Platform64

Now with 10% more Word Art!

Design Document

By Buu342

Index

Introduction

This document will serve as an introduction to my game project, currently under the working title of Platform64 (I know, very creative). This will contain a list of goals, ideas, and explanations over specific mechanics of the game, both so that the project's (potential future) contributors can get an overall feel for the game, but also for me to remember what I thought about in the shower. Having a fully written design document before plunging into the project will help with having clear goals, and allow for me to follow along with how much has been developed. While some aspects of the project are still being thought about, the initial engine write up will give some extra time to clear these out.

Essentially, this is a big set of sheets explaining what I plan on having in the game, so that the full scope of the project can be "visualized" beforehand. That doesn't mean that what is written in this document will be in the final product! I'm very certain some of my plans will not follow through in their entirety.

Much like what I did with my first released homebrew, Pyoro64, I'm planning on keeping documentation of the entire project. The documents about Pyoro64 were mostly written after the ROM had been finished, and they sure as hell took a lot of time to write. As a result, I have opted to instead make some bi-monthly-ish videos that outline where the project is currently, challenges that were faced, and more. I will still attempt to be educational and explain the techniques or nuances of the N64 or of this game's engine as much as possible, which will hopefully be aided by the video format as I can make the explanations a lot more visual and dynamic. This will be covered more in the **Documentation** section.

Tl;dr for the entire document, I want to make a sidescrolling 2.5D platformer akin to Mystical Ninja 2 Starring Goemon (Goemon's Great Adventure) but with a bit more focus on movement speed and flow.

Story

Currently, story is what I have worried least about, as the engine has to be taken care of first. Ideally, I’d like to tell a story about world cultures. Much like the Goemon games have always been about Japanese culture, I’d like this game to showcase world culture, providing a spotlight for more obscure folklore creatures and traditions.

In this world, the monsters and creatures which were used to represent the fear of the unknown, or to tell tales of heroic doings, are real. At odds against overwhelmingly strong and dangerous beasts, humanity was pushed back into heavily fortified towns/cities. Small villages and settlements simply aren’t feasible, so everyone took refuge to the closest heavily fortified community. As a result of all the asylum seekers being from all over the continent, these burgs became incredibly rich in various different cultures. However, as a consequence of trading between communities being highly dangerous, each settlement is essentially “stuck in the medieval ages”.

Our two main characters are mercenary partners, some of many who risk leaving the fortified walls to gather resources, deliver messages, or perform any sort of odd-job. Leo is a full time merc, having a lot of survival and combat experience as a result. Catherine, on the other hand, while very combat apt and adventurous, only performs mercenary work once in a while. This is because she also spends a lot of time in the burg, helping out the city’s wizard (who, in turn, gives her access to his library and knowledge).

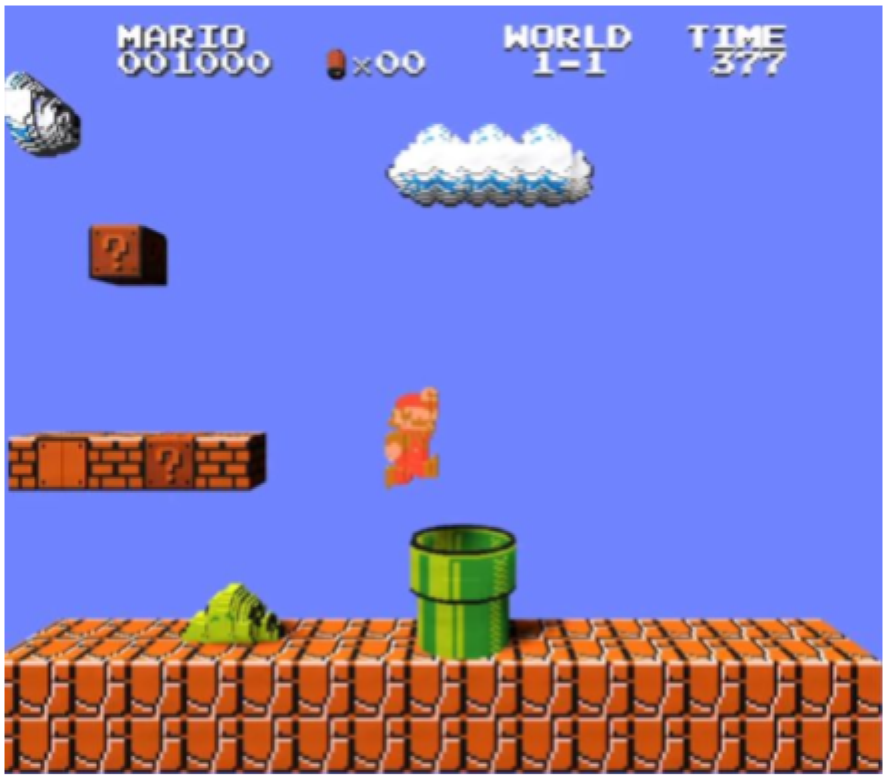
Due to his more rugged lifestyle, Leo is grounded, practical, and generally more composed. This contrasts with Catherine, who is typically more upbeat, adventurous, and genial. Despite her more gung-ho behavior, she is still tough and vastly knowledgeable about cryptozoology (I guess in this case it’s just zoology) thanks to her time with the wizard.

The story originally starts as a small quest: one day, the wizard requests the pair investigate a monster hotspot for clues on where they came from and what can be done to eradicate them. Upon completion of this small quest, however, the character’s journey balloons in size as they eventually come to the conclusion that eradicating all the monsters is entirely feasible… So long as the wizard is able to keep funding their adventure.

Gameplay

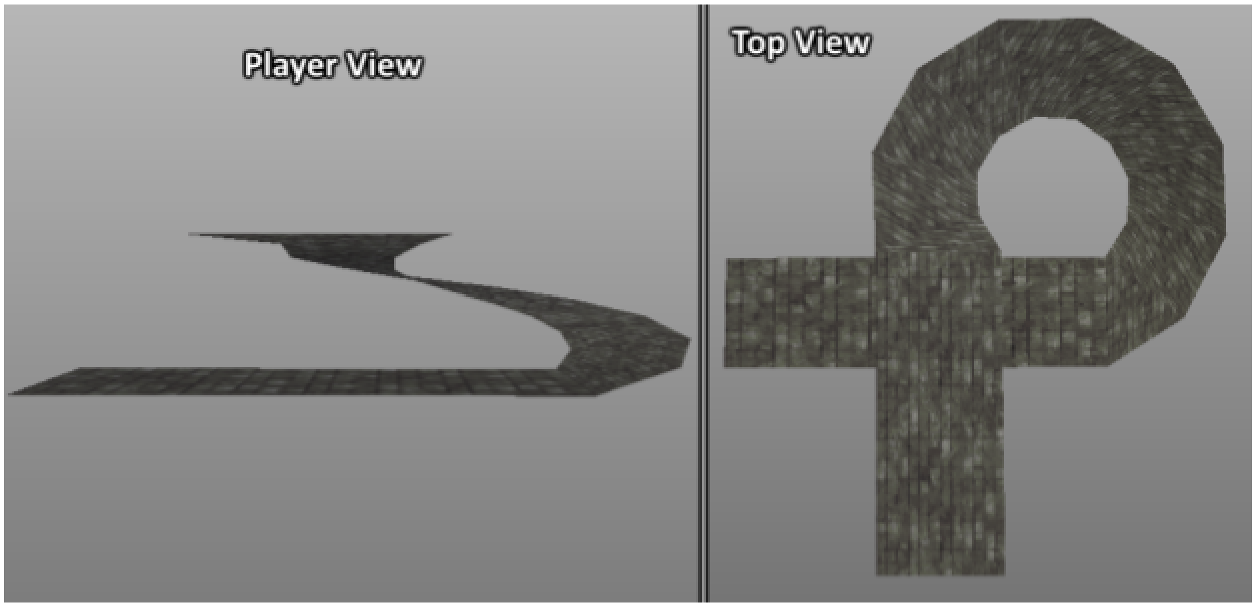
Platform64 is a 2.5D side-scrolling platformer, similar to Mystical Ninja 2: Starring Goemon (also known as Goemon's Great Adventure). In order to differentiate this project from MN2, however, this game will emphasize player movement. One of my favorite aspects of MN2 is how most of the gameplay consists of (mostly) holding the stick in one direction, and then just timing your jump and attack buttons right. While this sounds boring it is surprisingly fun, and observing speedruns of the game really goes to show how fluid the gameplay is. As a result, the idea behind this is to make the player feel like they can cut through the level like butter, which will be aided by a variety of different abilities that can be chained together. This doesn't mean all the levels will play out like "run and gun" (er, run and slash?), as the engine should hopefully allow for some interesting platforming and level design that will provide different "speeds" to each level.

Given that the Nintendo64 is a 3D console, it is key that the project utilizes the 3D capabilities to its "full potential": instead of just having a 2D platformer that has had the characters extruded out towards the screen.



While the 3D looks cool, it's not being used for anything gameplay wise, meaning the action is still only happening in 2D.

As a result, the plan is to take advantage of 3D to create more interesting level design, allowing for the level to rotate and provide the player with different paths. This could also allow for speedrunning tricks by creating intentional shortcuts.



Illustrated example. The player can choose to proceed to the right, which the game will automatically loop the player around. However the player can choose to jump to the top from the bottom directly, skipping the loop entirely.

At the start of the game, not including movement, the player is given the ability to

- Jump
- Attack
- Shoot projectiles
- Roll

The player can also use some other, more complex abilities that should allow for more interesting movement options, and can be chained together to create interesting ways of either traversing the level, or exploiting them for speedrunning purposes:

- A charged attack that unleashes a quick dash slash

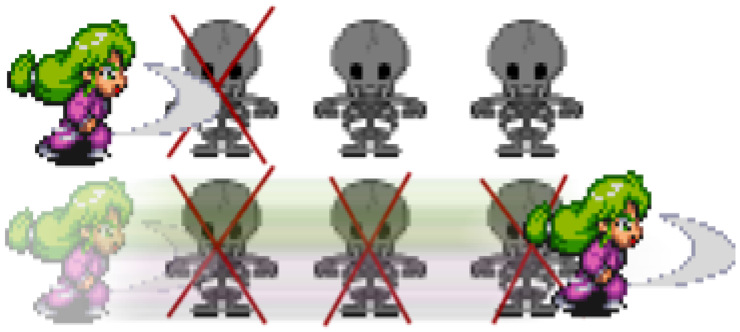


Illustration showcasing normal attack (top) verus charged attack (bottom)

- Charged arrows that shoot a more powerful projectile which pushes the player in the opposite direction it is fired
- Swimming underwater
- Double Jump

These should provide really interesting movement techniques that can be mastered with practice. For instance, to cross a really large gap, the player can double jump, charge slash, and shoot a charged arrow downwards to give one last jump boost.

The game should be both singleplayer and cooperative using a drop in/drop out system. Cooperative should be more than just adding in another player, as it should provide something interesting for the gameplay, for instance new movement options or the ability to combine both character's attacks to break open new paths.

Since this is a globe-trotting adventure, ideally the story campaign is split into 5 different "continents", one that represents Europe, Africa, Americas, West Asia + Northern Europe, and East Asia (plus Australasia). In each continent, the player starts in the town, which they can use to purchase items (consumables, clothing, upgrades), heal or sleep to pass time. To transition between continents, that must be done at the towns. The towns can also provide more fun side activities, such as a colosseum which can be used to allow competitive multiplayer instead.

Ideally, there should be no overworld in each continent, instead each area is connected to each other with some fast travel spots in between (to reduce monotony of having to traverse the same levels over and over).

Goals of the Project

High Priority Goals (IE Stuff I won't budge on)

- 100% console compatibility. The project is developed for the N64, so it makes sense to be N64 compatible. If the plan was to only focus on getting the game to work on an emulator, then I'd be making a PC game and not an N64 title.
- Because culture plays a huge part in the story, the game needs to provide an encyclopedia that contains information about the real mythological beings which the monsters are based upon.
- 2 player support with drop in/drop out co-op.
- Free and open source, licensed as permissively as possible.
- Retail game length.

Medium Priority Goals

- Day and Night Cycle. This can be used to change up the gameplay, such as enhancing certain enemies' abilities or opening new paths. The time of day can be changed by either waiting it out, or sleeping in a town.
- Extra side quests available in each town. Ideally these could have an actual plot, rather than simply being a fetch quest. The side quests would award money, which can be used to purchase consumable items/food.
- Dynamic split screen (Doesn't need to be as fancy as a voronoi split screen). I'd prefer this than forcing the players to be very close to each other (like in Mystical Ninja 2).
- Differently themed colosseum minigames for each town.
- Unlockable abilities as the story progresses.
- Having all the levels be seamlessly interconnected. By this I mean, no overworld: start at the town and just keep moving to the right until you reach the boss.

Low Priority Goals

- Multiple character outfits which reflect the armor typically worn in the respective region, purchasable in each town.
- Ability to calibrate joystick sensitivity from the options. I can't think of any N64 game that provides this, which I feel is somewhat important nowadays with all the different available controllers/sticks.
- Tweakable graphic options, such as the ability to disable anti-aliasing, texture filtering, etc…
- Voice acting would be nice, at least for the main plot cutscenes or, if not enough cartridge space, just the intro and ending cutscenes.
- Support for multiple languages. Ideally the game's strings can be written into a lookup table so that translation efforts can be done without having to dig through the entire source code, the translations themselves are what is of low priority to me.
- Rumble and Expansion Pak support.
- Built in speedrunner timer.

- Multiple difficulty levels… At least just a harder mode.
- A PC port would be dope, but this <u>will not be looked at until the game is actually finished first.</u>

Documentation

Much like I did with Pyoro 64, I would like to document the entire game development process because making a Nintendo 64 game requires a lot of thought put into the engineering of the overall project. There's a lot of hard decisions that need to be made, and I'm certain that there are a ton of people who are highly interested in the magic behind lots of retro games. As a result, I have decided to, while working on this game, document the game's development as a series of videos.

Some of the benefits of documenting the game as a set of videos are:

1) Videos are much more interesting to watch than just reading a huge set of documents.
2) Videos allow for a more dynamic way of explaining things. For instance, if I have to explain the decisions behind the memory structure of the game, this is infinitely easier to do in video with narration over a dynamically changing graph, versus having lots of images with probably a sentence explaining each diagram.
3) Building up an audience on YouTube is great advertising for the project you're actually trying to make. See: Yandere Simulator (God I really hope this project doesn't turn into development hell)...
4) If I build up a sort of following, making update videos is a fantastic way to force myself to actually commit to this project.

That being said, **video editing is an incredibly time consuming task**. I am making a game first and foremost, that is my priority, so I am not going to commit to a video schedule. I'll make a video when I find that there is something interesting to discuss about the project's development.

If only I wasn't a student, I could hire an editor instead.

Project Funding and Release

It’s been said multiple times that this game will be free and open source. This is a game for the Nintendo64, 25+ years after the console’s release, so I’m not expecting to make much money. I am starting this game completely solo, and if people are going to be joining the project (be it artists or whatnot), I can’t really pay them because I am a student, I don’t have a salary myself. That being said, if any money is made with this project, they will be distributed between project participants. Any money which is made **will be transparently disclosed**. I will not budge on any of this.

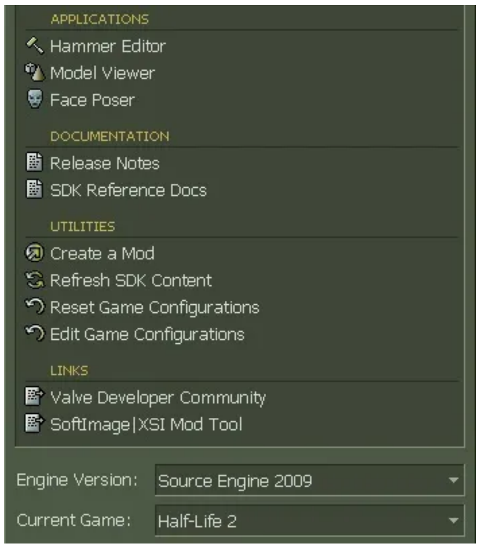
In the off chance that I do build a sort of YouTube following, if I decide to put advertising on the videos, all this money will be disclosed and go into a fund for this specific project (so that artists can be paid and whatnot).

Once the game is finished, the ROM goes on the internet for anyone to download for free. Ideally, the game can also be released as cartridges as well, which if any profits are made, can be distributed between any contributors. Whether a cartridge version of the game will contain extra content (as a thank you to people who purchased it), and whether said content will eventually also be made free for other people, is something I still need to think about. If someone is going out of their way to pay for something that they could otherwise get for free, I feel like I have to do something extra as a “thank you”...

Developer Tools

Considering that this is a Nintendo 64 project, a lot of tools will probably need to be made that are designed explicitly for my project. These tools will also be made free and open source, as I want to allow anyone to be able to pick up my game engine and start throwing assets into it. That being said, because of how little resources you have on the N64, I intend on these tools being specialized for my game/engine. I am not interested in feature scoping them beyond what is needed for this project.

I've designed quite a few tools for N64 stuff myself, but for this project I intend on having an entire tool suite. Similar to how you launch a source engine SDK and you get greeted with a map editor, model viewer, etc…



Engine Version: Source Engine 2009

Current Game: Half-Life 2

Because I want all assets to be bundled into a big binary blob, a tool suite will probably be necessary so that all of the game data can be read and interact with one another. For instance, models will need textures, so having a separate tool to import, view, and compile textures will be necessary. Now, the model viewer can use the information from the texture editor to help out with previewing models, among other things.

Since I am primarily a Blender user, I will provide plugins for exporting models, characters, animations, or maps. I can base this off my work on Sausage64. Probably the biggest hurdle, however, will be making sound tools. The ones we have available for the N64 are not very intuitive and/or useful without proper hardware. Sound is one of the last hurdles in making N64 development accessible, so ideally the sound software should not be too tightly integrated for my game, so that it can be usable for other homebrew projects.

I expect to need the following tools:

- Texture atlas - A software for importing textures, setting them up for the N64 (like combine modes, texture modes, etc…), setting up texture animations, and converting them into binary format.
- Model atlas - Import character models and props, set them up with the textures, apply specific properties (like billboards), and preview animations (as well as set up animation events).
- Audio atlas - A tool for importing sounds and performing whatever crunching needed to get it to work on the console.
- Particle editor - A tool for making particle effects from the imported textures and sounds.
- Map assembler - Import in the level geometry, set up collisions, place objects, and possibly playtest things like camera placement and whatnot before moving onto the actual console.
- Cutscene director - A tool for putting cutscenes together.

The tools will probably be written in C++, with wxWidgets and OpenGL. I want them to be cross-platform, so that they can be used irregardless of one's operating system.

All the game assets will be packed into a custom binary format, with some sort of compression (such as zlib).

Building a Community

I am not really interested in making a Discord server specific for this project and whatnot. I simply do not want to moderate a community, or have to deal with infighting because someone said something mean in the offtopic channel. Not only that, Discord is kinda against the whole "openness" of the project, due to the fact that Discord is a proprietary chat program which cannot be indexed by search engines, requiring you to have an account to access.

I intend on making a thread about this project on knockout.chat, a welcoming gaming community which I have called home since 2010 (back when it used to be Facepunch forums). That's honestly as much reach as I'm probably going to do this early into the project's life.

Any discussions about the actual project itself should happen on GitHub's Discussions. It might not be as fun or laid back as Discord, but it serves its purpose (that is, being a place where the project can be discussed constructively, and openly).

I am always available on Discord via N64brew, and this server will serve as a good jumping off point as well, but I'd like to reiterate that I would prefer any project discussion to be done on GitHub.