaps we could utilize Hanrui's outdoor levels and focus on the Summer vignette, which was written but never implemented. This portion of the story centers on Ada and Juno going on a hike together. In this setting, the two of them are the focus. In this scene, we can measure how often the player and Juno interact with each other, give Juno more autonomy in how she approaches and talks to the player, and test for what the player's perceptions of these interactions are. Does the system we built feel life-like or annoying? Is talking to Juno interesting or is it a chore? These are questions that are crucial to *Mending Hearts* that we largely failed to answer.

9 Postmortem

9.1 Overview

“Cozy, Sentimental, Uplifting” is a core we stuck with, and we handled pivots well. In this section we review the whole process of the team design and develop Mending Hearts. We will talk about how and why we made and changed those design and technical decisions. Beside the design and development matters of Mending Hearts, we are also going to look into how we work as a team for the past two semesters to see what we did well and what needs improvement as a team. Last but not least, there will be things about whether we will continue working on Mending Hearts and what we will do to improve it if we want to continue working.

9.2 Reflections

From the beginning, we had a solid idea of what we wanted the emotional core of the game to be, but we were unsure of how we were going to execute it. “Cozy, Sentimental, Uplifting” is a core we stuck with, and we handled pivots well. We took influence from several games we had played before, but definitely took a very cursory approach to our market research that did not help with our struggles to determine our mechanics. We took on a large scope and definitely tried to do more than our four person team could handle. On top of this, we followed a very low ceremony agile style that led to a breakdown in communication and made it much harder to follow our risk mitigation plans.

9.2.1 Game Design

One of the constants during our game design process was the emotional core that we wanted for our game. We wanted it to be very cozy, sentimental, and uplifting and we let those words guide much of our design. The issue that came along with this is that we didn’t have a solid grasp on the specifics of the game like mechanics. Because of this, we ended up going down a few different paths in

an attempt to find those specifics. Eventually, we did find what we wanted *Mending Hearts* to be, but it took much longer than anticipated and put us behind on our development schedule. Once we had a direction for the mechanics and story of our game, we took on a lot of experimental design choices in an attempt to get our emotional core across. The issue with this is that having so many experimental systems essentially diluted each other. Because we were spread so thin, each one of those systems felt like it got health-developed whereas if we had one or two of those systems, we could devote much more of our time to each of them. The end result would be two fully-fledged systems instead of three to four half-baked systems. On the positive side of our game design, because we knew our emotional core so well, it allowed us to thrive in areas that were direct contributors to that, such as character design, level design, and dialogue and story flow. In each of our playtests, these were aspects of our game that players always enjoyed and gave positive feedback about. They saw these three aspects of our game as the leading contributors to the cozy, sentimental, and uplifting feeling that we were trying to give to the players.

9.2.2 Technical Design

We tried a lot of more experimental technical aspects for *Mending Hearts* and while a lot of them worked in their own way, we learned a lot in the process of getting them to work. One of the biggest issues we ran into was the amount of direction changes that our game needed making a lot of our work unusable. This meant that a lot of the time we spent developing our game was either thrown away or needed to be reworked to fit our new directions. Another issue we ran into was that when we were coding our systems, we coded with short-term solutions in mind. This meant that when we developed systems to interact with those short-term solutions, they didn't fit well so we had to refactor them to allow for the interactions we wanted. Our production issues bled into our technical issues somewhat in that because we were so low ceremony, our task cards often had no acceptance criteria or definitions of done. This meant that when it came time to merge code into our main branch, the code often needed adjustments or additions, which again used up time we did not have to spare. Lastly, a lot of the coding we did was as individuals and was rarely collaborative. This meant that when it came time to mesh everyone's systems together to create our next

prototype, it was confusing and often required adjustments to code. All in all, our technical design worked, but caused a lot of problems on the back end that we could have solved on the front end and saved time.

9.2.3 Production

One of the biggest issues we have had with production was our decision to have a low ceremony process. We had very little task definition which led to a lot of issues on the technical side. We tried to have some towards the start, but we decided that it didn't add anything for us. Looking back, it would have saved us from a lot of back-end work. This also would have helped us to understand how much work we were deciding to take on during a given sprint and realize what our scope was. One of the biggest misses was not holding sprint retrospectives after each sprint. We held a couple in the fall semester and decided they didn't add value for us. These would have helped us to fix a lot of the process issues as we worked if we took the time to analyze them during our development cycle. Another set of issues we had are under the team dynamics and formation. Because three out of the four of us worked on a team prior to this and found success there, we assumed this team would work very similarly and find similar success with that process. This was definitely not the case. The context of our project mixed with personnel changes meant that this was a completely different team and needed a new process to match. With that, a lot of our meetings were slow to get started and often had a lul during because of the false assumptions we made about this team compared to our previous team. There were many times when a question would go unanswered for an uncomfortable amount of time. The big issue with team formation lies in the fact that not all of our team joined because of a unified vision and shared passion for the project. This meant a slow start with both the design and development processes and was a core factor in our meeting having that strange, slow atmosphere.

Our production process was not without successes! We included documentation in our weekly sprint-end meetings so that we would work on it as we went. This kept a lot of our documentation up to date with the active changes we were making to our game, saving a lot of time towards the end of the design and development cycles. We stuck to keeping our documentation as up to date as we could, or at the very least marking down where to find the information throughout

the project, even when it meant some crunch and late nights. We also handled the hybrid model that our team had very well. Since we were all forced to be virtual for our first year in the graduate program, we were used to being able to work remotely while still holding meetings and making progress in our game. With the addition of being able to be in the lab in person, we used tools like discord and zoom to help us stay connected with members who could not be in person that day. This gave everyone the flexibility they needed to handle life issues as they popped up while still allowing us to make progress on *Mending Hearts* and keep communication strong.

9.2.4 Playtesting

While we did not always feel prepared for our playtests, they were almost always received really well by the testers. Playtesters always gave us amazing feedback on both the mechanics as well as the dialogue and story flow of our game. There were moments when the feedback was hard to hear, but it helped us to iterate through visions for our game and get to the final product we have now. We feel like we were able to gather and implement feedback well. For example, there was a lot of feedback about Juno being from a military background. This feedback helped our team start the discussion of whether we should keep that or change her background. Without the feedback from testers, we would not have scoped down and gotten rid of the grid, which was confusing to most players and took away from the interactions in the game.

9.3 Conclusions

On a whole, we think this project went pretty well. We are very proud that we were able to identify and stick with an emotional core and use it to develop our characters and story. We are also very proud of our ability to understand when our iterations don't work and change direction in order to improve the game and the player experience. If this project were to continue, we would heavily revisit the production process that we had. We would look into having a higher-ceremony process than ours was and holding sprint retrospectives to ensure we don't get stuck in the development cycle and ignore process and design.

We are walking away from this project with a few major lessons learned. The first one being when developing a game, pick one really cool and interesting idea and work that to completion, don't pick multiple and leave them all half-baked.

Another takeaway is to form a team based on a very clear, shared vision and obvious passion for the game you are creating. Without these, passion turns to “I just want to graduate.” Our last major lesson learned is to find the fun in both the game and as a team. Doing this will help you to understand what your game needs to be and what you can do as a team to make your process that much better.

9.4 Future Work

Mending Hearts is an experimental game. Much of capstone design and development was focused on the initial research and prototyping of the project. In order for the game to be marketable, even in the indie space, we would have a long way to go. Though, we do not necessarily think a finished product of *Mending Hearts* needs to be marketable. We are happy to keep the game in the experimental category to help guide future developers who want to look into relationships and NPC design in games.

Currently, we have no plans to develop *Mending Hearts* further. All team members are in agreement that we have reached a stopping point. While we feel that we have a good understanding of the game finally, we all need a long break from working on this game to refresh.

If we were to continue working on this game in the future, we would break development up into segments, each segment focusing on a new vignette to add. We would also need to recruit more artists and possibly another developer. Regardless of whether we get more people or not we would also want to simplify the scope to focus on one experiment. Trying to juggle accurately portraying relationships with a realistic NPC was definitely more than we could chew. Either on their own should be enough to build a game around and make for a successful experiment, even if it does not make a full marketable game.

10 Appendices

Appendix A: Play Test 1 Notes & Materials

Player Instructions

Planning Phase

1. Your playtest runner will tell you what style of chair they want you to build. (This represents a conversation you would have with your partner about what furniture is needed.)
2. From the bundle of blueprints, pick from the available furniture blueprints to create the piece you and “your partner” (the playtest runner) would be happy with keeping in mind the style they told you.
3. Present what you have designed to your partner (the playtest runner) and receive feedback on if it fits the idea your partner had. You may choose to repeat step three if your partner doesn’t like the piece you have chosen, or you may go ahead and build the piece you have designed.

Building Phase

1. Use the scissors (or box cutter) to cut out the furniture presets that you have chosen. (These will be drawn on the pieces of cardboard you picked. Cut them right out of that)
2. Cut out the slots from the seat for the back and legs to be inserted into.
3. Drag the sanding paper over any rough edges created when cutting the pieces.
4. Assemble the furniture by sticking the pegs of the back and legs into the slots on the seat.
5. (Mixing Colour) Pick a colour swatch from the list of available colours. When you have picked the perfect colour, use the pastels and the paper provided to match that colour as closely as you can. (Being a shade or two off is okay.)
6. If you would like to upholster your piece of furniture, pick a pattern from the available fabrics and cut a fitting amount from that. Using the push-pins, staple the fabric to the base of your chair.

Emily's Notes:

Erika had a bit of confusion at first about picking a design.

She says she had a tough time visualizing a final product.

Erika skipped ahead without me giving her feedback.

Then she got a phone call and had to leave.

Not having the perfect choice made it more enjoyable.

Eric liked this a lot but he is a DIY guy

Weez really likes crafting, wants to mix and match colors

Lumos and Little Big Planet are all about materiality, very textural designs and materials

Player feels a lot of pressure about their furniture choices and design, being judged

OPTION to mix colors. Not a requirement.

Gav liked doing things with their body. Love Language: Acts of Service. A means of expressing love and care

People are definitely very nervous about me judging their designs

Sawing is very time consuming IRL. Students are using it as a time to talk to one another. Digital version could be faster, but this could also be a moment you talk.

People are really getting into the idea of doing things well for their partner. At all steps but especially in the beginning when they are making the most decisions

sawing badly Oh no my gf isn't going to like this

Scissors vs. Knife IRL, maybe different tools in the game.

Getting judged is a lot, according to one player.

Player wants to keep their chair

Player also noticed the talking during assembling. Dialogue between players was an interesting dynamic.

Another player said it was nerve-wracking.

Players want some way to know if they are doing a good job or not. Don't want to disappoint their partner.

Furniture customization was fun and a fun twist on interior design games.

Your partner should be more involved in the furniture building process.

"I really liked assembly"

"Why not make a physical version?"

Sawing was a struggle, everyone wanted the one knife.

Had fun playing with other people. Really liked the social mechanic to it. Games like this are really fun for sharing your creations with other players.

Players really liked the social aspect.

Players wanted more feedback

People like the fact that they had to *make* their design, and that they weren't just selecting their design.

Veronica's Notes:

Player 1: Modern Chair

- Needs a reference for it, goes to google
- Do you have any favorite colors?

Player 2: Victorian Chair

- "Can i just make a regular chair and have my partner wear a corset when she sits in it"
- Enjoyed choosing, did actually care about what the partner wanted even tho he was threaten
- Gets cutting fatigue
- Cleans up during

Player 3: Antique Chair

- How many colors can I take?
- Concerned about clean up

Player 4: Antique chair

- Wants tinted stain
- Going through pieces and debating the aspects of each
- Only wants to upholster the seat
- Has an idea that isn't in the pieces
- Wants to know what the property is like
- Cutting sounds unpleasant
- Likes imperfect choices

Player 5: Victorian

- Goes straight to cutting

KP: make sure that you are getting the expierence

Sawing: takes a lot of time, people usually talk among themselves

Ben Snyder: Modern chair

- What is modern???
- Comparing to furniture in own house

- Lost just given a style, just took as contemporary
 - Kinda made a high chair
- Was I allowed to ask for more clarification?
 - Need more info than just a style
- Build phase felt disconnected from beginning
 - Check ins possible?
 - Overall unsure of what could be asked
- Emotions not really there, a little from
 - Need prompted feedback probs?

What happens when u mess up

“Dark souls of furniture making”

-no check points no respawns”

Likes the in person component

- Talking with each other doing the stuff
- pieces not fitting in to what was thought they could be

“Most fun” - really liked the inperson elements

Appendix B: Playtest 2 Notes

The story is interesting to me	The characters seem interesting	I have an understanding of the character's personalities	Any comments on the story?	The furniture selection process is give me easy to follow	The furniture options process is give me creative	I felt as though the parts were interchangeable!	Any additional thoughts on furniture selection	The minigames were fun	The minigame felt like it correlated well with the process	Any comments for the minigame
Agree	Agree	Agree	I like the premise, but the interactions feel slightly exaggerated, speaking as a woman in a lesbian relationship. Maybe I just don't like pet names personally, but it feels like they're playing up their relationship for a camera rather than being organic.	Disagree	Agree	Strongly Agree	It wasn't clear that once you click, a piece has been selected. I clicked the chair backs a shit ton until I realized it had probably already been selected. Some sort of indicator (like a checkmark next to the selected piece) would be fantastic.	Agree	Strongly Agree	The click and drag mechanic wasn't immediately obvious, but it seems like that's just because the instructions haven't been formally implemented yet. Otherwise I thought it was a good minigame with an appropriate level of difficulty. Would've been easier if these mousepads didn't suck.
Neutral	Agree	Neutral	I haven't had enough time with the characters to have an idea of who they are, so take my responses above with a grain of salt. The interesting part is what Juno is trying to forget. Maybe it's just generic war stuff, but I trust it to have a little more depth than that. If not, then the character dynamics need to be extra fleshed out to compensate.	Neutral	Neutral	Neutral	The furniture types seemed very much geared to go with other parts, like they're coming in sets. No comments about whether that is good or bad. I noticed that, of the three types for each of the three sets, there weren't necessarily three different sets; it's interesting to have parts that seem like they "belong" together without having all your choices selected for you by that fact.	Neutral	Neutral	I connected the dots and assumed the minigame was to represent carpentry/sawing, but it's not abundantly clear. Extra effects could help with that. I noticed that every minigame had me follow the same path. I'm unsure if this was intentional or not.
Agree	Strongly Agree	Agree	I liked seeing the thought process of the two characters. I was fun to see that they were both nervous about the new house and help make both character feel grounded. It made it feel less like a one sided relationship. The only thing was I thought I would be playing both character equally instead of focusing on just one. Having seen the pitch, I knew I would be the carpenter, but that was the feeling I got from the game itself. The personalities of the characters themselves don't feel too different from each other yet. However, I enjoyed the back and forth between the characters. The dialogue was stronger than the internal monologues. The flashback felt a little heavy handed. I thought the bunk discussion served as a smoother way to tie in her military background and could have been a better time to have her go through a flashback instead of the forest.	Agree	Neutral	Disagree		Agree	Agree	Having some sort of instructions could be helpful, as I was a little lost at first. Additionally, having some way to take a break from holding down click might be helpful for some players.
Agree	Agree	Neutral	You mentioned that the furniture selection system fell through the cracks, and I fully agree. There was not much user feedback that things were being selected, and I was not sure what the whole thing was about until I pressed done. Maybe having the chair assemble itself in the black space under the selection while pressing the icons would help with that? Also highlighting the icons that were selected I feel would go a long way.	Disagree	Neutral	Neutral	Neutral	Neutral	Agree	It's hard to comment on the minigame since I know it's still in an early stage. I enjoyed the minigame itself and though it served as a fun way to tie in the woodworking. Although I think I would get bored with it if I had to do a whole bunch.
Neutral	Agree	Agree	I enjoy the dialogue and the characters, and I generally understand the story set up, but there is not much for me to say whether I *enjoy* it or not.	Disagree	Neutral	Disagree	I feel as though there should be a larger model that shows all of the pieces put together as you were selecting them, or at least have the selected item be highlighted to indicate that that was the piece being selected.	Agree	Agree	I felt that the minigame was too repetitive and a little janky. As I stated before, more user feedback would go a long way. I wasn't really sure what the game was asking me at first. I also think having the cuts change between each chair part would make the minigame less monotonous.
Agree	Agree	Strongly Agree	The story seemed engaging enough to get me to want to learn more.	Disagree	Agree	Agree	Agree	Agree	Agree	I think a variety of line tracing patterns would make the game more engaging and enjoyable.
Agree	Neutral	Neutral		Disagree	Disagree	Disagree	Neutral	Disagree		
Strongly Agree	Strongly Agree	Strongly Agree	butt > house	Strongly Agree	Agree	Agree	cute	Neutral	Agree	pls I don't have diamond adc mechanics T_T. Make it a bit more forgiving pls.
Agree	Strongly Agree	Strongly Agree	I liked it. I can sense some sexual energy and idk how I feel about that.	Disagree	Neutral	Agree	I could tell it needed to be ornate but I wasn't sure what counted as ornate or if it even mattered.	Strongly Disagree	Neutral	I personally don't like those kinds of mini-games. It's very precise and if I'm not using my own mouse / computer then it's a lot harder and stressful. Especially since I could see the paths being more complex and longer. If you plan to keep the same mechanic, consider adding some checkpoints / making it wider.
I am confused by the story and have no feelings towards it	Neutral	Agree	No comments on the story, but I did like the script. The script made the relationship very clear between June and Ana. It's quick and it gets the viewer up to speed without lengthy introductions.	Strongly Disagree	Neutral	Neutral	My issue does not lie with the furniture selection, but with the mechanics surrounding it. I thought it was very confusing on how to "select furniture." In fact, it was so confusing that I did not understand what was happening and ended up mashing the mouse button and clicking through it entirely.	Strongly Agree	Disagree	I thought that the minigame did not correlate well mainly because there was no context surrounding it. Yes, I was building a chair for my partner, but the game did not show any instructions on what to do or how to do it.
										I found an exploit where if you have a touchscreen device, you can select the start and then immediately go to the end. While it did skip the minigame entirely, I felt like I did not miss out on much. I guess you could say that I felt there was no merit in completing the maze legit; take it as you will.

Rye's Notes:

Erika

- Need some feedback for the Catalogue
 - Which ones they have chosen already

- Show them the preview and this might fix it?
 - Also, just make the one you have get like a change of colour or something?
- The minigame for cutting is way too hard
 - Make the path bigger
 - Make the path follow the outline so it actually looks good
 - Need to tell them when they enter a fail state and have to restart
- FEEDBACK
- “Dialogue in the beginning felt a little long”
 - It was nice to know “this is why it’s important that I make her happy”
 - Took her a while to tell who the player was playing as
 - “Who am I, and who are they?”

Rando 601 Student

- Did the survey
- Maybe they’ll talk to use about it

Another Rando 601

- “I love this” ~ talking about the art - character profiles
- Accidentally skipped over some of the dialogue
- Chuckled at the “worthless lesbian” comment
- Catalogue
 - Not sure they had clicked on it.
 - Again need some feedback on it
 - Something to tell you this has been selected already
- Got to the minigame
 - “Ooh!”
 - Sounded good?
 - Was way too hard for the player (Again)
 - Let out a little “whoo” when he did one successfully
 - Blamed the mousepad ... LOL
 - Used the console as feedback for now - kinda what we had going on but deff need feedback in the future
 - Sounds? Icon?
- 1 on 1
 - Art was really good

- Dialogue
 - If the dialogue was short sometimes it would skip with a quick
- Excited for the combo of visual novel and gameplay
- Cutting
 - “Felt it was pretty intuitive.”
 - FEEDBACK

Erik

- Sudden change in background
 - Kind of jarring?
 - I think just juxtaposition of the two themes
- Catalogue
 - Kept referencing what GF wanted
 - Again, need feedback to tell them that they picked one and it's set
 - I really need to put the preview there
- Minigame
 - Got the first cutout on the first try
 - Confused on why doing it three times
 - I think changing the outline would help out making sure they understand they are cutting other stuff
- After game thoughts
 - Brought up flash games that had it so you had to keep the mouse inside of a shape
 - What you have to track is dynamic
 - Like following one blade of a windmill
 - Neat opportunity for level design in a unique way
 - Used to the idea behind the cutting minigame
 - See above for “getting the player to feel like they’re using the tool” with the simple mechanic
 - Did say it’s not the same as mobile, unless you’re on a tablet
- Enjoyed dialogue and characters
 - Had one moment of “uhhh, okay?”
 - Enjoyed their back and forth
 - Picked up the difference between the two
 - Knew whose dialogue to focus on ONLY because we’ve had conversations with him about it before

- Background of text being different for which character is talking
 - Menus could change to fit who “interacts” with the menus

Rando Student That I guess was in my 106 last semester

- “Oh wow” when the scene changed from spaceship to the outside of the cottage
- “So much drama” ~ I think he likes the dialogue?
- “Art style is fun” :thumbs_up”
- Catalogue
 - Knew to click the side buttons
 - Knew to click the sprite, but was waiting for feedback to tell him he picked something
- Cutting
 - Read the console for feedback - kinda fine for now because we don’t have any yet
 - Pretty tough time first component, but fine after
 - Also expected different cutting shape for the different pieces
 - As he cut more, he got better and better
- Took the survey

A professor I think?

- Very quick transition from dialogue to catalogue
 - Comparative to the transitions between background
- Catalogue
 - Again, no feedback to tell him what image he picked from the options
 - Have something to tell him what he picked
- Cutting Game
 - Again, super abrupt transition. Maybe we could have like a “camera” that “zooms in” on a saw table or something
 - Needs directions, not intuitive
 - Asked about controllers
 - Assuming we won’t move to a controller because difficulty
 - There should be *some* leeway in staying in the lines
 - Need different shapes for different components
 - Could change the cursor to a saw
- Talked about the idea of picking up the “pieces” of furniture in the world

- Maybe show the characters in full grabbing the components for the chair during convos
- Talked about showing the items on the cutting table in the order so you can see ahead of time
- Talked about maybe just having them click on certain points and it “cuts” for them
 - Giving them the illusion of cutting it
 - Maybe not dragging the whole way
- “How many pixels are there”
- “Loved the story” ~ the juxtaposition of roughness and beauty
 - Rustic and

Rando Rye Thoughts

- I think the worthless lesbian line might hit well with queer audience, not so much with cis-het audience
- I know the idea of mobile was brought up
- I think we can combine the painting stuff right now (drawing as you drag) can be helpful for feedback
- Some of the dialogue is a tad too long and is covered up by the button

Veronica's Playtest Notes

Victim 1

Looking for feedback on furniture selection

- Clicking randomly and switching back and forth

Paying attention to console during cutting

- Seemed to move cutting smoothly tho

Gav

Suffering through cutting, heavily relying on debug

Story

Met their prior understanding of the game

Enjoyed it personally

“Love some characters who are going through stuff and have anxiety”

Only got to know their worries, if this decisions right for them as people and a couple

Curious to what choices they would make

Felt as though there was a tone switch, Juno led the conversation about decision after Ada seemed to be the one suggesting

Rhythm stops are impactful, liked Juno stopping for a moment and forcing the player to wait

Dialogue in the beginning was telling, would like it to show through situations, characters alone at the start was okay, prefer when they bounce off of each other

Show don't tell

Convey thoughts through conversation not through inner dialogues

Furniture not that compelling,

Need more feedback

Unused feedback

Unsure of how much room for error (ie, is there a best answer or is it up to preference)

Seemed to be 3 distinct matching sets

Mini game not fun

- Cutting takes too long, doesn't feel good
- Needs to feel more good for process

Needed feedback from Juno from process

Victim 2

"Wow that started off strong"

Seems to chuckle at the dialog, im taking that as a good thing

"Uh -oh" - when Juno kinda fades

Clicky clicky around in furniture selection

Looks over to console for feedback at the end of each line

Stamp overlaps text at one point

Enjoyed characters and story

Needed more feedback for selections on furniture

Ben

Why does the button change if u can click anywhere

"Uh-oh I was not warned there would be foul language"

Enjoys the thought of the od just coming into the atmosphere

Seems to be enjoying the dialog a lot
Dramatic pause was effective
Does not think chair is simple
Really enjoying the dialog
The pacing in the dialog is a little strange. The punctuation combined with the dialog loading in
“Is this a chair leg?” with a comment about are you sure this game is for everyone
Thinking about ornativeness of the pieces
Figuring out constraints of click and drag game, “but if ive already cut a line and i cut a line next to it *some thing something*”
Wants feedback on how it fits the theme
Pieces are too small
Does not know enough about the characters/ plot rn
A lot of layers happening at once, so the layers kinda get muddled together
The parts didn't seem interchangeable
Lacking in a lot of feedback
No breaks and feedback was not bueno bueno
Didnt get the feel of cutting out

Victim 3

Very quiet, umm i was mostly focused on ben's

Adi

Seems to be invested in story
Seemed to have an okay time with selection
Looking intently at debug
Cutting is hard
Changing angle to get the cutting right

“FF, im going to sleep on the floor, my girlfriend will mhate me”
Narrative “Excellent”
“I didn't know to make a chair u needed the mechanics of a diamond level ad carry”
Wanted feedback on selections

Liked the modularity of the pieces
“The game is not fun, i mean it was fun but i kicked my ass”

Shazam

“Oh this is like scifi”
Needs more feedback
“This reminds me of surgeon simulator”
Confused on whether or not doing the right thing
Would be nice if the shapes changes
Enjoyed the narrative set up.
Want feedback on part selection
Doing the same shape is easy
Want to see end product and place in house

Final Thoughts:

Need more feedback in minigames
Catalogue was really hard to understand the state of what you selected
Furniture didn't appear interchangeable
Option to “go back” in dialogue
Character portraits should emote
A bit too much dialogue before gameplay/agency
Less telling, more showing
A few too many layers of story all at once; gets muddled
Some players were confused about which character they were
“Useless lesbian” – Note that straight people might not get queer slang or in-jokes

Minigame was... okay? People were mixed. Biggest consistent feedback is more variety

Could this game be on mobile? Could it be on controller?

Everyone LOVED the character portraits
Lots of people liked the aesthetic
Most people enjoyed the characters

Appendix C: Playtest 3 Notes

Rye's Notes:

- Dropping items on day change is kinda jarring
- Accidentally dropped a box in front of them and can't move
- "Have to be in just the right spot to pick up the [item]"
- "Part of the confusion is not knowing where i need to be to interact with something"
- "Camera feels like it moves slowly compared to character"
- Moving back to start on day change is jarring
- Bugs with collisions
- Not intuitive to the player what to do
- Can hold items & double interact with GF
- Hard to pick which item you want
- "Clean furniture" looks dirty
- Need to handle multisquare items
- Movement is kinda janky
- Sponge in wall
- No take sponge out of wall
- Need a clear way to know where you are going to place furniture
 - And which furniture you're going to pick up
- Feedback for day & timer and when it's going to change etc
- Movement: i think the components aren't set to 0
- Struggling to figure out how to interact with an item
- Put an item inside another item?
- When interacting with Juno maybe tell them what they're picking before giving them the options
- For the intimate options "oh.."
- "What puzzles me is what Ada said for Juno to respond the way she does"
 - Wants to have dialogue come from Ada that reflects the emotions they picked
- Noticed the time cycle almost at the end of day 2
 - Need better feedback for it
 - Light and shadows?
- Resetting is jarring again

- Looking at controls frequently because they expect e to be both pick up and interact with
 - “I feel like pick up / put down is interacting with something”
- Put the sponge behind the table and can’t get it back
- Didn’t get how to get the furniture out of the box
- Size of the furniture not fitting in the squares
- Want an indication of open boxes being empty vs being full still
 - Having something poke out of the box maybe?
- “Action the player does is realistic, but the environment isn’t”
 - Size of furniture vs size of box
- Hitbox issues and pickup placement issues
 - Want it more smooth and consistent
- Confused on why adding day cycle
 - “Don’t know how long each day is”
 - Want indicator of how long it is
 - “Want to know ‘it’s almost night, so I want to be wrapping this up’”
- Like the loose cameral following, but might be an issue with a bigger level
 - Likes how it acts on day change better than insta-warping
 - “A bit relaxing almost. No sudden movements. Everything happens in a flow.”
 - Feel like it fits the theme
- Entering multiple dialogues
- “Kind of intuitive to take an item out of the box”
- Maybe want to move through tools - or make character smaller to fit through gaps
- Trash for the empty boxes or a place in the house just for them
- Loose following is good, but too slow at the moment
- Have placement indicator for pick up / put down
- Erika did not want to walk up to Juno and interact with knife in hand out of fear of messing it up
- Not super understanding the prereq for interacting with objects
- Also, we need to find a way to tell which bits of the level are interactable and which aren’t because a lot of folks tried to interact with those crates in the house
- Hotbox issues again :(
- A lot of folks think the clean furniture is dirty and try to clean it

- Confused and lost - intro would definitely help out
 - Who I am, why I'm here, what I'm doing
 - Princess Bride meme
- Need some direction in what to do (the order)
- Difficult to interact with objects based off player pos and hit detection
- Want dialogue to go with the emotion
 - Intro to the dialogue
- “Testing computer input rather than roleplaying a character”
- All design decisions should have meaning and be based off player data
- Interaction with juno persists to the next day and it seems like WASD moves the selection
- The loose follow camera wasn't great for one player because they felt it hid what was in front of them and they wanted to look ahead and what they're moving towards physically.

Emily's Notes:

Appendix D: Playtest 4 Notes

Rye's Playtest Notes:

Person (I don't remember his name :sweat smile:)

- Camera is a little slow for the speed of Ada
- Character art is god-tier
- A lot of "oh's" when picking up and placing objects
 - I think you can just grab an object from across the room
- More confusion with control scheme
- "I kinda wish the box cutter was in an inventory"
 - SO you don't have to hold it to use it
- Seemed like he was getting boxed in with all the stuff - need more room - new level design if we want this amount of items.
 - Maybe go for a design with no furniture in it
- "I do like the targeting {system} on the ground when you put down items"
- "I love the tables in boxes"
- Seems like he figured out why objects are highlighted after a bit
- More funky hitboxes
- Blocked off a whole part of the house.
- Not super clear that it's supposed to be in the future. Mention of space was kind of out of nowhere
- During memory the black didn't cover the whole screen - did we want it to?
- "Curious as to why (something something cryospace something something If it's only two years" - he mumbled :')
 - Is it a scope thing?
 - Would love to see art assets of the conversation in the background of it
- When there are tons of items, it gets hard to select the one you want
- Highlighting feels a little funky
 - But seems to be a good system for indicating what we are picking up/interacting with
- Tried to go through door
- Didn't try to break down boxes
- Liked the dialogue, but would like to see still-lives of the conversation as it appears
- Might want to make pick up / put down / interact feedback more obvious

- It's too subtle right now

Triton

- Another slow camera follow comment
 - Damn you delta time
- Hitboxes on table/chairs
- “Keep forgetting pick up and interact are two different buttons”
- Interacting with the item you want when there are a lot in the area is janky
 - Might want to adjust to be based on player direction
- Also didn't try to break down boxes
- Also very surprised at the mention of Sci-Fi stuff
- Not communicated that we can break down boxes
 - Do we need to? Or can we have the player exploration be the key to that?

Other human being (don't recognize them = not a grad student)

- Might want to lock the player during the intro dialogue
- Seems to have picked up the interactions quickly
- Trying to organize as best as she can
 - But you can't be super specific with where you place items
- Picked up on the highlighting / green cell feedback really well

Other not grad student

- Tried to interact when it was in her hands
- Immediately tried to break down boxes
- Caught on very quickly with what the sponge and box cutter were for
- Tried to interact with all of the objects in the house

David

- Liked that you could speed up dialogue
- Trying to be organized, but so little room
- Kept trying to interact with the pickup button. Make them the same?

Weez

- Thought of a player piano?
- “Cute outfit, but it reminds me of a school uniform”

- Make her look too young to be living with someone in a relationship
 - Do like the palette though
- For first dialogue
 - Tonally different from what it reads
 - Especially with the ellipses
- Understood clearly the starting objective
- Likes the slower camera movement
 - Need for walking animation though
- Confused at the tileset geometry
 - Will change later, not really an issue as we will change soon
 - The three different views of the world, the geometry, and the player are all different -> confusing
- Seems like she was able to put a box in the spot another box was in
- Was confused on the idea of taped boxes need the cutter and untaped don't
 - In the absence of this, she organized the boxes a little more
 - Super unclear that some need the box cutter and some don't
- Ran into bug with picking up box from across the world
- Understood the green highlights well
- Thought "can i open them" but didn't try it on all of them
- Thought you needed to interact when you were holding it
- Was confused on where the table came from?
 - Kind of hinting at the actions happening immediately is confusing.
 - Animations?
- "Mind mapping of how and where i can place an object isn't making sense right now"
- Green outline vs highlighting confusing
 - Seems like green should just indicate what you will interact with
 - Having two different feedback systems might be confusing to players
- I only have two options for keys, but it isn't making sense in my head how to piece them together
- The ability to move around being blocked by the empty boxes is rough?
- The packing tape makes it look like a chest, making her feel like she needs a key
- "If it's in the scene, I expect it to have a function"
- Consistency in the point of view

- When carrying objects, maybe shrink them in size and place them in Ada's hands

Adi

- Ran into interact from forever away bug
 - More or less accidentally figured out non-taped stuff doesn't need cutter
- "Makes it clear what you're targeting" -> except that was the green square telling you where you will place
 - Will follow up
 - Without the cells, it feels a little more clear. The green and the overloaded view of the cells makes it a bit more confusing.
Having no cells while keeping the item highlighting makes it more clear.
- Should try to avoid level designs that allows the player to get themselves stuck
- Quickly got how to break down boxes
 - I think he overheard from others tho -> claims that's not true
 - So pretty intuitive for some, others not so much
- "This game makes me feel lonely"

Rando First Year

- Loves the art style and the fungus dialogue stuff
- Likes the camera slow-pan
 - A lot to get used to though
- Table hitboxes.... Again XD
- Understood opening un-taped boxes pretty quickly
 - But I did hear an "ooohhh"
 - Then understood to use box cutter on taped boxes
- Opened all the boxes before taking stuff out of any
 - Interesting
- Loved the memory dialogue
 - Well written
 - Got a good sense of both characters as well as the world
- More of like an "ahhhh okay" than an "aha! I figured it out"

- This is in the context of cleaning the tables and with breaking down boxes
 - Like “Oh, okay, I *can* do this”

My thoughts

I think we should get a new art stuffs for holding items

Art of the tables definitely looks wack as a one-square object

Kind of hinders interactions with other objects

Loads of overlapping

We should have a title screen, I think. With like, a play button. And a quit button. And controls. And stuff.

Need to have a deactivation for the highlighting of objects

God our controls aren't great

Maybe we should start with one of the boxes open if we want the box-cutter to start in a box

Or just start it outside

Maybe make the highlighting more severe

We could rotate objects based on the player's direction when they place the object

Would need 4 sprites per obj

COuld just click r to rotate?

The bug where you can pick stuff up from far away also occurs with interactions

Definitely base what you are going to interact/pickup based on hitbox and player directions

More rendering stuff

Box behind the walls when you pick it up

Veronica's Playtest Notes:

Person with Red Hair

- Defs running out of room to navigate around
- Defs having some issues with the controls
- Note: fungus does not appear to take up the fun screen
- Tried to cut the tables

Kyle

- Rye sucks at camera movement
- Making a nice box line
 - Definitely confused

- Tried to open a door when box was highlighted away from door and summoned box
 - found the box cutter and figured out how to interact :)
- Dirty table is gross
- He gives off confused vibes and everything that happened was chance.

Red hair but like natural not dyed

- Oh you can stab boxys to kill them? - that was something i did not know but something the player

David

- Likes the wax seal
- Can i zoom? I cant zoom
- Hit escape to do something and quit the game
- How do i get out of here?
- “I am going to take my own life”
- The knife just brings out violence

Jason

- Confused by the shirt on ada
- Camera is slow
- Did i win video game?
 - I might have told him about stabbing boxes and then he got very happy
 - Very very happy

“I am so glad i could murder in this game”

Friend of red hair by natural not dyed

- Trying to interact with kitchen
- Defs a lil switched up on controls
- Trying to organize stuff
- Grid a lil off (lowkey we should play items with in the grid originally)

Pulkit

- Where is the box cutter?
 - Oh i can open the boxes!
- Did not realize the boxes had stuff in them and then did not know how to interact with them
- Defs just a bit confused
- Really likes the dialog and art and story heck yeahhhhhhhhhhhh

Alfie

- Defs taking a minute to figure out what
- Keeping everything very organized
- The grid is weirdddd i keep noticing it on everyones playtest and ahhhhh
- Figured out stabby stabby
- Very organized
- Doing stuff step by step is how i would explain it
- Replacing stuff to ger in play they want
 - Grid interacts weird tho
 - Yeah like the grid is really bad
- Loves the art style

Weez

- Juno looks like me!
- “Old timey” gives western vibes
- Outfit gives school uniform vibes
- Too abrupt of a shift from happy to entering to worried
- The only one who appreciated slow camera
- Very confused about box cutter
- Did the box teleport
 - Need to unhighlight a box when u are not with it

Bright orange sweatshirt

- Organizing boxies
- Defs reference controls a lot

Red baseball cap

- Did the table teleport
- Everything is very clutter9

Dude who I am pretty sure is a professor but i do not know him

- Took a minute to figure out boxes
 - Got it eventually and referred back to controls
- Moving boxes around to get to untapped ones
- Mixing up controls
 - Also having issues facing the right tile
 - Found sponge but i dont think figured aout you can clean it
 - Eventually did

Black Sweater

- Move during dialog
- Keeps teleporitng boxes trying to open cabinet and everything

- Using the same key to try and interact i think?
- Defs did not figure the game out
 - Understood picking up kinda confused about interacting i think
 - Did not relise some boxes need to be stabbed to open
 - Seems to gunderstnd the game better but i had to tell them about hte box thing
- Accidentally figured out the knife this or at least it looks like a mistake

Red Jacket

- Stardew valley vibes
- Seems to be enjoying
 - Make a table pyramid
- Shocked by scifi elements in memory
- Likes the color coding of the characters
- Tried to stab table
- Tile highlights are not accessibly
 - Maybe have inner square move
- Did not get that the memory happened a while ago

Appendix E: Key Presentations Slides

[All-Faculty Pitch](#)

[Capstone Design Final Presentation](#)

[Thesis Defense Presentation](#)

Appendix F: Writing Documents

[Dialogue Database](#)

[Core Storyboard](#)

[Memories Storyboard](#)

[Character Arcs Storyboard](#)

[Character Backgrounds](#)

[World Background](#)

[Memory - The Last Goodbye](#)

[First Day Script](#)

[Spring Breakfast Script](#)

Appendix G: Level Design Documents

Original thoughts for indoor level(Fall Semester)

Updated designs for indoor level(Spring Semester)

Outdoor Level

11 Individual Research

11.1 Abstractions of the Heart: How to Design Better Romance

By Emily Horton

11.1.1 Abstract

Central to the story of *Mending Hearts* is the romantic relationship between the player character, Ada, and her girlfriend, Juno. A core goal of *Mending Hearts* is to provide the player with an authentic experience of romance with Juno. Romance in video games is not a new concept, but it is a topic that requires special care and attention to do well. Tried-and-true conventions used in the industry, while useful and often effective, are just the tip of the iceberg for what is *possible*. In this individual research topic, we explore why romance is a difficult experience to represent in games, how existing games typically represent romance, and some considerations that designers should take into account when they are starting out in crafting the experience of romance in a game. Finally, I enumerate the techniques I used in order to capture romantic gameplay in *Mending Hearts* and discuss results and future work.

11.1.2 Problem Statement

Among *many* of the tasks required of a game designer is that of translating real world phenomena into ludic systems that emulate the real world's rules and parameters. This translation is easier for some subject matter than for others. The systems that designers create can vary drastically in their accuracy of modeling the real world. Is the ludic system an abstract metaphor, a precise simulation of reality, or somewhere in between these poles? Regardless of where the designer's system lands on this spectrum, it will always be constrained by the machine that it, which will always have its limits in modeling analogue reality in a world of discrete math.

These limitations become especially problematic in games with non-player characters (NPCs). Videogames have a plethora of tried-and-true conventions to represent interpersonal communication between player and non-player, but these conventions tend towards abstracted representations of the designers' understanding of romance. A general consensus among players and designers is that these conventions, while used to create successful and beloved games, are

extremely poor models for actual human behavior⁴. Now, what if the core player experience of a game *is* the experience of human interaction?. As if modeling human behavior in our games' characters weren't challenging enough, games that focus on romance are attempting to emulate one of the most complex emotions of the human experience.

Despite my claim that designing romance in games should be an expert-tier challenge, romance-focused games (as well as romance systems and storylines within larger, non-romance-centered games) are far from uncommon and often reach the mainstream market. However, even these products that are applauded for their work suffer from the same limitations and inaccuracies of interpersonal NPC communication, *and* they carry additional cultural baggage that they have a responsibility to address. Furthermore, romance is a deeply personal experience that is neither understood nor defined in the same ways by all people, meaning that a designer runs the risk of creating a romance system that *only* makes sense to *them*, or on an industry level, makes sense to the *dominant demographic* in the industry.

Given all of the shortcomings and dangers we can identify, how should a designer craft the experience of romance in their game? In which ways can an algorithm (or multiple, intersecting algorithms) emulate love? What are the important metrics that a designer should consider, and what handles should the player have when romancing an NPC? And finally, what is it that our players want and need from our game to experience genuine feelings of romance?

11.1.3 Background

Common Conventions of NPCs and Romance

Romance systems in video games are not a new concept, and the problems associated with them are fairly well-known. There are an innumerable number of ways that romance can be designed within games, but there are several consistent concepts that are worth discussing in order to contextualize this problem.

⁴ Schreiber, Ian, Trey Alsup, Ethan Ham, and Yuri Bialoskursky. "Participants: A.k.a. Almost Human - Project Horseshoe." Project Horseshoe, 2017.
https://projecthorseshoe.com/reports/featured/Project_Horseshoe_2017_report_section_5.pdf.

Romanceable vs. Non-Romanceable

Typically, games with romance feature a distinction between NPCs that are romanceable and those that are not. A game may have these distinctions for a variety of reasons. In most cases, there is a heavy cost associated with making NPCs romanceable due to the dialogue writing, voice acting, and behavior scripting that is required for every NPC. Appropriateness is also a consideration when limiting dating options. In *The Sims* series, adult sims of any age may romance one another, teens may romance one another in a much more limited capacity, and children, toddlers, and babies have no romance mechanics at all⁵. Finally, developers may assume the taste of their audience and only focus their limited resources on creating romance content for NPCs that they anticipate their players will want to date; although there's merit in this decision from a project prioritization perspective, it can lead to disappointment at best and exclusion of entire sexual orientations at worst.

Relationship Score

Romanceable NPCs almost universally have at least one backing variable to represent their romantic status with the player character (and occasionally, other NPCs) as a score. A higher score in this romance variable indicates the romantic closeness between characters. This score variable is used to alter NPC behavior or relationship with the player. For example, in the game *Stardew Valley*, the player must have attained a certain number hearts with a romanceable NPC before said NPC will accept the player's proposition to start dating⁶.

Love Tokens

Given this romance score, the player will have a means to increase this score. Though the exact mechanics the player has at their disposal varies greatly, gift-giving is a common method used to increase the relationship score. Utilizing an existing item system, the player has an opportunity to give an item to an NPC, which directly increases their romance score. Another common mechanic is in talking to the NPC in question.

⁵ *The Sims 4*, Maxis 2014

⁶ *Stardew Valley*, ConcernedApe 2016

Dialogue Trees

Although not universal, talking to a character to romance them often incorporates some sort of challenge or skill check that measures each character's social skill level. In *The Sims 4*, relationship points are gained when a sim attempts and succeeds at performing a romantic interaction with another sim. Failing a romantic interaction in *The Sims*, which occurs based on sim moods and established relationship history, will instead decrease the sims' romance scores with one another. Instead of testing a stat or skill, many games such as (but definitely not limited to) *Dragon Age 2* (DA2) present the player with dialogue options to pick from, and the relationship score will change based on if the player picked an option that the romanceable NPC likes, regardless of a player character's virtual social skill.

Special Quests

Finally, romanceable NPCs may have quests that must be completed as part of romancing them. DA2 incorporates a combination of these mechanics. The player must ascertain what type of dialogue their potential partner will respond positively to based on their personality and their backstory. Once they've gotten closer to the NPC, they will bring up a story from their personal life that will lead into a quest the party must complete in order to continue romancing them.

Limitations and Complications

Scalability

Projects will always have a limit to how much content they can handle. Even projects that contain generated content will have a finite volume of possibility for the stories, characters, and interactions that happen in the game. Budget constraints, along with a general need to finish and release a game each require that the game must be marked finished at some point even if its possibility space is far from exhaustive of all possible player desires. While most of the following issues boil down to project scalability, the *significance* of each issue is unique enough to warrant enumeration and explanation.

Chance and Unpredictability

We simply cannot account for the myriad of factors that can affect human mood, emotion, and relationship development. Additionally, although we could use random number generation to simulate unexpected life changes (as well as variations in mood), it would be difficult to do so in a meaningful way because the world and the characters would need to mirror a potentially massive number of random events, meaning the scale of dialogue and content would quickly become unmanageable.

Player Agency and Expression

By the nature of the medium, videogames have limitations on what actions a player may do within the game. Interactions with NPCs are limited to whatever actions were programmed or whichever dialogue choices the writers wrote. These restrictions limit a player's ability to express themselves to NPCs in the world. A player is more likely to develop (or vicariously experience) feelings of romantic connection with an NPC if they have the agency to be themselves and successfully form a relationship with the romanceable NPC on their own personal terms⁷.

Chemistry and NPC Agency

Most NPCs do not have agency and do not simulate it. It isn't possible for the player and the NPC to create chemistry out of a dialogue tree. While it certainly is possible for a player to have the *experience* of romantic chemistry and for them to *perceive* their romantic interest as autonomous⁸, this is likely an uncommon experience for most players.

Procedural Romance

Relationships evolve in unpredictable ways over time. Though we might identify stages and general progression as trends in relationships, they do not exist as

⁷ McDonald, Heidi, Jennifer E Killham, Arden Osthof, and Jana Stadeler. "Designing Video Game Characters for Romantic Attachment." Essay. In *Digital Love: Romance and Sexuality in Games*, 183–221. Boca Raton, fl: CRC Press Taylor & Francis Group, 2018.

⁸ Enevold, Jessica, and Peter Kelley. "Approaching the Digital Courting Process in Dragon Age 2." Essay. In *Game Love Essays on Play and Affection*, 46–62. Jefferson, NC: McFarland, 2015.

procedural rules that must be followed. Most games with romance have a critical path to follow in order to romance a character. The designer (or the writers) must determine when you can interact with a romanceable NPC and when your interactions become romantic. Many games have the player progress from platonic interactions with an NPC to romantic interactions to dating and then to sex or being in a committed relationship as an end state. What of one night stands? What about casual relationships or relationships that are on-and-off? What about taking a break due to personal matters interfering with the relationship? Or what about fights and breakups? Even if a designer wants to include these relationship dynamics in the game, there will always be a specific trigger for them to happen; said differently, there is a predefined, procedurally-determined means by which a relationship evolves, while in reality, any one moment— a date, a fight, a benign trip to the grocery store— might cause a spark of deep passion or mark the start of a fight that exposes underlying structural issues with the relationship. Romance in games does not allow for these organic changes in relationships.

Quantifying Something Unquantifiable

All of these shortcomings that I have enumerated could all be summarized by one core problem: games (digital or physical) require specificity and quantification that simply doesn't exist. By definition of the process, game designers must reduce real-world complexity into something calculable by machines and digestible to players. The real-world experience of romance (and human relationships of all kinds) is deeply subjective and lives within the seemingly irreducible complexity that is the human psyche. Until humans have a complete and discrete model of the human emotional experience, the best chance that designers have to create an accurate and enjoyable romance system is to abandon simulation for well-thought-out abstractions of the human heart.

Culture and Representation

Given that romance in games will always be an abstraction, that means that designers must make decisions on which parts of romance they are interested in capturing. The facets of love that concern a designer will vary based on the intended player experience of the game that is being designed. Is the point of the game's romance system to allow the player to feel a sense of bonding and depth with NPCs that they are on an adventure with, thus immersing them further into

the game's world? Maybe the game is focused on the pursuit of finding a suitor, and the focus is then on mechanically capturing courtship rituals. Or, maybe the game itself is simply a vehicle to provide means-tested smut to players willing to master a game's mechanics. In any case, the designer isn't only deciding how to represent romance with values and algorithms, but they are also picking which aspects of real world romance are worth translating at all. As a consequence, designers are creating a system of romance *as they understand it* while they are representing their interpretation in a context that can imply real-world accuracy. This personal interpretation and subsequent representation of romance has many potential pitfalls that I would like to address.

Gender-based Perspectives

For better or worse, romance is incredibly gendered, and the different genders in this wide wild world have different expectations for romance. If the author prefers dating women, the female characters may be more varied and interesting than the men because the authors were better able to capture desirability and interesting traits for women than for men⁹.

Of course, a completely different problem also exists when female characters are paper-thin and exist for lusty sex appeal; the game is reinforcing that women are objects to be desired and won, and that they are not people to connect and share experiences with. This framing, and the surrounding strategy guides for winning romance or sex scenes with NPC's in games is reminiscent of pickup artistry, or the idea that sex with women is attainable by follow a specific set of guidelines to manipulate them into giving consent¹⁰.

Sexual Orientation and Alternative Interests

Romance options are often limited by gender. It's understandable to do so when the player's character has a strong sense of authorship, or when in-universe characters have their own orientations. However, games in which the player *is* the character (or in which players may roleplay as any character they can imagine)

⁹ McDonald, Heidi, and Amanda Lange. "Love on the Farm- Romance and Marriage in Stardew Valley." Essay. In *Digital Love: Romance and Sexuality in Games*, 59–67. Boca Raton, FL: CRC Press Taylor & Francis Group, 2018.

¹⁰ Enevold, Jessica, and Peter Kelley. "Approaching the Digital Courting Process in Dragon Age 2." Essay. In *Game Love Essays on Play and Affection*, 46–62. Jefferson, NC: McFarland, 2015.

must account for queer relationships. Additionally, players may want to date characters whom the developers assumed nobody would ever want to date, whether it be because they were not appealing to the developers or because the characters were of an alien or fantastical species.

A Psychological Perspective

Human emotions, especially that of love, are difficult to quantify, but game designers are not the first ones to attempt to do so. The field of psychology has made some progress in both defining a theoretical framework for human relationships as well as in measuring relationship closeness. The research that has been done in this field might offer designers some valuable perspective on how relationships might be captured in an algorithm.

Possible Quantification: The ERM and the RCI

Dr. Ellen Berscheid, an American social psychologist, first described the Emotion-in-Relationships Model (ERM) in 1983¹¹. Berscheid credits the creation of this model to the previous work of psychologist George Mandler. Mandler's Interruption Theory, which states that human emotions are caused by interruptions to expected patterns¹², was the basis for the ERM. According to Berscheid's model, a relationship is defined as a causal linkage between people who have interpersonal influence over one another¹³. Changes in any relationship (romantic or otherwise) occur when one person interrupts an expected chain of behaviors. In this case, interruption can be either negative and detrimental to the relationship or it may be a welcome, positive change that causes delight and strengthens the bond between people.

¹¹ Kessen, William, Andrew Ortony, Fergus Craik, and Ellen Berscheid. "The Emotion-in-Relationships Model: Reflections and Update." Essay. In *Memories, Thoughts, and Emotions Essays in Honor of George Mandler*, 323–35. Taylor and Francis, 2013.

¹² Mandler, George. *Mind and Emotion*. New York, NY: J. Wiley, 1975.

¹³ Berscheid. "The Emotion-in-Relationships Model: Reflections and Update"

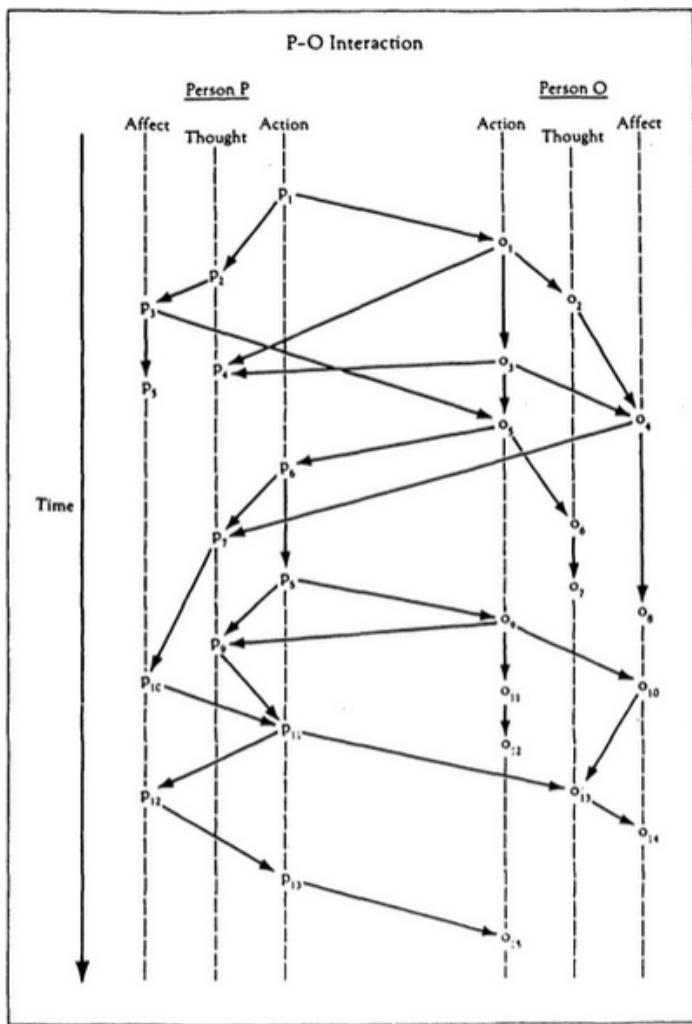


Figure 33: “The basic data of a dyadic relationship. Each person has a chain of events, each chain including affect, thought, and action, The events are causally connected within each chain (shown by arrows from one p to another or from one o to another) and the two chains are causally interconnected (shown by arrows from a p to an o or from an o to a p). The interchain connections constitute the essential feature of interpersonal relationships.”¹⁴

Because the ERM as proposed is by nature difficult to study in a controlled environment, Berscheid and others developed the Relationship Closeness Inventory (RCI) as a means of measuring relationship closeness¹⁵. The RCI assessed interdependence between people in a relationship using a questionnaire to determine the strength,

frequency, and diversity of activities and interactions that respondents participate in with their self-identified closest-person. The RCI was able to accurately predict romantic relationship longevity among participants based on their closeness score; those with higher scores had longer-lasting relationships with their partners

¹⁴ Kessen, William, Andrew Ortony, Fergus Craik, and Ellen Berscheid. “The Emotion-in-Relationships Model: Reflections and Update.” Essay. In *Memories, Thoughts, and Emotions Essays in Honor of George Mandler*, 323–35. Taylor and Francis, 2013.

¹⁵ Berscheid. “The Emotion-in-Relationships Model: Reflections and Update”

¹⁶. Additionally, these closeness scores were largely in-line with the respondent's personal assessment of their relationships' closeness.

My exploration into the psychology frameworks of understanding and qualitatively scoring relationships does not directly translate into a video game space, which exists in a controlled, temporally-suspendable digital space. However, I think that it serves as a useful starting point in designing an authentic and “universal” experience. How might a designer attempt to emulate these frameworks?

¹⁶ Berscheid, Ellen, Mark Snyder, and Allen M. Omoto. "The Relationship Closeness Inventory: Assessing the Closeness of Interpersonal Relationships." *Journal of Personality and Social Psychology* 57, no. 5 (11, 1989): 792-807. doi:<https://doi.org/10.1037/0022-3514.57.5.792>.
<https://ezproxy.rit.edu/login?url=https://www.proquest.com/scholarly-journals/relationship-closeness-inventory-assessing/docview/614311794/se-2?accountid=108>.

11.1.4 Methodology

Narrative Structure

Daily Schedule

Originally, *Mending Hearts* was going to operate on a daily schedule. Each day would pass by in real-time, and each day would end by calculating the relationship score for the day and use it to update the long-term relationship status. Most days in this system would be fairly routine and would be an opportunity for the player to interact with Juno and do activities around the house. However, depending on the extremity of the player's relationship score, some days would trigger a beat in a questline, which would represent the player's progression towards a breakup or a happy ending in which the ladies' hearts have been mended.

Weaving Questlines

I defined and outlined two story paths that would be intertwined together and served to the player based on the current state of their relationship system. Though there is a predetermined story and major story events, each player has the potential to experience the story in a unique manner for their playthrough. Each story scene is given a unique context based on what came before it instead of the player going down one single path.

- A player might play “perfectly” and the relationship smoothly progresses to a cozy, happy ending.
- A player has a rough time connecting with Juno and is clearly heading toward a breakup, but saves it at the last moment, thus surviving growing pains long enough to reach a happy ending.
- A player has a chaotic, back-and-forth experience in which the relationship could be heading in either direction, but will eventually settle down towards one conclusion.

Regardless of the player's overall path through the game, the intention with this design is that each playthrough is unique to that particular playthrough and mirrors the player's treatment of Juno.

Collecting Memories

Memories give a context to Ada and Juno's relationship history, their individual personalities, and a taste of their chemistry.

The Final Product - Story Vignettes

We significantly cut down on our original plans. The story of *Mending Hearts* is now told in a total of five vignettes, each of which will always be seen by the player. Four of the vignettes are grounded in time through the lens of a season, while the first vignette is a prologue that serves as both a tutorial for the player to learn the controls, as well as a means to set up the context of the game's relationship.

Quantifying the Relationship

Impressions

An enumeration that represents the personal, subjective experience of either an interaction or the relationship.

Mood

An enumeration that represents Juno's current mood. Her mood dictates how she responds to specific player interaction types. She might not be in the mood for flirtatious advances while she is feeling moody, but loves Ada expressing empathy towards her.

Changing Juno's mood during a vignette is a harder question to figure out. What events can possibly change Juno's mood? Can she change to any mood at all from any other mood? How often should she even be changing her mood anyway?

Attention Level

Is the player bothering or neglecting Juno? There is a quantity of interaction between the two partners that is both healthy and desirable; going outside of this range will make for a bad experience. Therefore, Juno has an attention level that represents the amount of attention she has been receiving from Ada. After a set amount of time, which can be configured in the editor,, her attention level will drop. With each passing of this time block, if her attention level is low, a negative impression will be added to the relationship tracker. Similarly, if the player interacts with her too often and raises her attention level to uncomfortable levels,

further interactions with Juno before she has had a time to cool off will result in logging negative impressions.

Short-Term Relationship Impression

In general, the relationship should neither tank, nor should it fall apart in the span of a few moments during one day. The total number of impressions that have been added to Juno from the player's actions around the house and direct interactions with Juno are averaged together at the end of the vignette to update her impression of the relationship.

Long-Term Relationship Impression

The long-term state of the relationship is determined by the average of the short-term averages. The original design was intended to work with a day-by-day system. The long-term average would trigger a story beat to occur on a given day if the long-term relationship score was either above or below specific thresholds. The long-term tracking as-implemented would serve as a loss condition if it gets too low before the start of each vignette, and it would also determine the specific ending of the game that the player would have achieved.

Juno Agency and Player Interactions

During each vignette, the player will have the opportunity to initiate dialogue with Juno by walking up to her and pressing the interaction key. When they do so, they are presented with a set of choices for *how* they want to interact with Juno. None of these interaction types are defined by an action or line of dialogue, but are rather abstract descriptions of the flavor of the interaction.

Interaction-Mood Combinations

A short-term impression of the interaction is created and logged with the Juno Manager class based on an impression level assigned to each interaction type. They represent how Juno would feel about a specific interaction while she is in any given mood. If Juno is Moody, she will respond very negatively to the player acting flirtatious. These combinations are similar to passing or failing a dialogue check, but instead of backing stats the combinations represent Juno's personality through what she responds well to. This is similar to the dialogue checks in *DA2*, which uses varying NPC emotional reactions when responding to similar lines of

dialogue the player selects¹⁷ – the player is trained to consider the emotional needs of Juno or risk damaging the relationship

Ignoring and Harassing Juno

What if the player won't leave Juno alone and instead interacts with her repeatedly in a short time? Or, what if they do the inverse and *never* interact with her? In order to add more context to the conversations and emulate a partner's actual reactions, I added a variable to track Juno's attention level. Initially I considered using a score-based approach here, but instead opted to define an enumeration of several attention levels and the requirements for moving up or down on the levels.

Instead of adding values numerically, moving up on the attention scale happens when the player interacts with Juno. A configurable number of interactions are required to increase Juno's attention level from where it is now. Meanwhile, moving "down" happens after a configurable amount of ticks, or scaled units of time, have passed since the player last played with Juno.

Dialogue Stitching – Design and Inspiration

I created unique dialogue lines for each interaction-mood pair. The intention was to give context-sensitive feedback to the player to communicate how the interaction was perceived by Juno, as well as *why* she perceived it that way. This distinction was a means to express Juno's personal emotional needs in every situation and to encourage the player to learn who she is as a person and what her needs are.

In the first playtest containing the dialogue interactions using interaction-mood pairs, players found it was awkward to just have inputs given to you without any social context. This lead me to consider more ways to expand the conversation and give it context. My idea then was to add a greeting and a question when the player interacts with Juno but before they pick an interaction method. This was a way to give players diegetic hints about Juno's mood *and* the current relationship state, while solving my context problem.

¹⁷ Enevold, Jessica, and Peter Kelley. "Approaching the Digital Courting Process in Dragon Age 2." Essay. In *Game Love Essays on Play and Affection*, 46–62. Jefferson, NC: McFarland, 2015.

In order to create variable and contextual dialogue, I used a method called stitching. Stitching, as defined by Ernest Adams in the international Game Developers Association (IGDA)'s book *Game Writing: Narrative Skills for Videogames*, is a technique used in games (and select other contexts) that combines disparate lines of dialogue or even individual words to create a cohesive dialogue experience¹⁸. Although Adams focuses on the audio recording and mixing techniques used for stitching in voice-acted games, his suggestions still apply to *Mending Hearts*.

A critical warning from Adams is to be certain that every line that is interchangeable has the *exact* same meaning as all of the other lines because the player needs to always receive the *correct* information about the game state¹⁹. To ensure this necessity in my writing, I followed a common design approach described by Ian Schreiber, a former professor of mine and author of the first book dedicated solely to the process of balancing games, appropriately titled *Game Balance*. The process is to create an anchor for the content you want to create in order to establish a baseline of how powerful or valuable something should be by default so that you can begin designing interesting variations that are balanced to that specific baseline²⁰. Though I wasn't balancing dialogue lines per se, I found it useful to follow this approach by writing very bland, almost robotic descriptions of Juno's reactions to Ada.

From that baseline, I created more human-sounding variations that used synonyms, changed the order of the emotional expressions within each statement, used more specific and visceral language, or combined these techniques.

¹⁸ Bateman, Chris Mark, and Ernest Adams. "Interchangeable Dialogue Content." Essay. In *Game Writing: Narrative Skills for Videogames*, 237–66. Boston, MA: Course Technology, 2008.

¹⁹ Adams, "Interchangeable Dialogue Content"

²⁰ Schreiber, Ian, and Brenda Romero. "Finding an Anchor." Essay. In *Game Balance*. Boca Raton: CRC Press, Taylor & Francis Group, 2022.

Dialogue Stitching – Implementation

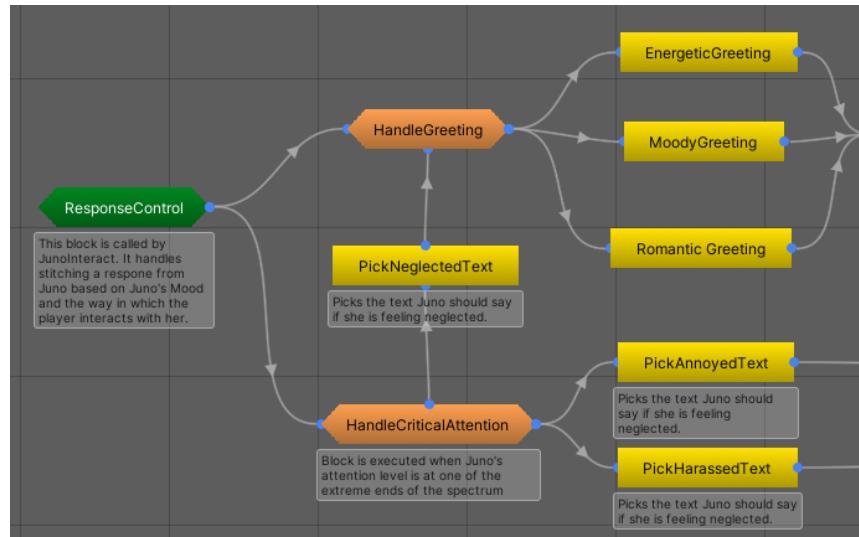


Figure XX: Conversation entry point in the Fungus flowchart.

The flow of conversations is managed using the Fungus plugin— more information on Fungus and its usage can be read in our [technical design section](#). Each conversation starts with a greeting. If the player has been bothering Juno with too many interactions, Juno will firmly tell the player to leave her alone and the conversation ends. Otherwise, Juno will respond with one of several greetings in a variety of mood-based flavors to communicate to the player how she might be presently feeling.

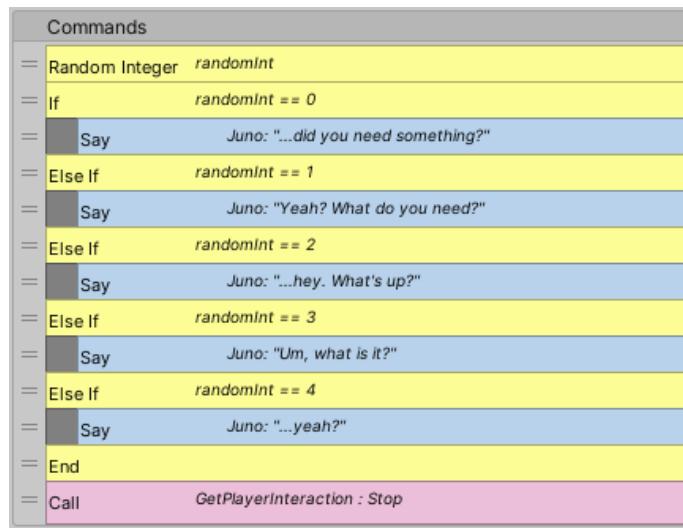


Figure XX: Examples of moody greeting questions Juno may ask.

Next, the player chooses how to interact with Juno. Because the player is taking the role of Ada, I picked four very different interaction types to be representative of Ada's range: Melancholy, Empathy, Intimacy, and Playfulness. Now, the game picks an appropriate response from Juno based on the pairing of interaction type and mood.

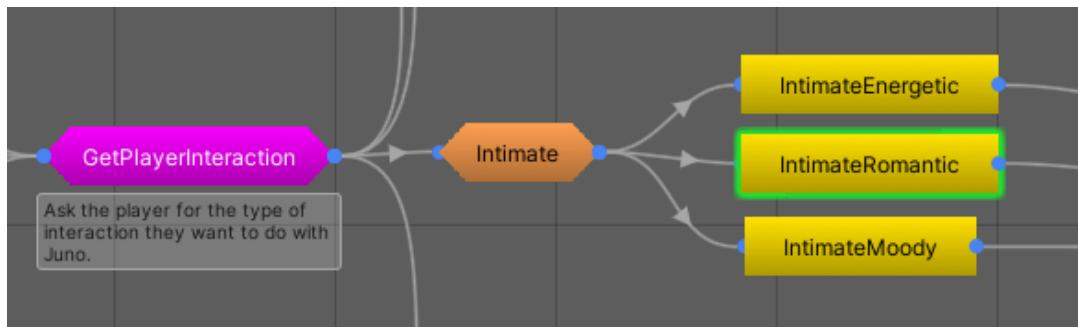


Figure XX: An example of the flow of picking an interaction-mood response.

Each of the blocks contain multiple responses to capture Juno's specific reaction to the player. My goal was to maximize variety while still capturing emotional nuance in Juno's responses.



Figure XX: An example of the available dialogue lines within a response block.

Relationship Evolution

Unfortunately, relationship evolution was not a concept that we were able to implement in *Mending Hearts* during this capstone experience. However, there were intentions to create a progressing relationship experience that was not on a completely predefined path and that would respond to the relationship history.

Changing Juno's Schedule

Originally, Juno was going to have a set of goals that she would be given for each day, and her behavior throughout each day would be her completing or attempting to complete tasks towards each of her goals. Over time, the player's interruptions (whether helpful or unwanted) would cause Juno's goals and the tasks she undertakes to complete them.

Changing Juno's Preferences

Juno's impressions of player interactions during each of her mood states could also vary with time. This change could represent a growth in understanding between Ada and Juno or even the personal growth of an individual character. For example, maybe Juno would eventually get tired of Ada acting melancholic while she herself was moody because instead of seeing the interaction as a bonding moment, she instead sees it as Ada always making everything about her. Alternatively, maybe instead of feeling a sense of apathy towards Ada while Juno is energetic and Ada acts melancholic, she learns how to channel her energetic mood to best help Ada while she's feeling down.

Changing the Relationship's Power Dynamic

Who is it that initiates conversations more? Which of the women is passive and which is active? These are dynamics that could fluctuate at different stages of the relationship and represent the growth and development of the relationship itself because the interactions are what form the experience of the relationship.

Other Considerations

Communicating Stat Changes

The dialogue will also stitch in Juno giving a very clear hint about what her current mood is. Although the player is not given any of the keywords that define Juno's moods, this direct statement, coupled with the mood-specific response, should give the player enough information to understand what Juno's current mood is and how she might respond to further interactions.

Utilizing Strategy for a Positive Impact

Although players can still strategize in order to win this relationship system, doing so is less transactional and requires the player to learn how to understand and respond to Juno correctly. Learning how to interpret, understand, and respond to a partner's ever changing *is* the core challenge of the game, and I would argue that training the player to think about how they can healthily and effectively meet their partners needs is a positive message that is independent from the pickup artistry taught²¹ in most romance systems.

11.1.5 Results

I don't think that I can completely and accurately speak to the efficacy of the romance system that I designed because *Mending Hearts* was not completed as intended. As previously stated, the relationship (and Juno's preferences) evolving over time was a key feature that was never implemented. Additionally, our scoring system and our dialogue system were built to work over the course of a longer experience, but the final build of *Mending Hearts* features less than twenty minutes of content, with most of this content being dialogue. Still, some playtester feedback was insightful and is useful to share for this research.

During our initial paper prototype, players were heavily invested in the role that they were playing. There were comments from multiple players that expressed intense emotion ranging from excitement that their piece of furniture was going to be well-received by Juno or anxiety that they weren't doing a good enough job.

Our first playtest that implemented Juno's moods and the player's ability to choose a method of interacting with her produced some mixed feedback. Players were generally in-favor of the abstract descriptions of the interactions they would be doing with Juno and did not seem to miss having exact dialogue. That said, at this stage in the game's development, dialogue stitching had yet to be implemented and Juno simply responded with the exact same line of text, which only depended on the mood-interaction-type pair. Players also said it felt as though Juno's mood changed much too quickly, and there was no indication when it changed.

²¹ Enevold, Jessica, and Peter Kelley. "Approaching the Digital Courting Process in Dragon Age 2." Essay. In *Game Love Essays on Play and Affection*, 46–62. Jefferson, NC: McFarland, 2015.

The rest of the feedback throughout our playtests focused on aspects of the game world, the story text itself, or our game mechanics. They were not focused on the relationship system or the relationship system.

11.1.6 Discussion

It seems that given enough context of the play space they are in, players are more than willing to become invested in the outcome of their connections to a character. Interestingly, player investment in the outcome of their interactions with Juno seemed much higher during our analogue playtest than our initial digital playtest featuring Juno. This discrepancy may be due to the fact that our team was running the paper prototype playtest and a response from Juno meant a response from a real person standing in front of them. There may have been an anxiety of being judged in real life.

Changing the behavior of an NPC character might require a more gradual, longer-term change. Although real humans might change moods multiple times throughout the day (or even contextually on a much smaller time scale), players perceive these state changes as too extreme to have happened in one moment or the next. Perhaps a different approach of mood changes would be in order for Juno; maybe there would need to be more neutral states that might have additional dialogue (or visual) hints of the *direction* her mood is going in.

I will note here that our results from playtester feedback and this discussion are definitely not generalizable— we have not discovered a universal solution to these problems, and a similar approach to ours may work within a different context or set of systems. Instead, my hope is that anyone reading the results of our work can both draw inspiration from our own efforts and have a prior warning of potential issues with approaches similar to ours.

11.1.7 Conclusions

Subjective experiences are difficult to capture in games. This is due to many factors within games as an industry and as an artform. The ultimate challenge boils down to the computational necessity of quantifying the unquantifiable. A love algorithm (or system of algorithms) does not exist; each relationship is a unique personal experience that is rooted in the ever-shifting context of an individual's life and surrounding world. Even within the field of psychology, scoring relationships is an incredibly limited tool that has mainly been used for confirming the results of

theoretical models, and not for producing a score of personal enjoyment—again, a subjective experience.

The limitations and complications do not mean that romance cannot (or should not) be represented within games; rather, the complications involved in portraying romance at the ludic level are deeper and farther-reaching than they may appear at a glance. Not only is a designer given the Herculean effort of quantifying the unquantifiable, but they are at a higher risk of creating a system that is exclusionary of their audience or promoting toxic ideas about love and relationships.

Because a designer is already being forced to imagine discrete variables that represent an analogue reality, they will be inserting their own personal views of the important metrics of relationships into the system unless they approach the design challenge deliberately. All good designers should have a player experience in mind; this requirement is especially true for romance. After understanding the intended player experience, the next question ought to be the means that the player can interact with the romance systems. If the game's *focus* is on the means of interaction, then a great deal of effort will be required to create something novel (or to improve upon an existing mechanic). When focusing on the system, a designer needs to consider what metrics of romance make the most sense for the player's experience.

Mending Hearts was concerned with providing the player the experience of being in a relationship. It was not interested in turning romance into a game with strategy nor was it interested in immersing the player in a larger world through delving into a character's lived backstory in the world. This intended player experience was my motivation to create a system base in real-world psychology. It is not an approach I would recommend to everyone, but the lessons I learned from this exploration reinforce the importance of defining, early on, what the player's experience should be *and* how they will interact with the relationship. Just as the interactions and interruptions within a relationship are what define its emotional experience, so too are our games defined by the moment-to-moment player interactions with the game that lead to unexpected, exciting, and joyful moments.

11.1.8 Future Work

Any one of the romance systems and interactions that I described in this research would be ripe for expansion and a larger, more-focused playtest to test how

players would perceive them. However, such tests are not a part of this project's scope.

Veronica and I have discussed working on a completely different version of *Mending Hearts* in the future that would focus on telling the story of Ada and Juno. If we were to move forward with this idea, we would likely use a dialogue stitching system similar to the one I created for this project.. This system would both allow for more responsive, context-sensitive storytelling and train players to pay attention to the emotional needs of their partner as a skill in order to "win" the game. Additionally, this system would allow for us to focus on telling the story without adding additional systems and mechanics. We might be able to fully realize our original goal of *Mending Hearts*: a relationship experience that evolves and is unique to the player's specific playthrough.

11.2 Knowing What You Want But Not How to Get There: A Look into Designing the Mechanics of Mending Hearts

By Veronica Vitale

11.2.1 Abstract

Mending Hearts takes a lot of influence from visual novels and cozy games. The game aims to captivate the player with the story of Ada and Juno while living a simple life in the countryside. In between the dialogues the player is able to move around and interact with the world. This research aims to look into different ways to leverage these interactions to keep the player exploring and entertained while not directly interacting with the story. As well as reflect on the process that our team took in order to come up with our end interactions.

11.2.2 Problem Statement

How can we leverage the simple interaction system of *Mending Hearts* to keep the player entertained throughout the game. This research aims to look into the design process of the interaction system and the steps we took to improve on our designs.

11.2.3 Significance

We will look through two designs for the mechanics of *Mending Relationships*, how they were developed, the research that went into them, and the conclusions from them. This will hopefully give a look into the design process and give future developers ideas of what to do with their games and how to go about it.

11.2.4 Background

We will look at two different possible systems of what the player could do in the game. Firstly, decorating the home, then secondly, interacting with the home.

Decorating the Home

At the beginning, we knew that we wanted players to be able to add their own flair to the designs while not straying too far from Ada and Juno's tastes. Gramazio's GDC talk, *Inviting Player Creativity Through Game Mechanics*, was a great starting point for designing this system. In her talk, Gramazio brings up that players need context and confidence to be able to begin. In our game we would give context through exposition of explaining the character's situations and memories giving insight into who they are as people. This gives the player an idea of how they should be designing the home without telling them outright what to do so they can feel free to design how they want.²² To give confidence, we would start small with just one room so as to not overwhelm the player and then move on to adding rooms as the player progresses.

Now that we have an idea of how to get the player introduced into the system, how will they get feedback and be encouraged to keep interacting with the system. Johanna Hall's paper *Expressivity of Creativity and Creative design Consideration in Digital Games* gives some insight into this. Hall found that some players use trial and error to solve problems in games, with this method leading to a sense of accomplishment when they find a solution.²³ At the time, we were considering having an end of day wrap up scene in bed with Ada and Juno to reflect on the relationship direction. We would use this time to rate the home and based on the home rating, Ada and Juno's interaction would change giving a player an idea if they were on the right track or needed to change something. Juul's *Without a Goal* paper looks into different types of goals in games and how games without explicit goals keep players engaged. Juul argues that *Sims 2* has no goal, the player can do whatever they want, but there are consequences for player actions.²⁴ Thus we should not have to force the player to actually decorate the house, but having the relationship with Juno get more or less strained based

²² Gramazio, Holly. "Inviting Player Creativity Through Game Mechanics." *GDC 2017*. Lecture presented at the GDC, April 24, 2022.

²³ Hall, Johanna, Ursula Stickler, Christothea Herodotou, and Ioanna Iacovides. "Expressivity of Creativity and Creative Design Considerations in Digital Games." *Computers in Human Behavior* 105 (2020): 106206. <https://doi.org/10.1016/j.chb.2019.106206>.

²⁴ Juul, Jesper. "Without a goal". In Tanya Krzywinska and Barry Atkins (eds): *Videogame/Player/Text*. Manchester: Manchester University Press 2007.
<http://www.jesperjuul.net/text/withoutagoal/>

on the player's actions should encourage the player to explore and see the different outcomes of their actions.

Interacting with the Home

The base idea of this system is heavily influenced by game play in *Stardew Valley* and *Animal Crossing: New Horizons*. In both of these games, the player can interact with items in the world through a simple button press. In *Animal Crossing* there are some items where the player can press a button and the item will either play an animation and or change its state. This interaction is overall just for flavor and does not have a large impact on the game. Both games have an equipment system where a player can hold an item to interact with another item to get a specific result.

In her talk *Rewarding Exploration with Collectables*, Leah Miller discusses how to use gatherables and collectables to encourage players to explore the world in the game. She explains that players will do what gets results over anything else. So to get the player to explore, there needs to be some sort of pay off. Miller argues that collectables, which are items with a limited amount that the player can achieve, and gatherables, which are resources that are relatively abundant, can be used to encourage the player to explore the world. When doing this, Miller suggests using these to teach the player mechanics and make sure that nothing goes against what the player has already learned. She also warns against having too many and having them out of the theme of the game.²⁵ This could be applied to *Mending Hearts* by having the player find the items they need to interact with other things in the game, such as a box cutter to open a box, or the food items the player would need to cook. This would also play nicely into the memory system where the player would find objects that give them insight into Juno and Ada's past.

In her talk about *Deathloops*, Dana Nightingale brings up some issues with teaching the player the game that might be relevant to *Mending Hearts*. Nightingale brings up how players were seeing *Deathloops* as a bad rogue-like²⁶,

²⁵ Miller, Leah. "Rewarding Exploration with Collectables and Gatherables." *GDC 2019*. Lecture presented at the GDC, April 24, 2022.

²⁶ Rogue-like: Genre of game where players attempt to achieve one goal and if they fail their progress is reset and they are thrown back to the beginning. The Genre is named after the game Rogue where players

when it was not even supposed to be rogue-like at all. Since *Mending Relationships* is combining visual novel elements and life-sim elements, there is a strong possibility that players might have different expectations for the game. *Deathloops* solved this issue by in part looking at how tutorials were done for their target genres and mimicking that in the game.²⁷ So when teaching the player the mechanics of *Mending Hearts* we should look into tutorials for games we wish to pull from, which are mostly games in the life-sim and visual novel genres.

11.2.5 Methodology

We will look into different things the player can do to interact with the world between story beats. Initially we will use other games and research to come up with a design for the system. After doing this we will implement and playtest it. We will then analyze both the playtest feedback as well as the implementation process so that we can understand the strengths and weaknesses of the system. After doing that we will determine changes to make to improve the system and repeat the process again.

It is worth it to note that scope will be a large limiting factor for this project. Because we are a small team without any external team members, each part of the project is being worked on by one team member, with some team members taking on multiple parts. This paired with a limited number of playtests and playtesters means that we will not always be able to get playtesting data. Though since our main goal is just to reflect on the design process and implementations of the interaction system, we should still be able to collect relevant information.

11.2.6 Results

What About Home Decorating?

Originally we took influences from the rating systems in games like *Animal Crossing* and *Style Savvy*, with a hope of creating a system that allowed the player to rearrange the house to fit Ada's and Juno's needs. This system would

would attempt to traverse to the end of a dungeon and if killed would have to start from the very beginning.

²⁷ Nightingale, Dana. "'Deathloops' User Research and User Experience Death Loop." *UX Summit*. Lecture presented at the GDC, March 21, 2022.

eventually track metrics like the cleanliness of the house, how it fit Ada's needs, how it fit Juno's needs, use of necessary items (bed in the bedroom, table in the dining room), and cohesion of the overall house. This rating would be tracked and then based on the results, would then affect the path of the story.

Implementation

I began to design a rating system that would be hidden to the player, so they would not be obsessed with gaming the system. This system would rate each room at the time, so that the player can be introduced to room decorating one at a time as well so each room can have its own requirements. It would take all of the items in the room, compare them to a list of required items for that room, then take the materials and colors used and log them. The last thing it would do is check the cleanliness of all items in the room and see if there was any clutter (i.e. cardboard boxes) and use this to determine how clean the room was. This would all be put into a struct that can be retrieved by other systems.

Why Not this System?

Eventually we would decide to scrap this system, mainly because of scope reasons. For placing items we used a grid, which had a lot of bugs. This was most likely because we were developing the grid alongside item interactions without much consultation about how the two systems would interact. On top of this we had a third system of Juno's AI that would crash if the player moved something that she was trying to get to or obstructed her path. We needed to scope down. Since the rating system, while important to the game, was not vital and depended on other bugged systems in the game we decided to look another direction. This also would allow us to have a more determined home layout so that Juno's AI could have more set points that it would go to.

What Was Salvageable?

From this system, we still had an interaction system with items where the player could press E and depending on what they were holding it would change what the item would do. We also had a memory system where certain items which, upon pick up, would trigger a memory that would give the player more insight into Ada and Juno's relationship.

What about interacting with the Home?

While placing items and moving the house around did not pan out, the system of actually interacting with items was working very well. This would also simplify keyboard inputs so that players would not be confused by having two keyboard inputs that do similar things. So this system works, but what are we going to do with it?

Implementation

When implementing this system, we used the same idea as the old interaction system, just de-coupled it from the grid. We kept any interactions we had before and decided that we would like to add one more major interaction to hopefully act as a proof of concept. For this interaction we decided to add cooking, since this has been something we have wanted to do with the game since the beginning and it would involve multiple steps. The players would have to find the ingredients, add them to the stove, and then cook them, and finally be able to eat the end product.

Since we were quickly approaching the end of the semester, we had few playtests left so when it came to designing the systems we needed to keep in mind that we would not be able to get a lot of feedback on how well players understood the controls. So we needed to go based on what we knew would work. Having a cooking system, would require some sort of menu that needed to be added so the player could see the recipe and add ingredients. For this menu, we used the same controls that the player had been using already for navigation, hoping this would help them get adjusted to the menu quickly. Also, the menu would have control pop-ups that would explain what each control would do. This is an inelegant solution, but will hopefully quickly on board players without taking too much development time.

Results

This system does work but the implementation was definitely rushed. Checking if the player is in distance to interact with the item took place on a trigger, whereas checking if the player had the items required did not happen until the player tried to interact with it. This meant players would see that they could interact with something but would not understand why nothing happened. While cooking was implemented, it was not tutorialized past control pop-ups, which players were

already confused about because they did not always do something. On top of this, players were unsure of where to get food from. Having an explicit tutorial for cooking and fixing the control pop-up hovers on all items would hopefully clear up this system and allow players to better interact with it.

11.2.7 Discussion

While decorating the home would give players a chance to express their creativity and learn about Ada and Juno's styles, it greatly increases the amount of variables the developers have to deal with. Making it very hard for a small team to execute, especially when they have an AI that has to be able to interact with the items in the home and we cannot guarantee what will be in the home or where. Though there is a lot we can do with just interacting with the home. Though with a large shift in mechanics it is important to try and salvage as much as possible. This also greatly cuts down on the scope since we have to take into account time lost from trying other routes. Even with these scope limitations, interactions in the home seem to be working well. With some clean up and expansion this system could be used to keep the player entertained and further the story.

11.2.8 Conclusions

Mending Hearts has been built on the idea of creating a cute, sentimental, and uplifting experience. We were able to stick to this idea throughout the iterations of the game. Though we were unsure of exactly what mechanics we would use to show this. We approached discovering this by choosing a direction to go in then working on implementing the entire system. This took up a lot of development time and greatly limited the amount of mechanics we could look into. In the future, attempting to make smaller prototypes of potential mechanics might be a better idea so that we can test things more quickly and fail-fast. In the future when developing a game like this, where we understand the emotional core we want but not how to get there, a rapid prototyping approach might be better to get a quick taste of what each would look like and hopefully set the game up for a fail-fast approach. The system we did end up with seems promising, having more time to experiment with it would hopefully allow us to expand the boundaries of it and integrate it more into the other aspects of the game.

11.2.9 Future Work

Currently the interaction system is loosely connected to the narrative and Juno AI. The player does have to collect certain memories and perform specific interactions to progress. Juno is able to interact with items in the home and assists the player in cooking a meal. Looking into how to better connect these aspects together would make them feel less like several pieces strung together and more of a complete experience.

We have also only touched the surface on interactions with memories and cooking. Having interactions that have to take place in a certain order on one item might be interesting to explore, as well as looking into how to better leverage collectables and gatherables in the game might help.

The player has a one item inventory, this was chosen because the idea was that they would be holding the item and not have room for more. As interactions get more complex and more items are added, it might become necessary to look into expanding this.

11.3 Storytelling Level Design -- Hanrui Zhang

11.3.1 Abstract

This project is mainly telling a story about Ada and Juno trying to get back together and return to the way their relationship once was after several years without seeing each other. This research is about using level design to make the feeling about each part of the story clearer to the players and they can feel the changes in their relationship through the whole story. I will look into other games (mostly 2D) that tell stories about relationships to see how their level designs work in helping the storytelling. I may also do some research about changing details of levels while players are playing to make the level fit the feeling of our story. There will be at least three levels from our scope and I hope to give players the right feeling about each story that happens in the scene.

11.3.2 Problem Statement

It's hard to deliver the whole story and world to players only with words such as dialogues and descriptions in video games. Level designs can also play an important role in storytelling in video games. So, what I would like to try to solve is how to design the level so that the scene can help players get the story and emotion better. Especially how to do level design for a story about relationships for this capstone project.

11.3.3 Significance

Imagine if a video game has an amazing plot but there are no details or design in the level that fit the plot well. In that way, players can only understand the story from the conversations and may get wrong ideas about the whole story. I believe that level design can contribute significantly to the narrative and storytelling in our project.

11.3.4 Background

The Importance of Everything: A Crash Course in Design-Related Analytics²⁸

²⁸Jim Brown, "The Importance of Everything: A Crash Course in Design-Related Analytics" GDC Europe 2014

<https://gdcvault.com/play/1020844/The-Importance-of-Everything-A>

Information is now everywhere. Understanding the basic framework of analytics will help me determine, for myself, what the value of this information really is and how it can be used to improve my design process. I have to think about what level I am designing, why am I designing this level and who I am designing it for to make sure I understand what feelings I want the audience to have from my level design.

In this talk, it talks about the importance of understanding analytics in design-related areas. I believe this talk would help me get better ideas on how to use the feedback from the playtests to improve my design.

What Happened Here? Environmental Storytelling²⁹

This lecture examines the game environment as a narrative device, with a focus on further involving the player in interpreting (or pulling) information, in opposition to traditional fictional exposition. They provide an analysis of how and why some games in particular create higher levels of immersion and consistency, and they propose ways in which dynamic game systems can be used to expand upon these techniques. The lecture presents the techniques for environmental storytelling, the key to the creation of game spaces with an inherent sense of history; game spaces that invite the player's mind to piece together implied events and to infer additional layers of depth and meaning. In addition to commonly-used environmental storytelling tools such as props, scripted events, texturing, lighting and scene composition, they present ideas for using game systems to convey narrative through environmental reaction. Environmental storytelling engages the player as an active participant in narrative; game systems that reflect the player's agency can do the same. The lecture will analyze existing cases and provide a framework for dynamic environmental storytelling in games.

This lecture can definitely help me understand how environment details can help on the narrative part and the techniques for environmental storytelling shown in this lecture can give me some basic ideas about how I should design and improve those levels for our project.

²⁹ Harvey Smith, Matthias Worch "What Happened Here? Environmental Storytelling" GDC 2010
<https://www.gdcvault.com/play/1012647/What-Happened-Here-Environmental>

Single Player Level Design Pacing and Gameplay Beats³⁰

This tutorial focused on pacing, gameplay beats and how to structure your level design narrative storytelling to a cohesive gameplay experience. It talks about gameplay beats and how to use them for level design storytelling, and how to pace my level designs.

Although this tutorial is mainly about level design in games that contain fights in them such as FPS and RPG games, I still think the pacing and gameplay beats part can help me understand the pacing in level design and use them to improve my levels on the level design storytelling part.

Level Design Workshop: A Narrative Approach to Level Design³¹

Level design can help players get better ideas about the whole story not only by some detailed assets placed in the scene but also by the pace of the level. This GDC Talk shows me many things about level designers from the pace of their level to an understanding for which assets should be placed where to tell a convincing story using physical space.

In this talk, it said that pacing storytelling methods can be divided into two parts which are allow for gameplay and tell the story and the allow for gameplay leads to player agency and tell the story leads to certainty of delivery and each method has its own player agency and certainty of delivery such as cutscene has high certainty and low agency but environment level design has high agency and low certainty.

Design patterns in level design: common practices in simulated environment construction³²

The author advances the premise that by examining the use of design patterns in architecture, computer science, and interaction design, a foundation can be

³⁰ World of Level Design, “Single Player Level Design Pacing and Gameplay Beats” August 18, 2015
<https://www.worldofleveldesign.com/categories/wold-members-tutorials/peteellis/level-design-pacing-gameplay-beats-part1.php>

³¹ Jolie Menzel, “Level Design Workshop: A Narrative Approach to Level Design” GDC 2017
<https://youtu.be/FhKjv7CPUqw>

³² Bacher, Denise, “Design patterns in level design: common practices in simulated environment construction” 2008-01-01
<https://dr.lib.iastate.edu/handle/20.500.12876/68969>

created to better characterize commonly occurring problems and solutions within level design. When multiple patterns are applied, the group becomes a language. This language can then be used as a means for creating novel levels as well as a lexicon for analyzing existing games.

There are several basic patterns in level design for player behavior in many different game genres like Action Games, Strategy Games, Role-Playing Games etc. In this paper, I learned that design patterns are solutions for common problems happening in a particular design example, and design patterns have a distinct template consisting of five key components:

- (1) the name of the pattern;
- (2) the context in which the problem is being examined;
- (3) the complete description of the problem, including the background of the problem and any ways the problem may occur;
- (4) the solution to the problem in the context described earlier;
- (5) any patterns that may work in conjunction with the current pattern.

The patterns shown in this paper are only a starting point for level design, there will be more patterns when I try to create new levels or evaluate existing levels with those patterns shown in this paper.

Level Design Patterns in 2D Games³³

Talks about six common level design patterns present in 2D Games. Although most of the patterns are about designing a 2D level that has combat elements which may not suitable for this project since Mending Hearts is a story-driven, no combat 2D game. However, the branching, guidance, and the pace breaking pattern could be useful for designing a level for Mending Hearts.

Stardew Valley

A very cozy and relaxing 2D game. This game may not contain a lot of storytelling in the level designs, but I think I can look into this game for the cozy and relaxing

³³ Ahmed Khalifa, “Level Design Patterns in 2D Games” June 10, 2019
<https://www.gamedeveloper.com/design/level-design-patterns-in-2d-games#guidance>

feeling for the level design in this project. I can also research in this game about how the top-down type 2D game looks like both indoor and outdoor to help me do the level design for this project.

To The Moon

A 2D game has no combat, party system. Instead of those, To The Moon is more about a story-driven game and the story about this game is mainly about discovering an old man's memory and fulfilling his dream. The story contains a lot of relationship elements that can help me get some ideas on how to design the level for a game telling a relationship story.

11.3.5 Methodology

I will look into other games (mostly 2D) that are story-driven and the stories contain or mainly about relationships to see how their level designs work in helping the storytelling and also read some papers about design patterns in level design to find out some design patterns for telling a story about relationships for this project.

I have learned that there are several basic patterns in level design for player behavior in many different game genres. Thus, I am trying to find a basic level design pattern for pacing and storytelling to tell a story about relationships and I can add my personal ideas and designs to the level with this basic pattern.

Last but not least, I will check on the feedback of the playtests to see if the level works and make improvements to my designs based on the feedback.

Indoor Level

For the indoor level, I chose to start with making sure what rooms do we want to have in the house not only based on if there will be some stories happening in there but also based on if they can make players feel cozy and relaxing and make the house feel more real to the players. Then I came up with a couple floor plans and searched for some cozy house pictures in real life to have a basic idea about how the house is organized and may build several 3d house models to give myself a better idea about how the house will look like. After that, I can quickly give several ideas for the indoor designs with the tilesets we have.

Outdoor Level

For the outdoor level, since the outdoor level design has a higher degree of freedom than the indoor level, I chose to read through our stories that happen outside to get a simple and basic idea about what we need outside the house. Then, I can figure out where the start and end point of the level is and can have some ideas about how to guide players to the places we want them to go. I can also get a basic understanding about personal characters of Ada and Juno and I can do some detailed environment designs that correspond to their personality to the outside level.

Limitation

Since we did not have any external artists in the team, the outside level is going to be designed and built with some free resources found on the internet and the resources I find on the internet may have a very different art style from the asset that is used for the indoor level. This may be a limitation while I do the designing and want to have some environment storytelling details in the levels. The scope of Mending Hearts may make me focus on the critical path design in those scenes and I may not have time to design some forks in the forest and hiking levels to let players explore due to our scope.

11.3.6 Results

After those background researches and designing some level samples, I notice that before let my environment detail designs give players better understanding of our story and the feeling about Mending Hearts, the levels should have appropriate guidance, flow and pacing to make players understand what is happening, what they should do, and where they should go easily at first.

Guidance in the Levels

For the indoor part, I decided to make a square area that is clearly protrude from the whole house to indicate where the main door is instead of just place a door in the house since the door asset we have used for each room in the house is the same as the main door and I do not want our players to be confused by that. Furthermore, I believe that putting the main entrance to the bottom of the indoor

level makes it feel more real when players get into the house from outdoor level than entering from a door on the top of the indoor level.

For the outdoor part, just like To The Moon, Stardew Valley and most of the 2D top-down type games, I used contrast color between grass field and road to indicate where the critical path is for players to go and design a fork that is shaped by trees to let players explore.

Pacing & Flow in the Levels

Beside guidance, the flow and pacing in level design are also very important and can help in storytelling. Based on my research about pacing in level design and other story-driven 2D games such as To The Moon, the basic idea is giving players some time to rest and understand what happened after a bunch of events and conversations. This can keep players immersed in the story and give a balance between engagement and rest in the game.

In the game To The Moon, there is always a simple puzzle for players to solve at the end of each small chapter. Those little puzzles will give players time to rest and understand the story the game shows to them in the last chapter. Thus, for my level design in Mending Hearts, I decided to add one more simple forest level with almost nothing happening in it between the outside house level and the hiking level to give some rest to players after many events and conversations happen inside the house and before the story while hiking. I believe adding a simple level like this could help balance the pacing and not make players feel like the story is happening too fast.

Environment Narrative in the Levels

After giving guidance in those levels and making them engaging to players, we finally get to the environment narrative design in Mending Hearts. Based on my research in environment storytelling, the basic idea is using environment detail to tell players what has happened in the scene. Since the limitation of art assets and scope, I decided to use the environment detail design to indicate something else such as the view after hiking level, Ada and Juno's personality, and give players some cozy, comfortable feelings.

Design Patterns

In Mending Hearts, while doing level design, I always read through the story that happens in the level and tried to make sure what Ada and Juno would do and where they should go in the scene at first. This can help me get some basic ideas about how the level looks and how I want to set the guidance in it and with those ideas I could make the basic view of the scene that is playable but without any environment details. Then, I started to add some details that makes the whole level more live and real to players and use those details to form some clues about our story between Ada and Juno.

Playtests

Since we did not have time to put all my levels into the playtest build, I did not receive much feedback about the outside levels. However, most players can more or less get the cozy feeling from the indoor environment. There is some feedback that the indoor space does not look very real since there is no bathroom and it's a little too big and they did not get the comfortable feeling from that part. However, after we put all the furniture, boxes and Juno and Ada's stuff inside the house, it does not make players feel too big any more, but the not very real part is still there since the furniture and boxes do not have any shadow with them.

One other problem found in playtest is that there are some design mistakes in those levels such as no collision at the edge of each level and this kind of mistake could easily drag players out of the immersive experience. Based

11.3.7 Discussion

Environmental narrative design can directly help the levels' storytelling in video games, it lets each individual player of the game fill in the gaps their own way but a level only focusing on environmental narrative still can not help players understand the whole story and get the right feeling about the game. Fine guidance, flow and pacing control in level design can all help the storytelling part indirectly. The environment storytelling will be meaningless if the level is lacking guidance and can not make players feel immersed which may let players not even continue playing.

11.3.8 Conclusions

Storytelling in level design is not only using environment details to give players some clues about the story, but also use level design to make players want to know what happened and what will happen in the story. The guidance, flow, pacing control etc can help players understand what is happening, what they should do, and where they should go and also keep them immersed in Mending Hearts. After the fine guidance, flow and pacing control, environment storytelling is a wonderful way to deepen the player's immersion in a virtual world.

After analyzing playtest feedback and the talk with R*, I have learned that beside those techniques in level design, it is also very important to make the level feel real to players to keep them immersed in the story. Before I consider how to make some detailed designs to give some clues about the story to players, I should make sure that there is not anything in the scene that could potentially drag players out of the immersive experience.

11.3.9 Future Work

I would like to make some of the environment details be able to change based on the state of Ada and Juno's relationship while the story progresses, such as adding more obstacles and grass on the main roads while the state of their relationship is going in a bad direction and adding more flowers on the grass field while they are going very well. I think this can give players more clues to the story and the relationship between Ada and Juno.

I did not do much design in the flow control part for Mending Hearts because of the scope, I think adding some forks in the hiking and simple forest level with some small events and dialogues on the end of each fork may be a great way to improve the levels.

In the game To The Moon, the levels of mountain, seaside cliff and the lighthouse feel very real and beautiful with the sky and sea layer behind and the camera movement. I tried to do the same things for our hiking levels but it did not work well. Thus, I would like to figure out how to make long distant views in 2D for our mountain level in Mending Hearts in the future.

After talking with R* and getting the playtest feedback, I would like to add something to my design pattern that I should make sure that the level does not contain anything that would drag players out of the immersion at first and after I finish the level. Furthermore, I think adding some shaders to the levels and all the objects placed in them to make it feel more real to players is a great direction to take in improving levels in Mending Hearts.

11.4 Bringing Autonomous Agents to Life

By Rye Ress

11.4.1 Abstract

In *Mending Hearts*, Juno is half of the relationship present in the game. Accordingly, she has a huge impact on how the player navigates the game and the message the player gets from the game. Juno should have wants and needs as well as the ability to go out and achieve those wants and needs. I looked into other games that have autonomous agents and how they achieved that on a technical level. I also attempted to implement Juno as an autonomous agent in *Mending Hearts* while following the norms and standards that major game studios employ. The catch is that Juno will be the reverse of what many autonomous agents strive to be. In most games, the design for autonomous agents is to be predictable, but Juno has been designed to replicate a human being and is much less predictable.

11.4.2 Problem Statement

Most of the autonomous agents present in today's games are simply there for the player to interact with. They act more as fixtures in the game than they do agents with agendas and the means to achieve them. I employ standard means of creating an autonomous agent while allowing them to be more than the classic, predictable, boring NPC (non-playable character).

11.4.3 Significance

The impact of this game is tied very tightly to the player's interaction with Juno. Juno is designed to be as close to a human being as possible to create that massive impact. Because the player interacts with Juno frequently and we want her to behave as humanly as possible, finding an architecture to match that behavior and get it functioning is a main pillar of the technical design of our game. I am not trying to shape a new architecture to introduce to the games industry, but I am trying to replicate and alter existing architectures based on what our game needs.

11.4.4 Background

Base knowledge

Before diving too deep into the world of autonomous agents and bringing one to life in Mending Hearts, some base knowledge and understanding of autonomous agents would be helpful.

What is an Autonomous Agent and what are the existing Archetypes³⁴

The term Autonomous Agent usually refers to the goings on of a single agent working in the space. It is used to include concepts such as sensing, motivation, and action selection/planning. All of these are touched on with Juno, our game's autonomous agent. Nareyek talks about how an autonomous agent has a goal, can sense properties of its environment, and execute specific actions. Normally, an agent will sense their environment, use that information to pick a goal, and take steps to achieve that goal.

Nareyek then goes on to describe the different archetypes of autonomous agents: Reactive Agents, Triggering Agents, Deliberative Agents, Hybrid Agents, and Anytime Agents.

Reactive Agents

Reactive Agents work in a call and response manner. When specific information is sensed, a specific action is taken. These are usually simple sets of if-then logic. The trade off is that goals are represented only by the rules stated and it is hard to ensure the desired behavior from the agent. Essentially, every possible state of the game must be considered and accounted for in advance in order to avoid holes in the logic. Large systems mean an incredible amount of rules, which makes it costly to encode these systems. While these agents react quickly, all they can do is react. They don't have long-term goals or reasoning.

Triggering Agents

These agents introduce internal states of the agent. This can be past information or something like a Finite State Machine (FSM) that is used in conjunction with rules to achieve longer-term goals. Because these still use rulesets like Reactive Agents, they still respond very quickly, but their use of states allows them to achieve longer-term goals in comparison. However, the same issue as Reactive Agents still exists: the agent is still based on hard-set rules that the programmer must think of every possible game state when encoding them.

34 Alexander Nareyek, "Intelligent Agents for Computer Games," AI Center, 2002,

<http://www.ai-center.com/references/nareyek-02-gameagents.html>.

Deliberative Agents

This system uses the world model's information to build a plan that achieves the goals of the agent. These planning systems are often identified with the agents themselves. The base of these planning systems is based on the current world state, the desired world state, and a set of actions that can move the current state to the desired state. The whole sequence of actions that would bring a current state to a desired state is the plan of the agent. This can be made more difficult if the agent only has incomplete knowledge of the world.

Deliberative agents are great for achieving longer-term goals with the sets of actions that makeup their plan. There is no longer a need for extensive rules as the agent can re-define its plan the moment its current plan is no longer applicable. The issue existing here is how slow these agents tend to be. Every time the world state changes, the agent's plan must change to achieve their desired world state.

Hybrid Agents

As their name might suggest, a hybrid agent combines both reactive and deliberative agents. They still look at the world state and come up with a plan to change it to a desired state for higher level planning, but the minor refinement and alternations for single steps of plans are left to a reactive component. This does need a well-defined line between higher-level planning and hard-coded reactions. This is still considered too slow for most modern computer games with their constantly changing environments.

Anytime Agents

The idea behind these agents is to have a continuous transition from reaction to planning. Ideally, there will always be a plan available for the agent to execute. The way this is achieved is by improving the plan as the agent moves through its steps. When it is time to execute the next action, the agent improves its current plan until a predefined calculation limit is reached, then the next step is taken. The bounds of the computational limit defines who the agent will behave. The lower the limit, the closer this agent is to a reactive agent, the larger the limit, the closer this agent is to a deliberative agent. The biggest positive is that because the plan is being adjusted as the agent goes, there is no need to recalculate an entire plan anytime the world state changes.

More Autonomous Agent Archetypes³⁵

Pedamkar also attempts to break down the various archetypes of autonomous agents. Pedamkar defines them as Simple Reflex Agents, Model-Based Agents, Goal-Based Agents, Utility Agents, and Learning Agents.

Simple Reflex Agents are described in the same manner as Reactive Agents from the above article. Model-Based Agents are defined in a similar manner as Triggering Agents from the article above.

Goal-Based Agents are similar to the Deliberative Agents from the article above in that they understand the world state and strive to bring it to a desired state. The main difference is that this article's Goal-Based Agents don't have a set of actions and iterate through them. Instead, they just take one action at a time and re-evaluate the difference between the current world state and the desired world state.

Utility Agents are described here as a way to choose between possible solutions to a problem. They take into account the possible solutions and possible alternatives based on the world state and perform a cost-benefit analysis of them to select the best one.

Learning Agents are described here as being able to learn from their past experiences. They normally start with very little knowledge and build that knowledge up over a long period of time of working in the environment. This article gives four key components that allows these agents to learn: a Critic that evaluates performance, Learning Elements that the Critic gives as means to improve, Performance Elements that decide on the action based on the Learning Elements, and the Problem Generator that takes input from the other components to suggest the best possible actions. These sound closely related to machine learning, which is out of scope for this project.

Pathfinding Algorithms

After looking at some background descriptions and archetypes of existing autonomous agents, I started to look into some pathfinding algorithms to find one that suits Juno and *Mending Hearts* best.

At first I did this research with a grid system in mind because earlier versions of the game used a grid for item placement and Juno's movement.

³⁵ Priya Pedamkar, "Intelligent Agents: Top 5 Types and the Structure of Intelligent Agents," Intelligent Agents (EDUCBA, November 8, 2021), <https://www.educba.com/intelligent-agents/>.

After some design changes, we swapped Juno's movement to a node based graph so I re-looked into many of these algorithms. Realistically, there was no difference between the two.

Depth and Breadth First Algorithms, Uniform Cost Algorithm, and Dijkstra's Algorithm³⁶

Miller and Ranum talk about various different tree and graph traversal and search algorithms. The algorithms discussed in the article include: Depth-first search, Breadth-first search, Uniform Cost search and Dijkstra's Algorithm.

Having some experience with most of these searching algorithms, I knew immediately that Depth-first and Breadth-first search would not be what we wanted because they don't take into account the cost of the edges between nodes (in our case the distance from the current node to the destination node). Uniform Cost search starts to take that into account, but falls into the issue that it explores nodes in every "direction," making it slower than what we would need.

The next pathfinding algorithm Miller and Ranum talk about is Dijkstra's Algorithm. I have implemented this algorithm before so I knew some parts of it, but I needed to look more into the costs and benefits to know if it was what would fit best in our game. The upside is that it would always find the shortest path. The downside is that it checks every node of the graph, making it very computationally costly.

Greedy Search and A*³⁷

Greedy Search and A* are closer to what we would want, taking in the direction of the edges between nodes (the estimated distance from the goal node). The idea here is to explore nodes closest to the goal node and traverse the graph that way. Greedy Search comes very close to our ideal, but in its worst case, it just turns into a Depth-first search, which we already discarded.

Additionally, it tries to move towards the goal even if it's not the right path. Since

³⁶ Bradley N. Miller and David L. Ranum, "8. Graphs and Graph Algorithms," in Problem Solving with Algorithms and Data Structures Using Python (Decorah, IA: Brad Miller, David Ranum, 2014).

³⁷ Amit Patel, "Introduction to A*," Introduction to A* (Red Blob Games), accessed March 2022, <http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html>.

it only considers the cost to get to the goal and ignores the cost of the path so far, it keeps going even if the path it's on has become really long.

Last up then is A*, which will never explore the same node more than once while taking into account the distance of the current node from the goal node and the distance of current node from the starting node, using the relevant edges. This solves the same issues Greedy Search does without running into the worst case scenario of turning into a Depth-first search.

The question now turned from what to implement to how to implement the algorithm. The biggest question was how to find the neighbors of a node in the graph and how to find where the player is, since the player is independent of the node graph used for the autonomous agent's pathfinding.

The neighbors of a node in our graph are based off of line-of-sight. Essentially, if you can draw a line between two nodes without hitting any level geometry, those two nodes are neighbors. While we could define the neighbors of a node during run-time with raycasting and calculations, we decided to pre-fill a neighbors list for each node in the Unity editor to avoid expensive calculations at run-time. The downside of this is that it shifts the work from the programmer to the level-designer. This also means that instead of coding a way to detect neighbors once, each level needs each node to have a predefined list of neighbors. We did it this way to avoid hard-coded values or using raycasting calculations on the start of the game. Also, it allows us to have more freedom and agency with the node-graph. By using the editor to assign neighbors, we can be more specific about which nodes are neighbors and which aren't than we could with code.

To find the player, we implemented a relatively simple solution with game logic. We find the node closest to where the player is and give that to the autonomous agent as its node to move to. Once there, we check if that is still the player's closest node because Juno could arrive at the node given, but Ada could have walked away. If it is, we found the player! If not, we have to re-do the first steps of calculating the player's closest node and giving it to the autonomous agent for pathfinding.

Autonomous agents architecture

Now that we have a base understanding of autonomous agents and their archetypes and pathfinding algorithms, we can move on to the next step: The architecture of getting our autonomous agent to make informed decisions based on the world state.

Decision Making Loop³⁸

Balduccini and Gelfond give a possible solution for how an autonomous agent might select an end goal and move towards it. They name this solution the Observe, Think, Act Loop. This is exactly what it sounds like: the agent uses its sensors to take in information about the current world state, uses that information to compute an end goal G , breaking it down into a sequence of actions (a_1, a_2, \dots, a_n) that would achieve that goal, and does those actions in order. It is important to note that the agent should observe the world in between each action to see if they have their desired effect, but does not always necessarily have to. At the very least, it should check the world state after the set of actions has been completed to see if they had their desired effect on the overall world state.

Balduccini and Gelfond say the key to being able to compute an end goal and its associated actions is a transition diagram of the game world. This is a table of sorts that tells you what actions bring about which changes in the game world. The article uses a very simple electrical circuit diagram as an example. In that diagram, if you were to open or close the switch associated with bulb one, it would either turn on or off based on its current state.

The autonomous agent would observe the world, compare the current world state to its desired world state, and use the transition diagram to populate the set of actions to achieve the goal that is the desired state. The article talks about how the agent having perfect knowledge of the world isn't a total necessity. The agent can draw conclusions based on the limited information they have to fill in the gaps of their knowledge about the world state, but it all depends on what sensors the agent has at its disposal.

Balduccini and Gelfond mention that unexpected observations could cause an issue for this logic loop. These unexpected observations could be

³⁸ Marcello Balduccini and Michael Gelfond, "The Autonomous Agent Architecture," *The Autonomous Agent Architecture*

(Association for Logic Programming, March 2010),

<https://www.cs.nmsu.edu/ALP/2010/03/the-autonomous-agent-architecture/>.

observations that contradict the expected effects of the actions the agent has taken. Using the electrical circuit as an example: if the agent opens the switch associated with bulb one to turn it on, but the bulb does not turn on, what should the agent do next? When this happens, the agent should investigate the objects it interacted with during its actions to find the cause of the issue. Again going back to the example, it is possible that the bulb has blown out or the battery in the circuit has blown. In order to be able to investigate these world objects, the state transition diagram needs to be detailed enough to tell the agent the possible issues when a set of actions does not achieve its desired goal.

Low-level Architecture³⁹

Anthony talks about agent architecture in the low-level context of game engines. As such, the minute details won't be as important to me, but there are still key concepts that I can apply to the autonomous agent in *Mending Hearts*.

This article has a similar idea to the one above for how agents should look to make choices in the world. They should use a communication method to talk to the engine and understand the world state. Then, they should use a heuristic to compare possible actions to alter the world state, then take the actions it decided on.

The article talks about possible components that an agent should have and supporting classes that an agent needs to take actions. The article recommends a separate action class that represents the actions and operations an agent can make in the world. It also recommends a state class to encapsulate the state of the world as it currently is. The state class would be a container for information about the world state and be passed in between agents so that they can make decisions about what actions to take.

The article also loosely talks about a “Referee Engine” that the agents would interact with. From the article, it sounds like this is an abstraction above the game engine that the agents would get their information from and pass their information into.

An example UML diagram is given to represent what the agent would need and how it would connect to the engine.

³⁹ Lisa Anthony, “Agent Architecture,” Agent architecture, May 2001,

<https://www.cs.cmu.edu/~ianthony/classes/SEng/Design/agent.html>.

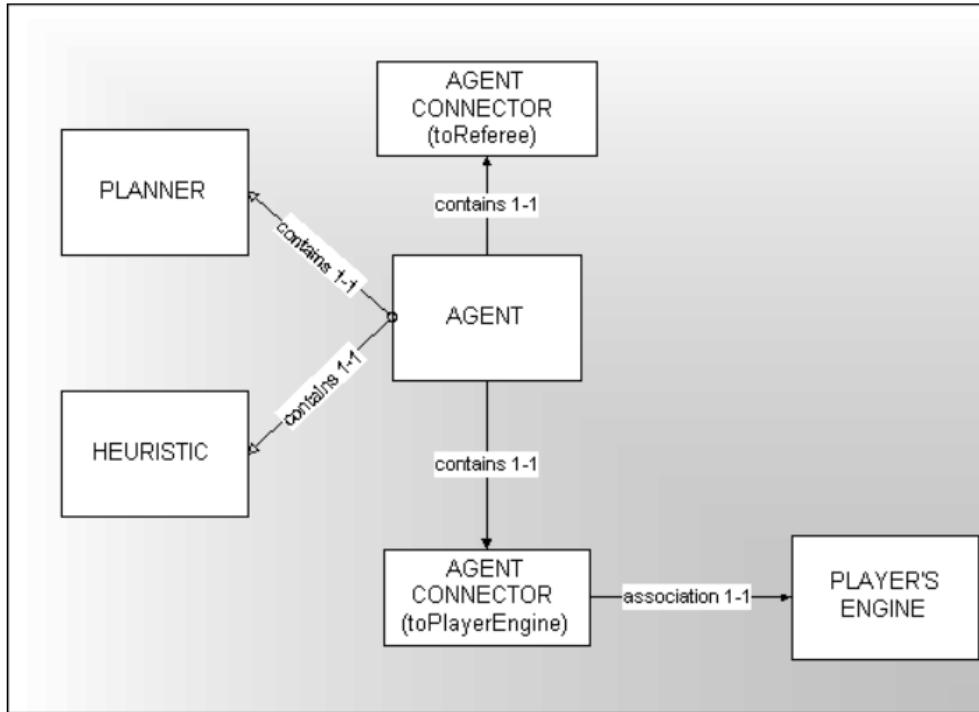


Figure 34: UML diagram of an example for an autonomous agent architecture

11.4.5 Methodology

Process

As I mentioned earlier, I had a three-phase plan to implement the overall architecture of our autonomous agent, which I will refer to as Juno for the rest of the paper. The first phase was just to get the basic movement and path-finding down. The second phase was to get Juno's interactions with the world in. This included picking up and placing items, interacting with items, and talking to our player character, Ada. The third phase was to combine the first two with an “intelligent” system to set goals and achieve them step by step. We want Juno to be more of a Hybrid Agent where there are predefined goals that she has, but based on the world state, she chooses which goal to execute.

Overall Architecture

The overall architecture consists of one manager class that tells two function-based classes what to do. The idea was to have a separation of concerns so that I can have one class that only deals with movement and one that only deals with interactions with a manager class that calls methods of the two in

order to achieve the goals Juno has. The classes are aptly named: JunoManager, JunoMovement, and JunoInteractions. The JunoManager class would have a list of Goal objects that would serve to represent the goals Juno wants to achieve during the game.

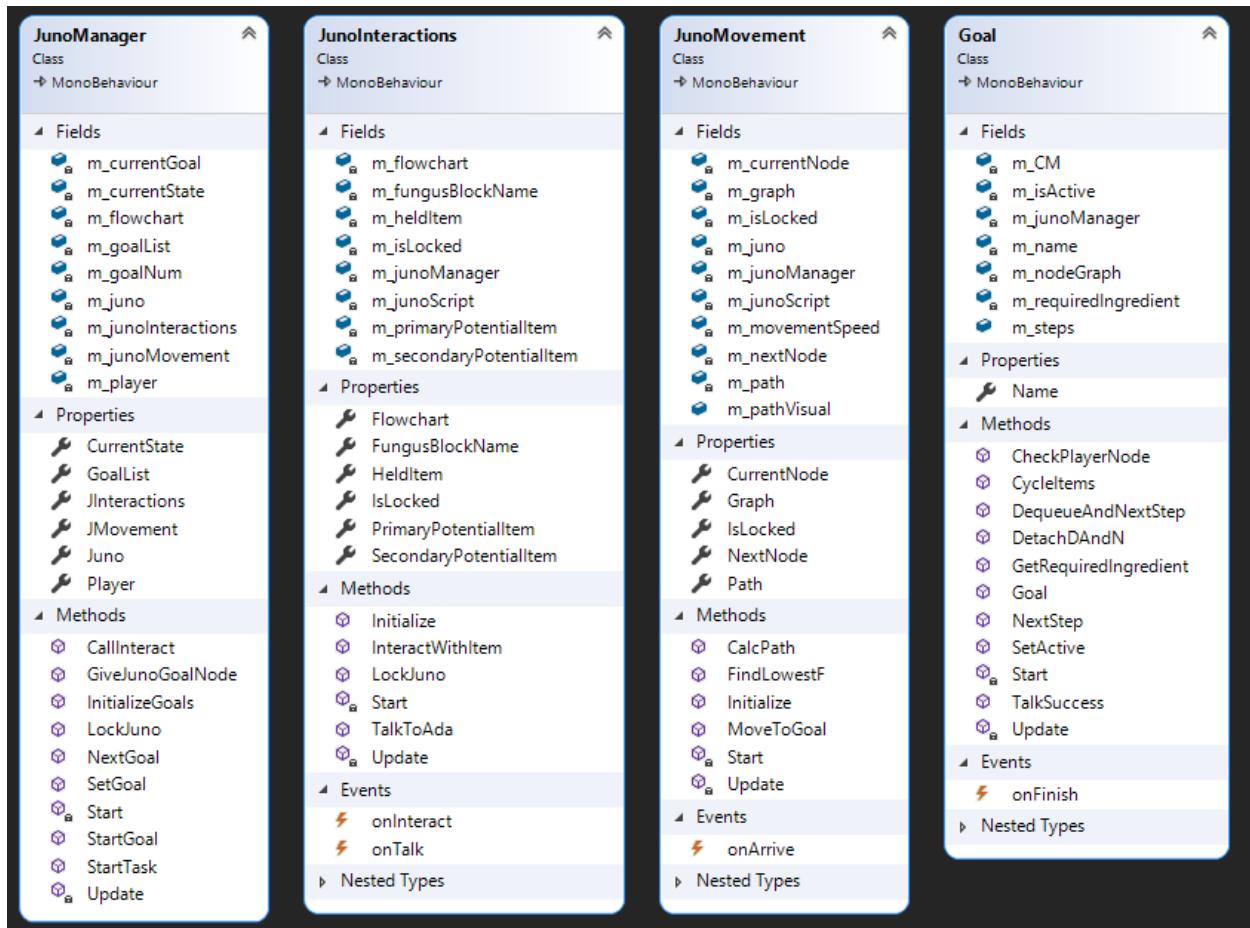


Figure 35: Class Diagrams for all Juno-related classes

JunoManager

The JunoManager class creates an instance of both the JunoMovement and JunoInteractions classes. The main objective of this class would be to compute the logic to set a current goal from a list of goals and iterate through the steps of that goal. Since this class has access to the function classes, it would be able to easily call methods from them to achieve the necessary steps. This class is also the only class that the other systems in the game interact with, which is why it holds an instance of both functionality classes.

JunoMovement

The JunoMovement class handles everything to do with Juno moving through the world and nothing else. The main methods of this class will be to calculate the path to a goal cell (later node) and to iterate through the path while moving Juno along it.

JunoInteractions

The JunoInteractions class, similarly to the JunoMovement class, handles all of Juno's interactions with the world and nothing else. Ideally, we want Juno to be able to do anything that Ada does. As such, it needs methods for picking up and placing items, interacting with items, and talking to Ada.

Goal

The Goal class is going to be used to represent a goal that Juno wants to achieve in game. It would need a collection of methods that makes up the steps to the goal. This collection would need to be iterated through easily, leading me to something like a List or Queue. The only issue I see is storing methods becomes an issue, even when delegates are involved because of the need for different method signatures depending on the end goal. When instantiated, this class would require a parameter to get passed in that would tell it what the end goal is. This parameter would get passed through a switch statement to populate the collection of methods that define the steps for the goal. This class would also need a means to iterate through the collection of methods and call them when ready.

11.4.6 Results

On a whole, I mostly stuck to my plan while developing Juno. The overall architecture remained the same: there is a JunoManager class that has an instance of the JunoMovement class and the JunoInteractions class that each take care of their own respective sections of Juno's behavior. The JunoManager class handles all interactions with other systems in the game and relays that information to the JunoMovement and JunoInteractions classes. This way the other game systems only have to interface with one script instead of all three. The JunoManager also handles instantiating which goals Juno will set out to

accomplish given the current season by finding what season we are in and adding the appropriate goal to the list of goals.

The JunoMovement class turned out to be a tad more complex than I intended. The path finding algorithm that I implemented was A*. When we still had a grid-based system, this fit our needs perfectly. Implementing this was relatively simple and fast. The issue came with moving Juno along the path. At first, I thought of a simple while loop that would add a force in the direction of the next cell (node) until there were none left in the path. However, while loops don't play very nicely when called every frame. To find a solution, I made a method that moves Juno towards the next cell (node), updating its current and target cell (node) as it went, as long as there is something in the path. The pseudocode for this method is along the lines of:

- 1) If there is nothing in the path
 - a) Return
- 2) Set my current node to the closest node and my target node to the node at the top of the path stack
- 3) Move towards the target node
- 4) If the target node is the same as my current node
 - a) Pop the target node from the stack and reset my target node to the top of the stack.

The JunoInteraction class was relatively simple to complete. I just needed to allow Juno to interact with everything that Juno could interact with. The only real issue here was the dilemma I was faced with when coding. The way that the player interacted with items was by passing in the script on Ada on the item's interact method. This meant that I had to either make the Ada and Juno script come from a single parent class and use polymorphism or make each item have two separate methods for interacting, one for Ada and one for Juno. I chose to write two separate methods. I didn't think it made sense to have them inherent from a single parent class because of how different the two were in terms of what info they needed and their functionality.

The Goal class is essentially a glorified information container. This class holds what the name of the goal is, passed in as a parameter, and the breakdown of individual steps required to complete the goal. While we originally wanted Juno to be more of a Hybrid Agent, she turned out to be more of a Triggering Agent. She does react to the state of the world with predefined goals. However, the way she reacts is very set in stone using the StoryManager. As its name suggests, it

handles everything that touches moving the story along in the game. The goals Juno has are already broken down into steps defined inside the Goal class. These are defined with a big switch statement based on the string representing the name of the goal. This was done by holding a queue of actions. Actions in C# function similarly to a delegate. However, unlike a delegate, they don't care about parameters or return types. This means I can essentially have a queue of any methods I choose. Hooking up the Goals with the events completed by the JunoMovement and JunoInteractions class was completed with a system of events and delegates. These allowed me to hook up methods in the Goal class to specific moments in the game, for example when Juno arrived at her location or interacted with an item.

```

case "TalkToAda":
    // steps:
    // go to ada
    // check if she is still there
    // talk to ada
    m_steps.Enqueue(() => m_junoManager.GiveJunoGoalNode(m_nodeGraph.FindClosestNode(m_junoManager.Player)));
    m_steps.Enqueue(() => m_junoManager.JInteractions.TalkToAda());
    break;
case "Cook":
    // STEPS FOR COOKING
    /*
     * Check what recipe is current on the oven - call get recipe
     * Check what ingredients the recipe needs
     * Check what ingredients are in the recipe right now - call get ingredients
     * see what ingredient is needed and return it
     * ----- ALL THE ABOVE STEPS ARE IN THE GETREQUIREDINGREDIENT METHOD -----
     * go to fridg
     * interact until the ingredient in your hand is the one needed
     * go to stove
     * interact
     */
    m_steps.Enqueue(() => GetRequiredIngredient()); // step one, get the info you need
    m_steps.Enqueue(() => m_junoManager.GiveJunoGoalNode(m_nodeGraph.FindNodeByTag("Fridge"))); // Step two, go to the fridge
    m_steps.Enqueue(() => CycleItems(m_requiredIngredient)); // step three, get what you need from the fridge
    m_steps.Enqueue(() => m_junoManager.GiveJunoGoalNode(m_nodeGraph.FindNodeByTag("Oven"))); // step four - go to the oven
    m_steps.Enqueue(() => m_junoManager.JInteractions.InteractWithItem()); // step 5 - interact with the oven
    break;

```

Figure 36: Code to add new actions to the queue of actions in the Goal class

11.4.7 Conclusions

While progressing through the stages of creating Juno were fairly uncomplicated, we did run into some issues while developing our Autonomous Agents. The largest issue we encountered was having to refactor a lot of the code for movement and interactions. The reason we needed to refactor the code is because of the swap we made from the grid system to the node system. After one of our playtests, we sat down and had a conversation about re-scoping our game and what we needed to have in it to bring out the core experience. As a result of this conversation, we dropped our grid system and moved Juno to a node system instead.

This change meant that I had to go into the JunoMovement class and change just about everything there. While this wasn't a massive overhaul in terms of what was changed, it was a lot of code that had to be swapped to our new system. Mainly the methods for A* and for moving Juno through the path.

Since the JunoInteractions class had almost all of its methods based on the old grid system, the change meant I had to refactor the way Juno interacted with items in the world and Juno as well. Since we got rid of the idea of picking up items and placing them on the grid, I was actually able to disregard those previously needed methods. I only had to focus on the Interact and TalkToJuno methods. While these were previously based on whether or not Juno was next to something on the grid, they were now based on if Juno was colliding with the item's outer hitbox.

11.4.8 Future Work

Overall, this architecture for our autonomous agent works for our game, but can definitely be improved upon. The biggest part of this architecture that I would improve on is how I strung together the JunoManager class with the JunoMovement and JunoInteractions classes. While it is nice for the JunoManager to be the only script that interacts with the other game systems, I think it might be worth looking into connecting the JunoInteractions and JunoMovement classes directly to each other. This way if the interactions class wants the node graph from the movement class, I can grab it directly instead of having to go up to the manager class and back down to the movement class.

I would also love to be able to change Juno from a Triggering Agent to a Hybrid agent. We had a small start to this when we were still using the grid system. Veronica had developed a system that would analyze the entire grid and see what elements of the world were missing, out of place, or too abundant. This system could have been used as Juno's sensors when picking out what goal she wants to accomplish. This would have brought us closer to our original goal of a Hybrid Agent when creating Juno.

I would also really enjoy being able to take a look at some source code for an existing simple autonomous agent in a game. A lot of the background research I did talked about the work floor of autonomous agents as opposed to specific existing architectures. I think being able to look at the source code of a

successfully implemented autonomous agent would have given me some really valuable examples to work off of and improve upon when coding Juno.