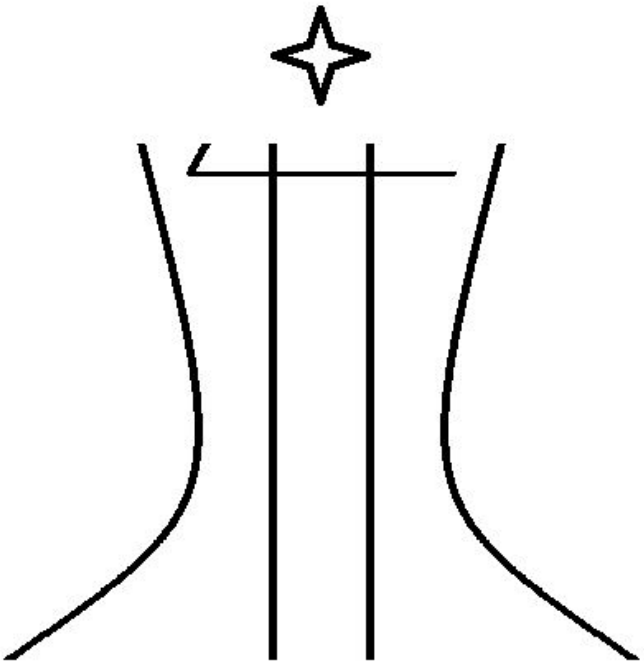


Design Document for:

RAM

"We tried our best!"™

Written by J4TC

Monday, December 11, 2017

Table of Contents

Game Overview

Philosophy

It's so lonely together

This game is trying to simultaneously simulate loneliness and companionship. Games typically leave you alone to make you feel lonely or give you a companion to or multiplayer options to create companionship. Our single player game gives you a companion but portrays the world as all a false fantasy. The companion character is so integral and connected to the character but was never anything real this whole time.

PC

Our game was designed for PC because the mouse and keyboard made the most sense for the controls. We also consider PC to be the easy platform to develop for due to its openness, availability and the vast amount of tutorials available to it.

Other platforms might be considered depending on the market performance of the current release, but remain out of scope of the current iteration of the project.

Bash can be a game by itself

During our initial brainstorming, we discussed many different topics. We did this with the hope of finding an interesting theme that everyone in the project would be happy to develop around. During this brainstorming, it was suggested by one of the team members to create a sci-fi game, with a magnet pushing and pulling different objects as the main mechanic.

During our discussion, we found that the magnet mechanic was not interesting enough to create a great game around it. From here one of the team members suggested the bash mechanic. The "Bash" mechanic from *Ori and the Blind Forest*. In his time playing *Ori* in the *Blind Forest*, our teammate found the bash mechanic to be really innovated, and it truly changed the way players can move within the game word. The bash was such an interesting mechanic that we felt you could create a whole game around it. So we did.

Narrative

The initial intention for the game was to make it tell a story that was backed up by the gameplay. There wasn't enough time to succeed at this philosophy, as the game relies heavily on dialogue and mandatory cutscenes, but the original idea wanted to bring the player into the experience. Narrative-driven games are often more like movies than games, and there are few that really convey themes and a story without sacrificing core gameplay or making the gameplay mostly irrelevant to the story. R.A.M. was an idea that stemmed from this fact, and we tried to make it work with the little time we had.

Player Agency

The game has a message or two that it wants to convey, but in the end, much of the game's ending culminates less in being about the characters and more about player choice. There are only two instances in the game that allow the player to pick anything, and the goal was to make at least one of those instances charged. That is, the player makes a big choice that has no right or wrong answer to it, despite there seeming to be a more "correct" answer.

Common Questions

What is the game?

RAM is a narrative-driven puzzle platformer about a man named Grant and his mysterious companion, Eden. They adventure through unknown places to find the keys needed to open a large door that might just be their way out.

Why create this game?

In the initial brainstorming stages, the idea of focusing on a central theme for the game was something all of our members agreed on. What this central theme would be was a mystery, but it was a good starting point for thinking, and we brought up the idea of conveying an idea or belief we strongly believed in. The core inspiration for the game’s story was in the genre we agreed on: puzzler platformers. Trying to remember something you’ve forgotten is kind of like piecing together a puzzle: you try to recall the bits and pieces until they finally come together to form a full picture. There was also the idea that memories come in good and bad flavors, and as terrible as it is to remember bad experiences, all of them can be taken as a person’s molds. We

are who we are because of both the good and bad things that happen, and that's an idea we're hoping to convey with the game.

On a more personal note, I (the game's writer) have always enjoyed narrative-driven games, but especially the characters they present. Writing certainly isn't my forte, but I wanted to create characters that could be relatable. The game's main focus is on remembering, but the characters talk about other ideas that may or may not be something the player is able to relate to and think about. There's also the element of narrative. There are many games that, when played a second time or given a second look after completion, give the player a sense of realization that some of the smaller details that seemed insignificant had relevance to the overall story and themes. This was another element of storytelling that I wanted to implement into the story's narrative.

Where does the game take place?

RAM takes place in a virtual reality world of the main character's creation that's been altered by the main character's companion. The various levels were designed to inspire recollections in the main character in an attempt to remind him of what he's locked away without overwhelming him too much.

What do I control?

You control a middle aged man named Grant. Grant is followed by his companion character, Eden.

How many characters do I control?

For the majority of the game, the player controls only the main character Grant. At some point in the game, the player will take control of his companion character, Eden, due to events in the story.

What is the main focus?

The goal of the game is to collect the various puzzle pieces littered around the virtual world in order to unlock a large, mysterious door. By the end of the game, the player is given a choice to make, and the ulterior goal of the game is to prime the player for that choice.

What's different?

Mechanically, we wanted to introduce a new gameplay experience. Games provide the player with the ability to become part of the world they're playing in. This game attempts to convey a message to the player instead of just telling a story that may have no relation to said player.

Feature Set

General Features

2D graphics
Original 2D pixel style sprites
Original Music compositions

Gameplay

Interesting new movement mechanic
2 player experience with just one player
Compelling introspective story

The Game World

Overview

The game world is a virtual world based on the main characters memories. The layouts of the levels let the player explore the intricacies of the movement mechanics while the visuals provide story context and pay homage to past games.

Doors

The doors in the game allows the player to roam freely throughout all the worlds and levels in the game.

Moving Platforms

These platforms emphasize vertical gameplay because it allows the player to travel to higher places in the world.

Key

This component is very essential to the game because it pushes the player to explore the map. Since we need a key to open a door, players will be more inclined to explore the ins and outs of the map to travel to the next destination.

Breakable Boxes

Breakable boxes are essential to have in this game because it allows the throwing mechanic to be used. Not only is this a bonus, but it provides a sense of enjoyment to the user that they are able to break an object.

Gaps

This component is vital to the game because it promotes using the jumping mechanic. There are various gaps through the game that tests the user's ability to jump in any direction.

Death Areas

This component was heavily introduced in the tutorial world because we wanted the player to get familiar with the jumping mechanic. If the player falls in one one of these zones, then they spawn in a designated respawn point.

The Physical World

Overview

The game world is a virtual world created by the main character. Each one of the levels is themed after an important memory for the character.

Key Locations

Door Room - this serves as a level select and hub world for the game. The room contains various doors that lead to all the different levels of the game.
Abandoned Lab - The final level where the main character restores all their memories. In this level you play as the companion character and at the end of the level you are given the final choice of keeping the world or not.

Travel

Between Worlds:
Movement between levels is done through the Door Room.

Scale

Each one of the game levels is design to represent some kind of structure. There is no set the scale to any of the levels, but all of them can be say to be divided between rooms. Each one of the rooms can be of different sizes, but they never become big enough as to take away the enclosure feeling from the player. This approach was taken in the hope of adding to the story and making the environment add to the idea of being "trap".

Objects

Key Memories - Fragments of lost memories scattered throughout the worlds. They are the keys to filling in the holes in your memory as well as the keys to open the final level.

Camera

Overview

The camera will generally follow the main character, Grant, to cover all the distance within the map. It follows the keyboard movement of "A", "D", "Space", "Left Mouse Click", and "Right Mouse Click". The camera will follow and catch every movement of the main character, focused directly at centered to capture the player movement.

Fixed Camera

Camera will put the main player as the center, and move accordingly to the movement of the player.

Other Camera Styles

The implementation of camera moving according to the developers was something our group strived for within the game; however, due to insufficient time and small amounts of developers, we weren't able to implement any other camera angles.

Game Engine

Overview

It is fairly simple to use even for beginners who had little to no experience in coding. Unity Game Engine provides users with resourceful assets as well as user sharing-based asset shops, resulting in easy development. It doesn't always require coding as it can also just be done by using the user interface Unity provides as a preview of how it will look overall.

Save Mechanics

The Game Engine provides user with a resource called PlayerPrefs, which is pretty much a saving mechanism, which allows developers and users to record and save either a boolean, integer, or a float even if the game or the Game Engine would close unexpectedly. It is essentially like a dictionary, where you are able to set values to a key and retrieve the value by using the key. The information is stored within the device in a form of data, and can be retrieved whenever the user calls the get method.

Easy Development Tool

When you first get into Unity Engine, almost everything is straightforward to use and requires little to no use of coding. Although it is better and more efficient to use coding to hasten the process of creating a game, it is user friendly to those who may have trouble in computer languages. It would almost be more beneficial to use at least a little bit of coding to form a hybrid development where one can use code at one point, then use Unity UI tools and assets. From coding in Unity for almost a year, I came to realization that the most effective way of creating game using Unity is to utilize the assets that are given or buyable, as well as using codings to supplement the assets.

Physics/Gravity

By using Unity, it cancels out the need for us to create our own gravity and physics, which saved us tremendous amounts of time. By the time we began to take on this game, Unity had recently came up with a mechanism that allowed users to pick and choose if they wanted to develop a 2D game or a 3D game. Before, the 2D games were developed in a 3D settings, but they are now providing its users with an ability to choose which development they wish to do.+}

It was our first time utilizing the physics and gravities within a game engine, where the developers are able to choose the gravity level, the drag level, mass level, modes of collision detection, and much more. We utilized the gravity scale as well as its body type. By choosing Dynamic body type, the object would act natural as it would in real life, but by choosing Kinematic, you are able to turn off all the physics and gravity in an instance, similar to ragdoll physics. This was especially useful for us since we used the Kinematic to stop both of our characters to move, then put some physics back to launch either of the characters into the air.

Animation

There were two objects within Animation that we used throughout our game: Animator and Animation Controller. The Animator made it easy for us to insert any Sprite animation to any objects. For example, if we wanted to implement an animation for walking, we would have to use several sprites to interconnect them to make it look fluent when walking. The only issue we faced is that each Animation Controller would have one Idle state, which would mean that we had to use a set of certain values to trigger certain animations. If the player were to jump, a ground state would equal false, which would trigger a jumping animation and so forth.

Collision Detection

For collision detection, we used BoxCollider2D and EdgeCollider2D. At first, in our earlier development, we decided to use the BoxCollider2D, which is essentially a collide detecting box that can change size and shape. We used it as a method of creating a space for a player to explore and walk around in. Not too long before the game was due, we found out that a collection of BoxCollider2D sometimes cause a character to

get stuck between the colliders, as if there is an invisible wall that's blocking the player's path. In order to fix this issue, we then decided to use the EdgeCollider2D, which is a one or more line of collision detection, where it prevented the character from getting stuck on an invisible wall. The only downside of the Edge Collider is that it is very tedious to arrange it in a way that it lines up perfectly with other colliders. Also, if there were to be a block on a 2D ceiling map, it would require 3 single Edge Collider lines to cover up, where Box Collider would only require one to cover up the block

The World Layout

Overview

The world of *R.A.M.* is structured like a hub world, in which there's a central location that leads into other locations. The player traverses through the contents of a door, but will ultimately always find him or herself returning to the hub map.

Tutorial

In the world of *R.A.M.*, the first world that the player experiences is a Forest, where forest objects are used to assist the player in learning the controls of the game. One of the main goals of the Tutorial is to create many different obstacles that can illustrate the mechanics of the game. Due to the character being in a virtual setting, we wanted the game to feel a bit more dangerous and adventurous, therefore one of the first obstacles the player encounters are the wooden spikes. The player continues to feel this sense of danger and adventure throughout the game. For example, the player is forced to go down through a tunnel because tall trees are blocking passage to the other side of the map. Once the player drops down the tunnel, it will venture through a big pond of water where it has to jump on and off platforms, in order to not drown. By having this pond, the map allows the player to get comfortable with our jumping mechanic that will be utilized throughout the whole game. In addition to this, the player will encounter various cliffs to help improve the player's abilities on using the jump mechanic to move vertically throughout the map. By choosing a layout designed based off of a Forest, it allowed us to easily find objects that we could incorporate on this level. Overall, the decision to choose a Forest proved to be an advantage for the player because

they can really get a first look at the look and feel of the entire game as well as understand how to play the game.

Hub World

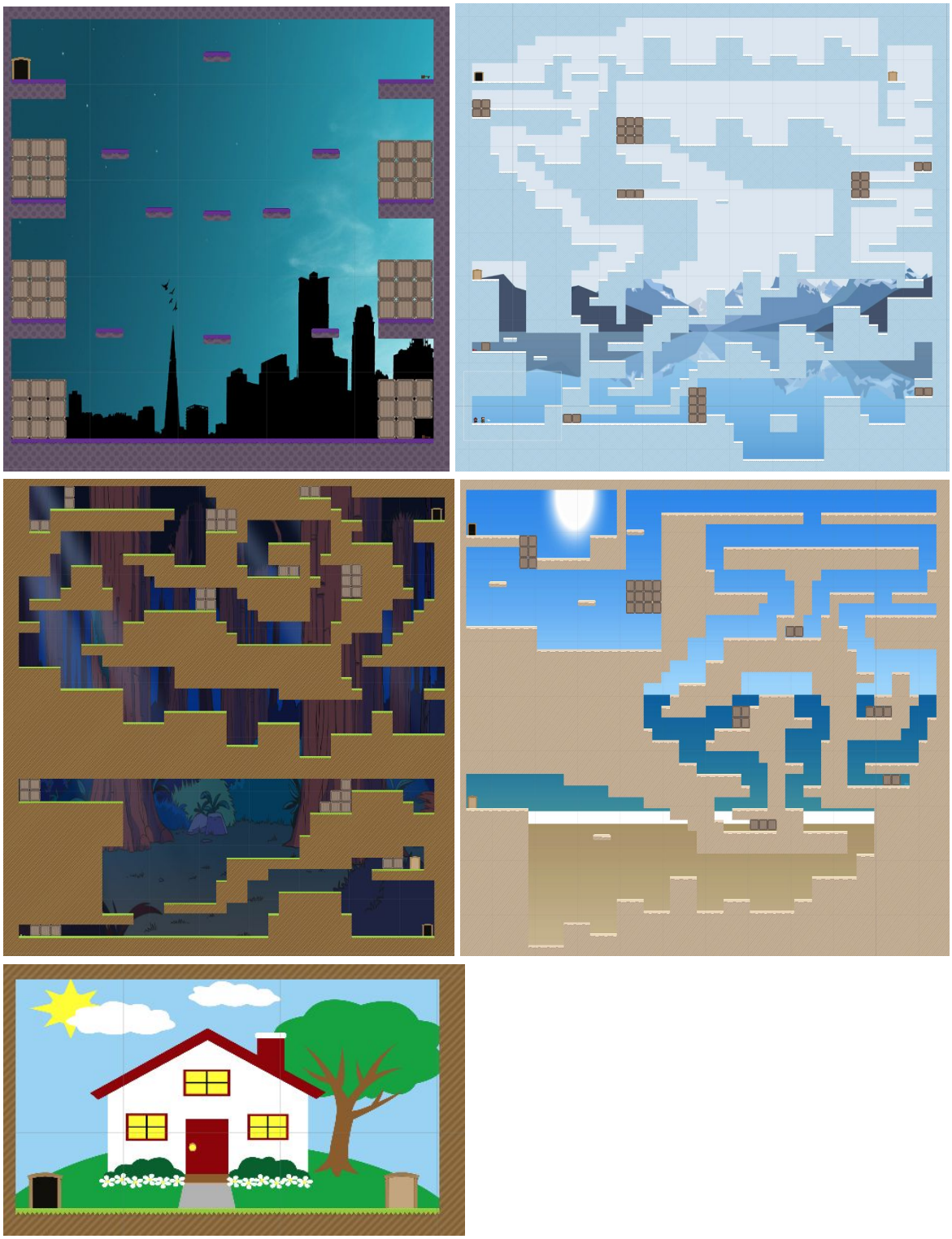
The main world is the Hub World. In this world, we created a very simple and straightforward design because it is primarily used as transportation to other worlds. This world contains four doors that allows the player to enter different worlds freely, if they have obtained the correct items. The first three doors have a normal sized door. However, in order to reach the final level, we have created a larger door to illustrate the significance of the final level. One of the design choices of the Hub World was to have platforms that were above all the doors. The reasoning behind this was to have allow the player to easily make a decision on the desired level they would want to travel to.

Level One

Level One was designed to showcase the game mechanics and allow the player to experiment with what can be done in the game. To do this the level presents the player with a series of different challenges, these challenges are meant to push the player outside the box and encourage them to experiment with the bash mechanic. To drive the exploration, the level is divided into different sections, each block by a door. The player is then tasked with finding the switches to open each one of the doors to reach the end of the level. The level presents the player with a series of narrow turns and passages that force them to think creatively, so they can successfully place their character to clear the level. These passages are also meant to test the accuracy of the player's movements and to allow for mastering of the movements if the player wishes to do so. In the opposite side of the spectrum, level one presents the player with the biggest open space in the game. This open space is designed for the player to put everything they have learned through the level to the test. In this section of the level, the player is able to explore the big open spaces in their search of the last few door switches. The level adds to the search theme by adding a series of mock objects, which are simple placeholders to diverge the player from finding the switch too fast. Overall, these design decisions encourage the player to get to know the level, to backtrack and to search in every passage.

The theme for the level is still a little unclear. The level was originally designed as a laboratory, to accommodate the idea of the main character being a scientist trap in his on VR dream. During, later meeting it was agreed to change this theme to a cemetery theme, so it could be used as the first level thanks to its exploratory design, and to add to the backstory of the characters and the death of the main character's wife. During the final production, the level was finalized as a cemetery but due to time constraints, it had to be used a science laboratory since this was a more crucial part of the story.

Level Two



One of the more creative levels on *R.A.M.* is Level Two because of all the various worlds the player can visit. In this level, the player encounters a snowy terrain, beach terrain, dark forest terrain, and an evening city terrain. In addition to this, each level has some form of vertical gameplay. Because of this, we needed to incorporate moving platforms to offer a new dynamic to the game and increase the difficulty. Throughout the map, we designed the level to force players to jump in various ways. Another item that can be found in the map are breakable boxes. Throughout each terrain, these boxes can slow down the player's

progression or hide key items that are needed to open the next door. by forcing the players to go through doors and enter different worlds, the layout of the 5 terrains was created to disrupt the sense of direction in the players. Lastly in the "Night World," we created three accessible areas that are blocked by a box to have the player search for new doors and keys. In addition to this, there are many moving platforms that are difficult to navigate through to reach elevated areas. With many different dynamic features, Level Two can be seen as one of the most creative levels in the game that heightens the difficulty of the game, essentially making it more engaging and entertaining.

Game Characters

Overview

All of the characters present in the game are humans or the remains of humans (in a non-morbid way). The character that the player spends the most time playing as is a middle-aged man named Grant. The other character that the player will play as is a teenage boy named Eden. There are no other playable characters in the game, though other characters that are related to the main characters will be mentioned.

The Main Characters

Grant

Age: 30

Grant is the main character of the story. He's an oddball that loves to read and tell stories. Optimistic and fairly carefree, he likes to go on adventures and doesn't ask too many questions when he can find the answers to those questions for himself through experience. He works from home as an inventor of sorts, though it's never quite clear to anyone what his occupation is.

Eden

Age: 15 (in the virtual reality)

Eden is Grant's assist character in the story. He's an orphaned child that Grant adopts. Eden has a bit of a dry personality, but exhibits a fair amount of intelligence. He's familiar with the general layout of the virtual reality due to being the person who changed it in the first place, and he spends the majority of the game trying to lead Grant to the end without jeopardizing the final objective. Due to being ill, he was unable to make the virtual reality to his specifications before dying, so he implemented backup abilities that his virtual character could use to help Grant reach the end. Eden has an interest in art, and he occasionally sketches in his free time. Ever since Grant froze himself and entered the virtual reality, he's since abandoned that hobby in favor of studying to try and convince Grant to come back to the real world.

The Side Characters

Rosanne

Age: ???

Rosanne, dubbed "Rosie" by Grant, is Grant's wife. She doesn't appear in the game, but Grant often mentions her when telling Eden stories. Based off of what Grant says about her, we can conclude that she enjoys playing video games and is a relatively good artist. She also has a fear of children due to their fragility.

Christie

Age: ???

Christie is Grant's daughter, who hasn't been born yet in Virtual Reality Grant's memory. Grant doesn't say much about her because he doesn't remember her, but she died at a young age. Other sources say that in her childhood, she enjoyed watching Eden draw.

Enemies and Monsters

The game does not have any enemies or monsters. Being a puzzler platformer, the only enemies are the obstacles in the level that the player must overcome in order to reach the end and acquire the necessary item to continue.

User Interface

Overview

The interface is design to be a simple as possible while still being function. Inspiration is been taken from the 16-bit era. This is being done to maintain the same retro style through both the game and the interface.

Main Menu

The main menu can be accessed by pressing the Esc key. From this you can adjust audio settings, restart the level, or quit the game.

Home Screen

The home screen is where you can start a new game, load a saved game, or close the client. The home screen is present when you start the game and you can return to the home screen by hitting quit in the main menu. Within the home screen, the title changes according to how many puzzles the user has collected.

Key Prompts

During the game prompts will appear to guide the player in the controls of the game. These key prompts were each individually spirited and animated for this game.

Dialogue Boxes

Dialogue is a main part of the storytelling in the game. Dialogue boxes will show the conversations the characters have with each and can be interacted with any key. Occasionally dialogue boxes will include choices that the player can select to guide the relationship of the 2 characters.

Attacks

Overview

Due to the story driven gameplay of the game the only attack available will not be performed by the main player. The attack will be perform by his companion character. This attack will only take place during the final level in which the companion character becomes the final boss.

Attack Details

During the final level the companion character will take the form of the wrecking ball from the bash mechanic secondary move. While in this form the companion character will follow the main character throughout the level, in the hope of making contact with him. If contact is reach …

Musical Scores and Sound Effects

Overview

The goal of the overall sound design is to bring together three major elements and themes in the game. Reminiscence, regret, and adventure. This is explored in the variance of modes in the keys in each piece of music to have moments of hope that are directly contrasted with momentary despair seconds later, with much of the sounds being light and airy in order to convey a nostalgic feeling.

Soundtrack

Virtual Forest

The track is designed to be conducive of a puzzle game, being complex enough to mirror the complexity of a potential solution to what challenge the player is currently platforming through. It allows for a nostalgia factor in utilizing light and airy synth that is in the happy key of A major whilst using the mode switch to its relative minor to foreshadow the events truly at work underneath the lighthearted events transpiring on the player's screen. The heavy usage of synth refers to the fact that these events are not naturally transpiring, that is to say, that it is a replication of the former world the player character used to be a part of just as the synths are replications of sound that don't naturally occur in the world.

Level Selection

The most obviously sad and foreboding of all the songs on the soundtrack. This song starts and ends with the emotive undertones that connect the stories which the player has to traverse. In a way it is very obviously foreshadowing the true underbelly of all the events that are transpiring in front of the character's face, but hidden behind the screen just as the level selection system is showing momentary truths hidden behind the facade of memories the player must wade through.

Given Up

Rooted in both the act of giving up and being given up, representing to duality represented by both characters in this game, this track relies on distorted bass synth and light, airy treble tones in order to convey a message. The "verse" part of the track paints a happier picture which is starkly contrasted by the "chorus" part where everything seems slower and have an almost haunting usage of the same synth that defines such a happy melody in the "verse". This track also draws heavily from trap music as well to keep the energy high.

Sound Design

The goal in mind whilst developing sound design was to use household items and settings in order to convey the importance of reminiscing to the main character and thus injecting small imagery of the nostalgia implied by items in one's household through each of the sounds the player can make.

Single-Player Game

Overview

In the single player of RAM you explore the world and story and overcome the various levels and puzzles. The game launches you straight into the single player campaign where you learn to play the game, learn about the characters you control and learn about the nature of the world itself.

Movement

Movement through the levels includes basic movements mechanics such as:

Walking right:

Is done by pressing on the "D" key or on the right arrow key. The character speed is always kept constant for the main character. For the companion character the speed is kept constant until the character passes a certain distant threshold (the specific distance change between levels due to scaling difference during level design) ones this threshold is passed the character companion character's speed is doubled until the character goes back into the set distance threshold.

Walking left:

Walking left is done by pressing the "A" key or the left arrow key. The movement works the same way as it does while walking right.

Jumping:

Jumping with the main character is done by pressing the spacebar. The distance and the height of the jump depend on the character speed and on the jump force of the character. The characters jump force changes between levels due to scale differences between the levels. Jumping with the secondary character only happens when the secondary reaches the spot on which the main character jumped, after doing this the secondary character proceeds to jump.

Special Movement Mechanic

The unique movement mechanic in the game involves a jumping, throwing, and bouncing your characters. We have named this mechanic the Tech Toss. While in the air or on the ground the player can press on of the 2 mouse buttons. Doing so will freeze the player in place and gives a directional arrow that can be rotated with the mouse button. Depending on

what mouse button you clicked determines the following action. The left mouse turns the second character into a spring and bounces the main character in the direction of the arrow. This bounce is higher than the jump making it your primary movement option. The right changes the character into a ball which the primary character throws. In this iteration of the game the primary function for this is to break boxes in the stage to search for keys.

Story

The game is a story about the main character, Grant, as he remembers events in his life that he's forgotten. Years prior to the beginning of the game, Grant loses his wife in an accident and is stricken with grief. During this period, he adopts another child so that his only daughter can have a companion for the times he's unable to leave his workstation. One day, in a fit of grief-stricken insanity, Grant accidentally knocks his daughter down the stairs, killing her. The event causes him to lock himself into his workstation so that he can complete the virtual reality world he'd set up for himself to cope with the loss.

After locking away his memories behind a closed off area of his virtual world, Grant sends himself to live out his life in a world from before everything in his life went wrong. The adopted child, Eden, doesn't agree with what Grant's done to himself and spends the remainder of his life working on the virtual reality using the information that Grant left behind. Before his death, Eden alters the virtual reality Grant is in and inserts a younger version of himself into the world to assist Grant in reaching the locked off area.

Hours of Gameplay

The game is short but relatively hard. With complete mastery of the movement and knowledge of the locations of hidden key items the game can be completed within a few minutes. We expect the game to take much longer as the player will be tasked to traverse labyrinth like levels, search for items, read the story. After the implementation of the game, the average player will be able to finish the game in about 2 hours. This can vary depending on the player's skill level with our jumping mechanic. In addition to this, the game duration can increased based off of the player's reading speed.

We hope that the story of the game will engross the player enough to draw them in for longer and invest more time in the game.

Victory Conditions

The game is completed when the player makes it through the final door and confronts Grant with a choice: will the player choose to have Grant reset the entire virtual reality, deleting Eden in the process permanently but sending Grant back to his blissful fake reality, or will the player choose to have Grant shutdown the virtual reality entirely and return to the real world to deal with his circumstances?

Multiplayer Game

Overview

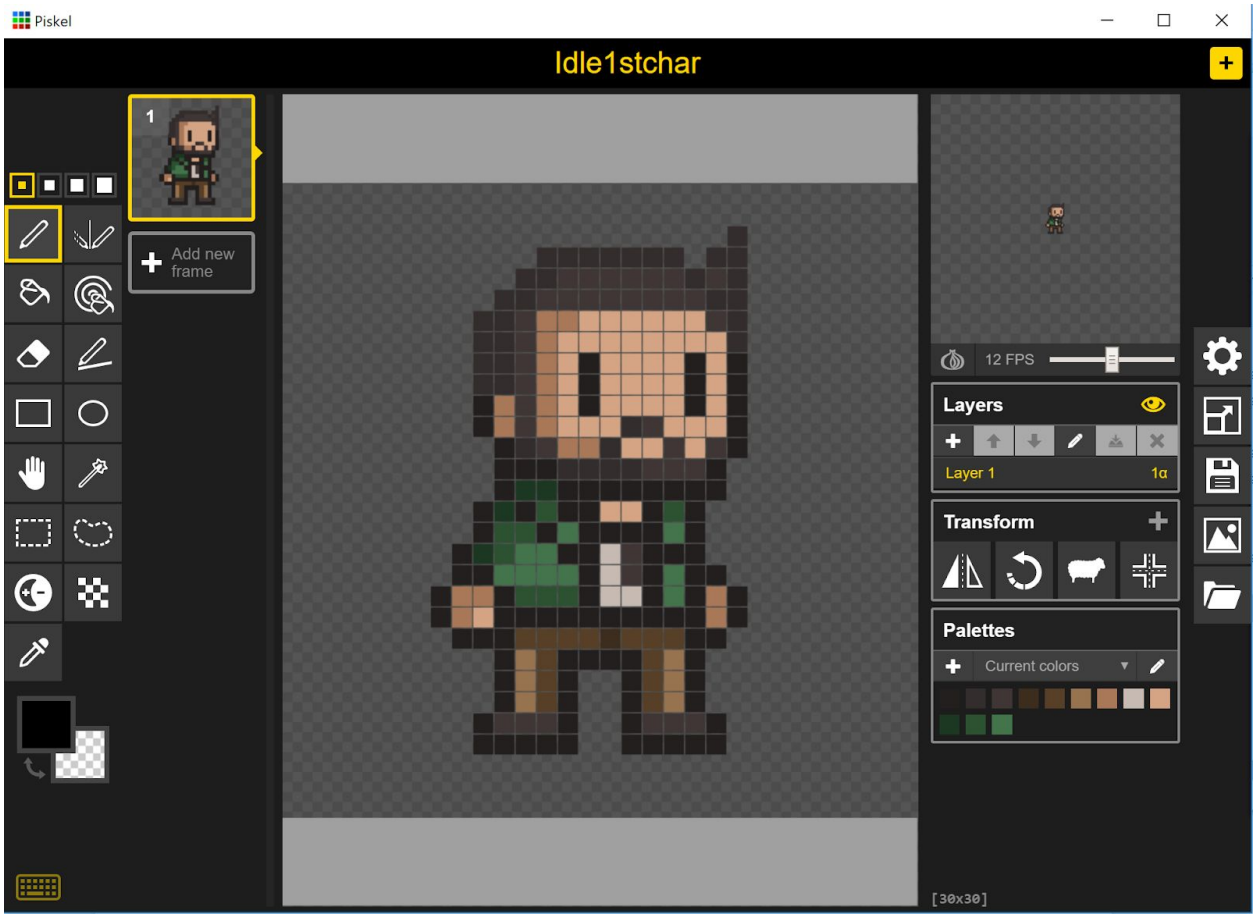
Currently the game is only playable in single player mode. Adding a multiplayer option was promptly eliminated as an option during brainstorming. The reasons behind this decision were to:

- Allow more development time to the current version of the game and to polish the gameplay as much as possible.
- Offer more movement options to the main player, adding to the possible different gameplay styles.
- Separating the characters movements between two players would break the flow of the game.
- Adding more characters would affect the believability of the story.

Character Rendering

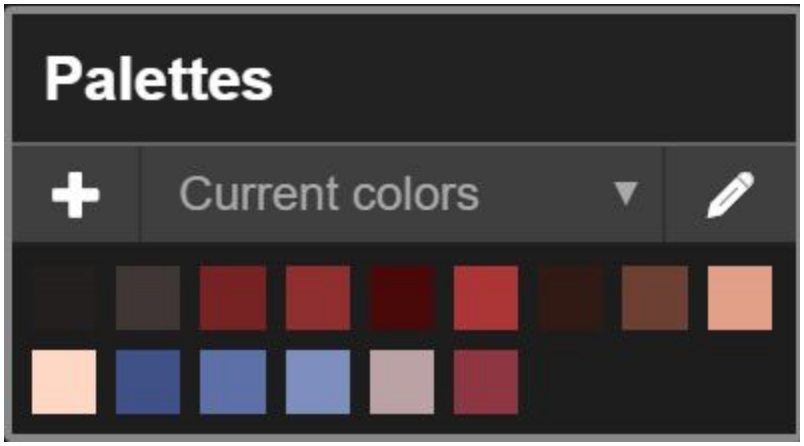
Overview

The characters' sprites in the game were rendered using Piskel, a free browser software for making and animating pixel art. The design was meant to be simple while making the two characters distinct.



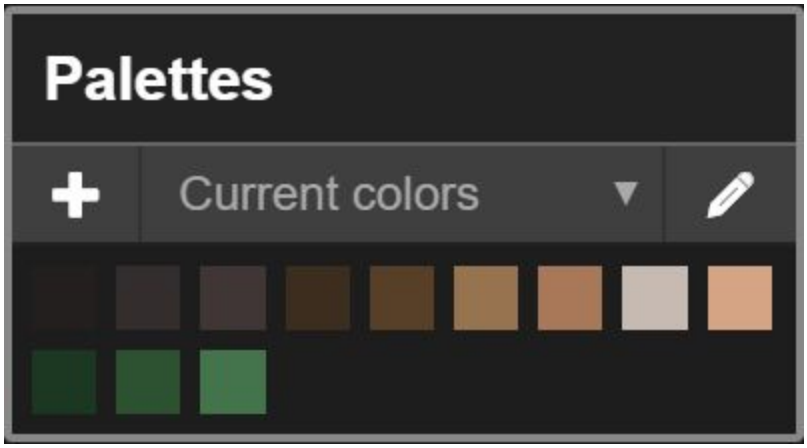
Color Scheme

Between the two characters, there are only a few colors shared between them. It was imperative that they could be told apart, but it was also important that the colors held some significance on their own, as well as in relation to one another. The primary color on Eden is red. Contrary to the implications of the color, Eden isn't hot-headed or over-the-top. It isn't until the end of the game that the significance of his color shines through: he's a passionate person at his core, and the actions he undertook to end up where he is are indications of his dedication.



On the other hand, Grant's primary colors are more subdued. Green and brown are the colors that pop on him, but he's a much more excitable person and less down-to-Earth than

Eden is. By the end of the game, he becomes grounded to the point of negativity. However, he and Eden aren't opposites. Their colors (green and red) are complementary, and likewise, their personalities are meant to complement each other. They're also Christmas colors.



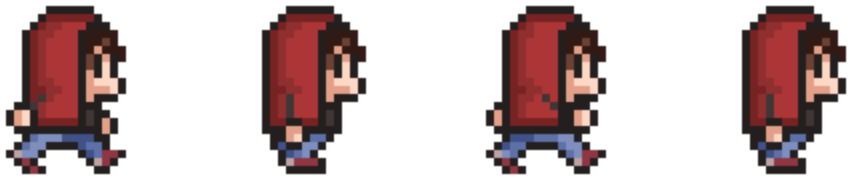
Height

It's hard to tell the player outright that Eden is younger than Grant without inserting their ages and making it seem out of place. Their height difference is meant to emphasize the difference in their ages, despite their personalities sometimes saying otherwise. It was also there to emphasize their relationship. Making Eden as tall as Grant would make them seem more like friends (and they are), a key component in their relationship is family ties. Eden is shorter to make him seem more like a son and less like a brother or just a friend.

The Process

The character sprites in the game are all handmade, and so the goal was to be reminiscent of older games without being too similar to them. There were no reference models used when making them, so there were some difficulties getting some of the actions to look just right. Piskel has the issue with being unable to rotate an image any less than 90 degrees, so many sprite manipulations were done by the game coders in Unity. Both character sprites went through multiple iterations in trying to find just the right look, and many of them could still use more work.

First iteration of Eden's walking sequence



Second iteration of Eden's walking sequence

World Editing

Overview

The World Editor in Unity contains multiple tools that allow us to create a new world. For all of our designs, we used Unity 5, specifically it's 2D developing environment. We found Unity to be a great tool, in part for its simplicity, but more importantly, thanks to the thousands of tutorials available online for the game engine.

Ground Objects

As a team, we found that creating the ground objects was the most difficult part of creating the levels. During our first interactions, we used box colliders 2D for all the ground sprites. This technique was successful at the beginning, as we learned to use the engine and its tools. Later in the developing process, we found that our character kept on getting stuck on the divisions between sprites. After some searching, we found that this is a common problem, with box colliders 2D, and the using edge colliders was the best option. After discovering this, we decided to change all of our levels to edge colliders, but due to the late-developing phase that we were on, we were not able to truly polish the ground objects as we could have if more time were allowed.

Scale

During the last phase of our development, at the moment of merging all the different levels together, we found that the scale of each level relative to each other was completely different. At this point of the development we were unable to make any changes to the level designs and we did our best to accommodate each level with a universal character sizes as a temporary solution to the problem. Adding some kind of universal unit to the editor would greatly improve the usability of the editor, since it would allow for more consistency between scenes.

Missing Goals

Overview

During the development of the game the team had a lot more ideas that were not able to be implemented because of time and technical limitations. With more time we could possible

implement more features that would improve the feel of the game and better support its main mechanics and tell a better story.

Door Puzzles

Currently our game is mostly just a platformer with maze like level designs that test the player's ability to navigate the world. We wanted to add more traditional puzzles in between each level. These puzzles would serve to richen the story and diversify the gameplay. The collected puzzle pieces that are scattered through the level only serve as an arbitrary key to open the door to the next level. Giving those pieces a gameplay mechanical element would make them much more interesting.

Saving

We currently use an auto save method for how we handle death and respawns . A more interesting approach to this would be allowing save spots to be manually assigned by the player. This would be implemented as the protagonist texting their SO (but not really because the SO texting the protagonist is actually the assistant character pretending to be the SO, who's dead) and leaving a sort of memory trace where they come back. This would use a consumable resource, cell phone battery which makes it so that the saving becomes an interesting game in itself and give the player more control. The cellphone and texting could also be used as a story plot point further fleshing out the world and characters. A mechanic that would be paired with this is collectables: Cellphone Batteries that would recharge your ability to save.

SFX

The implementation of the sound effect would have improved the user experience as they would hear the jumping sound, Player1 throw, Player2 throw, etc. We wanted to include sound effects for each and every mechanic Player1 and Player2 would have to make it more entertaining and immersive. Problem was that while we began developing, we were more focused on the actual mechanics and the maps that we did not find enough time to implement the sound effect to the mechanics. We tried to sort out our priorities by focusing on what was more important since the sound effect was merely an addons for us. We wanted to produce everything to its performance stage, then set the sound effects.

Character Name Reveal

Within the game, the player has the freedom to choose if they want to see the optional dialogues or just choose to ignore and move on. One of the implementation we wanted to incorporate was to give merit to the players if they were to read those optional dialogues. For example, in the beginning of the game, Player1 and Player2 begin the dialogue with Player 1 name being revealed as Grant and Player2 name being the question mark ("???"). If the user would read this one specific dialogue, the Player 2 would reveal his name as "Eden" and throughout the rest of the story, his name would not come up as "???" but rather as "Eden". This implementation would mean that I would have to implement a code that would replace all the

words that equals “Eden” and replace it as “???” and vice versa if the user didn’t go through that optional dialogue.

Dialogue Details

During the optional dialogue and cutscenes, the Player1 and Player2 would sometimes require for them to face each other so that it looks as if they’re talking. Not only that, but we also wanted to try to make the characters “shake” to show that the characters are actively talking. This would mean that we needed to create another system that will check where the players are positioned, which way do they have to change their direction, and create a “shake” animation. Sometimes the positioning of the players needed to be changed because either they would be overlapping each other, other times the dialogue would start and the Player 2 would not be seen within the dialogue since we implemented the dialogue scene to have both Player 1 and 2 to be not moving.

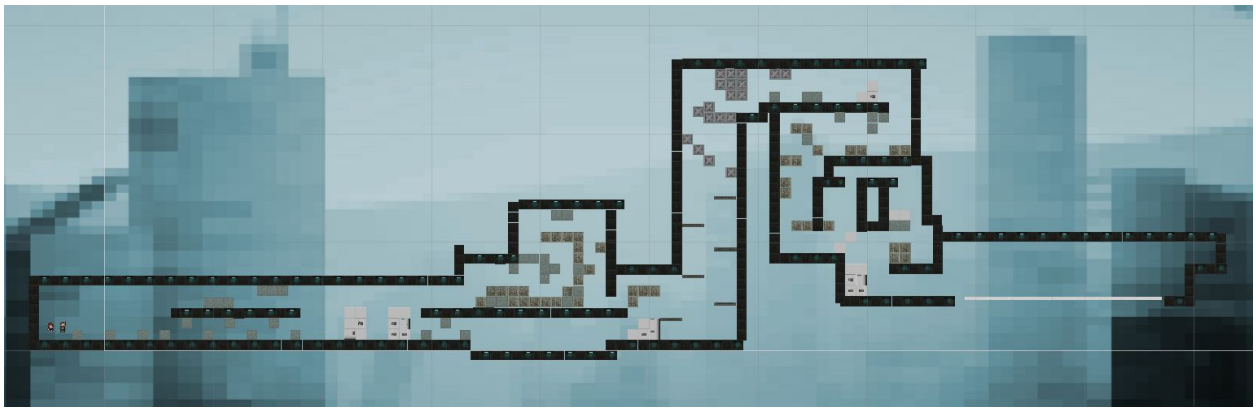
Music For Every Level

We wanted to create individual music for each levels the user had to go through to give the feeling of immersiveness. The only problem we encountered is that we struggled to find sprites to use for each of the levels, which delayed the presentation of how it would have looked. For us to create the music, we needed to know how the level looked like in order for us to recreate the aura that the level has. Fortunately, music wasn’t a big part of our ‘Goals’, but it was something we were looking forward to implementing.

Credits

The credit would not only include our names, but also some honorable mentions for the sprites and assets that were given out freely on the internet world. Although the credit would mostly include our groups, it would have been nice to put it out there as a way of completing the game. If the credit was to be implemented, we wanted to include it either at the end of the game, or within the Menu_Scene where users can choose Credits to watch it as an optional video.

Final level



The “boss” level. Lab-themed, to match with the main character’s lab in real life. It’s drab and rundown, with electrical wiring ripped out and broken machines/chemical vials.
Locked door puzzle: None

Background concept: By unlocking the final door and entering this level using pieces acquired from previous levels, the main character will remember all of his memories and go temporarily insane from grief. The two main characters are separated from each other, and using the second main character, the player will have to chase down the “boss” to be met with the game’s final decision of staying or leaving. The backstory of using the lab for this level is to represent the final moments before entering virtual reality, as well as letting the player have the revelation along the way that the main character inadvertently killed his own daughter in an experimental run of the virtual reality during the early stages of development.

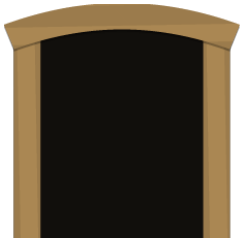
We decided to make it a sort of Super Monkey Ball type of level where the second protagonist is stuck in his “wrecking ball” form (due to being too far away from the main protagonist), and can only roll around the stage rather than walk. Unfortunately this level was unable to be implemented.

Appendices

Objects Appendix

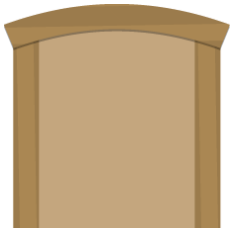
In this section, we will go over all the objects that were used throughout the game and a brief description of the object’s functionality.

Door_Open_Light_Top_Round



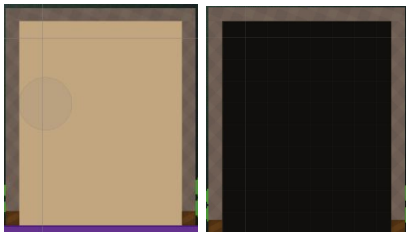
This object is the open door that allows the player to enter a different scene or area in the game. We can enter the object by pressing the “F” key.

Door_Open_Light_Top_Round



This is the closed object door that was implemented throughout the game. It is considered a locked door and can only be opened by a puzzle piece or a key.

Open/Closed_Final_Door



This is the Final locked door that it is noticeable larger inside the level. The size of the door is illustrate the significance of the final level, the boss battle.

Final_Key

The key that opens the Final door where the player can enter the Final Terrain from the Snow World.

Grass_Bottom_Key

The key that opens up the door to the bottom portion of the grass terrain from the sand world area.

Night_Key

This key can open up the door from the bottom portion of the grass world to the Night world.

Sand_Key

This key opens the door to the Sand world from the beginning area of the snow world.

Grass_Top_Key & Snow_Key

Both of these keys open up the doors to the top portion of the Grass World and the last portion of the snow world. They are both the same keys to in the same world to force the player to explore the worlds more.

Goop/Grass/Sand/Snow_Tile_Flat

These objects are primarily used as “Ground” objects throughout the game. These tiles act as the ground that the player can walk on throughout the map.

Goop/Grass/Sand/Snow_Tile_Half_Round-01

These objects are primarily used as platforms that move independently throughout the game. A player can jump on these platforms and access higher areas of the map.

Puzzle_Pieces

These puzzle pieces can be found throughout the game. They are essential to the gameplay because it allows the player to enter/exit the hub world and leave the end of each level.

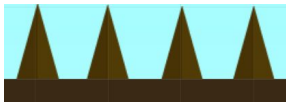
Tutorial_Platform

In the Tutorial level, these platforms are used to represent the ground objects that a player can walk on.

Trees

These objects are trees to illustrate that this level is based in a forest. In addition to this, these objects act as boundaries that the player cannot walk through.

Spikes



These spikes were used to create an obstacle for the player to jump over in the tutorial level. If the player falls into these objects, then they will respawn and pre-designated spawn point.