

Tools of the Trade

Antoine Racine-Gingras

ID: 40008247

Presented to Mr. Jonathan Lessard

Concordia University

Game Studio

CART 415

November 27, 2019

### First iteration: Protest-like approach to making a J-RPG

The initial design problem I wanted to tackle was to **make RPGs less frustrating, less repetitive and encourage creative thinking as opposed to pure tactical choices**. The design process of Tools of the Trade started with this initial frustration towards traditional J-RPG design, and the many elements that are still used today in modern western and Japanese RPGs. Indeed, in the past few years, I decided to delve into the Dragon Quest series. Dragon Quest I is actually the first console J-RPG ever created and the formula has not evolved that much throughout the years. The issue is that if you look at other RPGs, even modern ones, they take many traditional elements from Dragon Quest.

*For instance, here is a non-exhaustive list of elements present in most RPGs that could quickly become frustrating for a lot of players: random encounters, level grinding, repetitive battles (using the same physical and magical attacks for every single monster with very few exceptions), equipment and weapons incrementally better as the game progress instead of being used in different interesting ways, repetitive and meaningless quests such as “collect 10 things”, lots of useless “filler” dialog with NPCs (non-player characters), numbers game during battles (the only goal is to inflict the maximum amount of damage until enemy’s health reaches 0) and no emotional attachment to the enemies, as they often lack any personality aside from their design and animations.*

My initial goal was for my game to be a form of protest against those traditional elements by making an RPG that purposely avoids all of those aspects. While full-fledged RPGs are usually considered too ambitious to be undertaken by one person alone, especially when time to work on them is limited, my initial objective was to limit myself to one town and one dungeon, and to make sure I deliver a concise experience devoid of filler content and repetitive gameplay.

To achieve that, I had the idea of making the gameplay revolve around collecting various tools with different purposes. This is where the name “Tools of the Trade” came to mind. Thus, the player would have to find creative ways to use tools to defeat foes instead of simply mashing a button to attack them aimlessly. The goal was to offer multiple ways to approach battles, so I knew there would have to be multiple strategies to defeat enemies. Indeed, I did not want to make a puzzle/adventure game where you have to find *THE* tool that works. Instead, I would offer player freedom and expressiveness. For example, when fighting giant fish monsters, the player finds out that they are extremely strong when hidden underwater. To circumvent their hiding spot, the player has many options. The “brute force” method is to go swimming in the water and try to fight them where they are at their strongest. Obviously, this is the hardest way to go about the fight, but could offer a decent challenge to the ones that want it. Having this option in the first place and it being the hardest and most cumbersome to achieve makes the player want to explore the other, easier options. Those include using the fishing rod to fish them out of the water and force them to fight on land, using the bow and arrow to shoot them from land so that the player cannot be reached by their melee attacks as long as they stay underwater, and using the icy sword to freeze the water and the fish with it.

From the ground up, the design diagram I produced (see appendix A below) focused on the tools themselves and their use when fighting enemies. They would act as a solution to my design problem by making the game less repetitive and remove the frustration of failing battles just because you did not grind random encounters enough. Instead, the player would have to find a tool that can sensibly be used against the foe, taking into account its personality and physical characteristics. The reasons why a tool works on an enemy should always be deductible by conversation with the enemy (e.g. goblins tell you they come from cold mountains in the north so they are never in contact with heat and fire) or common knowledge (e.g. fish are strong underwater).

## Second iteration: Shifting focus away from J-RPG to an Action/Adventure game with RPG elements and a strong focus on music

With the ideas above in mind, I started searching for models (3D meshes) that would allow my vision to come to life. I also delved into Unreal Engine 4 and started exploring the many possibilities it offers. When considering what could and could not be possible to do in the timeframe available, I quickly convinced myself that the way I had thought about tools was not the right way to go about solving the initial design problem. Indeed, I would either have to design and code an entire turn-based battle system from the ground up to allow the use of tools, or make a real-time action game with RPG elements that would include so many mechanics (one or more for each tool) that it would take months to finish.

Thus, I settled on a new design objective that would incorporate some aspects of the first design problem: **Use hearing instead of sight and perception to make the player figure out an enemy's strengths and weaknesses.** As someone who studied music before settling on programming and video games as an eventual professional career, I have always been curious to explore music in games as an essential tool for gameplay instead of as a backdrop to the action with no real gameplay involvement. Even rhythm games usually do not require players to identify sounds and whether or not they fit in a given situation, instead focusing on the timing of when those sounds are/should be played.

The tools mechanic designed to circumvent repetitiveness and frustration in traditional role-playing games was thus transformed. Now, the generic "tools" that could essentially be anything (fishing rod, shoes, a glass, a lamp, etc.) were streamlined into regular weapons (swords, maces, hammers, etc.) fitting assets I already found. To distinguish them from one another, each tool would have an instrument and a melody associated to it of type "question". Furthermore, the weapons could be enchanted with various elemental effects like water, lighting and fire (using Unreal's particle effects system). Each enchantment would also be associated to a different melody, this time of type "answer". Finally, the player's chosen class would be associated with a particular chord progression.

The "question/answer" mechanism is used in classical music, notably in Bach's fugues. This is a process where the first voice (or "melody") raises a question and the second voice answers it right after the first one established its melody. The result is 2 voices sequentially taking the lead and answering back to each other.

In the game, using different combinations of tools (weapons) and enchantments would lead to different question/answer combinations and would be backed differently with chord progressions that depend on the class chosen. Initially, I wanted all 3 of those elements to create the battle theme. The gameplay would revolve around finding the right combination of sounds to affect the enemy.

I quickly realized how ambitious this idea was and how hard it was to implement concretely, but I had the feeling I was onto something. To put it simply, I would have to compose different questions and answers melody lines and make sure they could all answer back and forth to each other, as well as designing them to fit with different chord progressions. This was not possible for one person to do alone in such a small time frame. I asked my friends for help, some were musicians, the others gamers, and we all had a hard time finding the formula that would work.

However, as you can see in Appendix C, even with those ambitious ideas in mind, I was still convinced that the right approach to implement this was through a full-fledged (but minimalistic and simplified)

RPG. The gameplay loop described in Appendix C is reminiscent of one in RPG games, even though I clearly stated that I wanted to defy traditional RPG norms.

The ideas related to tools and music were shown to the whole class during the 2<sup>nd</sup> presentation. In my opinion, it did not go well at all, as the overall consensus seemed to be that my ideas were not well defined enough (i.e. too vague) and far from being concrete and understandable by an audience. I was told that it seemed like minimal progress had been made since the first iteration. This was honestly a terrible experience, as I felt like presenting a game idea in front of the class was a way of exposing a part of my own self, especially since my ideas combined the 2 things that I am the most passionate about in life (music and video games). I had put my heart and soul into my design ideas but had felt to deliver something to show the class. With that prospect in mind, and after a plethora of emotions, I knew I had to see Mr. Lessard to re-evaluate my design ideas and streamline them even more, as, in this case, the only solution to disappointment is hard work.

### Third iteration: Concrete application of the design objective into a playable game

My meeting with Mr. Lessard was fruitful, as we managed to define my idea in more concrete terms. Indeed, I was able to state my design objective precisely and through his suggestions, we established a concrete gameplay loop.

Every fight would revolve around finding the enemy's weakness. Instead of being visually identifiable, the weakness to a certain weapon would have to be found by listening carefully. To increase interactivity, instead of simply having a melody associated to each weapon that plays when using it, there would also be random notes being played when the player swings it. Those notes would be in different scales depending on the weapon (C Major for the sword, A Minor for the dagger, etc.). Thus, the goal would be to find the weapon that produces sounds matching the backing track associated with the foe.

The gameplay would revolve around trying out different weapons against a foe until you find one that best fits the backing track produced by the enemy. By swinging a given weapon, random notes in a scale would play. This would be accompanied by a melody automatically being played and looping with the same timing as the backing track produced by the enemy. This melody would have to always be in the same scale as the random notes being dynamically played. Understandably, if the melody and notes are not in the right scale, a dissonance will happen and it will sound cacophonous, a cue that this is not the right weapon to inflict damage to the enemy.

One thing we would like to try and avoid is having players use trial and error. This would be a case where they try out each weapon quickly until they find the one that inflicts damage. To circumvent this as much as possible, there will be no perceptible cues indicating that the player is hitting the enemy with the right weapon until a certain threshold of health is reached. This forces players to rely on their ears rather than visual cues. However, to compliment the satisfaction of finding the right tool to use on a given enemy, when its health goes below a certain threshold, it will start reacting to being hit (hit stun), and run away from the player. This is a clear indication that what you have been doing actually works and that you have found the enemy's vulnerability, hopefully using your ears!

Another important point is that the enemies should not put too much pressure on the player, unlike traditional action/adventure games. Indeed, since the player has to pay careful attention to what they hear and experiment with the different weapons available, they have additional concerns on top of avoiding enemy attacks and predicting AI behavior. Thus, I designed the simple AI so that it only attacks seldomly and adopts a somewhat predictable behavior. This adds some kind of pressure on the player, but not to the point where they are entirely focused on the enemy movement and attacks instead of what they are hearing.

Initially, the only instrument used was a MIDI piano, as it seemed to be the one that is the most easily scalable. Indeed, to be able to shift a given melody in different scales quickly and easily, MIDI data is used. The only thing required is to take the notes and transpose them into another scale, without changing the instrument or other settings such as compression, reverberation, EQ, etc.

### Prototyping the game to teacher, students and friends

When prototyping that playable demo to friends, students and Mr. Lessard, my biggest concern was validated. Indeed, I feared that using MIDI piano for everything would make sounds less distinguishable. In fact, Mr. Lessard could not even hear the random notes dynamically played when swinging a tool/weapon, because of the piano playing the backing track and the other playing the melody line. Thus, it seemed clear to me that I had to find MIDI libraries other than the piano one, in order to distinguish weapons between each other, and the different instruments playing synchronously.

On the other hand, I was extremely happy and satisfied to realize that all of the people playtesting my prototype were able to find the right weapon to defeat the first enemy, even someone who specifically warned me that music and games were not his strong suits.

The third iteration, which was in the form of a 5 minutes game pitch, came quickly, before I had the time to implement all of the feedback I got from prototype testing. Indeed, implementing different MIDI instruments would take a bit longer, but this time, the idea was fully implemented in a way that students could properly react to it. The comments I got from the presentation were in accordance with my own concerns and ideas. For instance, the idea to have battles involving more than one enemy came to mind, but I quickly discarded it, as either being too complicated to implement or just plain useless. Mixing different backing tracks together just sounds like a convoluted idea that complicates things for both the player and the developer. Another comment was that the musical track selection was not ideal for a battle. Indeed, a smooth piano piece does not usually come to mind when we think about an epic confrontation against an evil skeleton foe. This is purely a choice made to simplify playtesting. This brings us to the state of the design and the potential future developments of this project.

Since then, I managed to implement multiple MIDI instruments, making sure that each sound currently playing is from a different instrument in order to make it easy for the player to distinguish the different tracks. This was done by associating a different instrument for a given weapon's melody line and random notes. For instance, in the case of the sword, one of the musical pieces has it play the melody line using strings (violins, cellos, etc.) and the random notes using a guitar-like instrument. While I did not have time to make other people try the game with the different instruments, I personally feel like this really helps to distinguish the different tracks playing at a given time, making the game more enjoyable as it is more intuitive to recognize dissonances and harmonies.

### Summary of the current state of the project and potential future developments

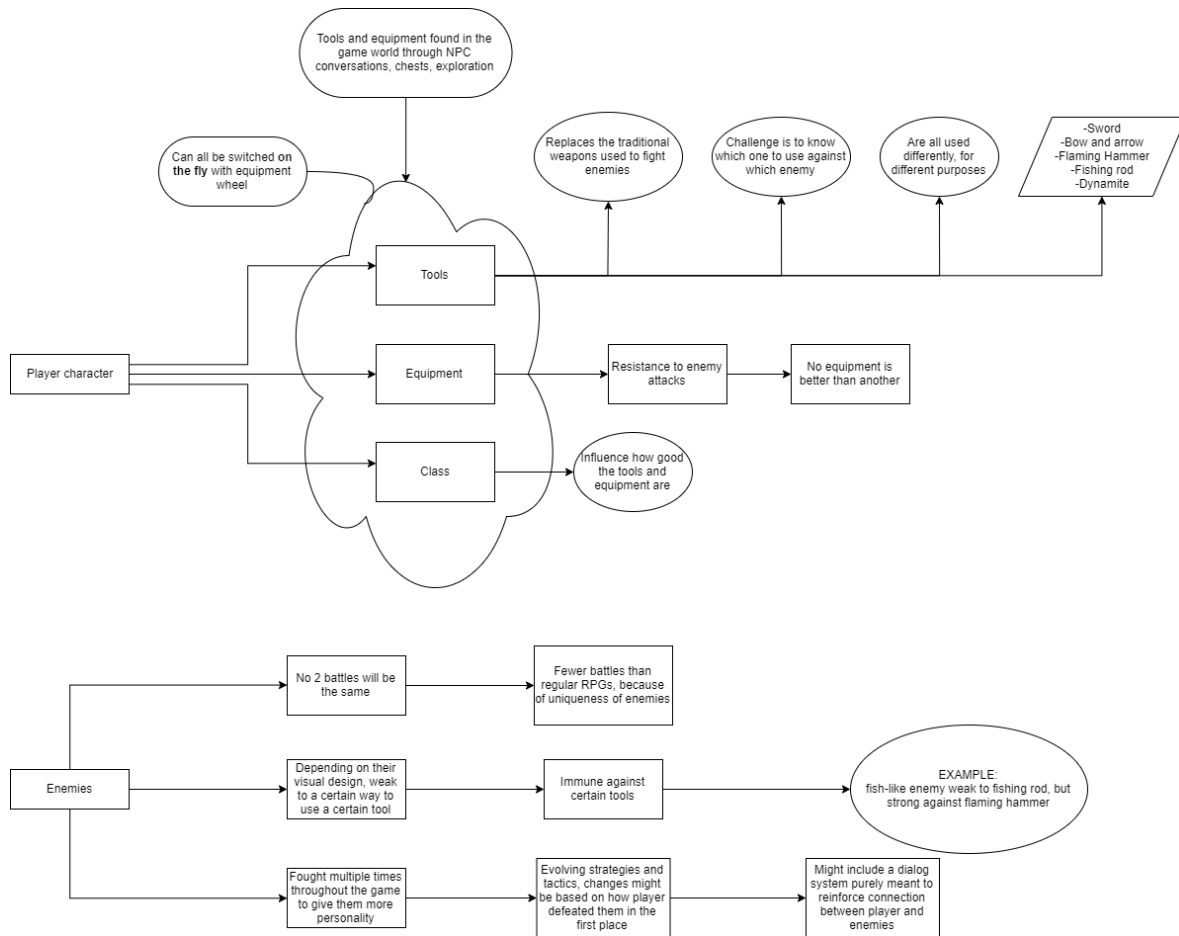
Considering the reception I got from friends, students and the teacher, I feel safe to say that I partially achieved my initial design problem and truly achieved the new design objective I set during the second iteration.

Indeed, my final idea addressed the frustration expressed in my initial design problem of playing games that felt too repetitive and too similar to previous ones by bringing an entirely new idea to the table (to my knowledge) in the form of musical expression in an action/adventure game. While I did not make an RPG that would challenge 30 years old conventions like I originally intended, I did challenge some conventions of action RPGs/action-adventure games by adding the sense of hearing as a way to detect an enemy's weakness. Where most games rely on visual cues (i.e. sense of sight), I allowed players to explore their sense of hearing and to detect dissonance. I laid the groundwork of a new style of games where the identity of an enemy is defined by the backing track it plays and the player's ability to understand it and adapt its own tools to its enemy's.

In the future, I would like to expand the game by making music pieces that feel more like battle themes instead of calming music. Also, just like I discussed with Mr. Lessard, it would make battles much more interesting and last longer if they would be divided into different phases. After each phase concluded with a given amount of damage dealt with the right weapon, the backing track would go into its second part (kind of like from verse to chorus), requiring the player to change weapons multiple times to defeat the enemy. Expanding this idea further could even lead us to test the player's memory by having a final phase to every battle, where the backing track changes often between the different backing tracks heard during the previous phases.

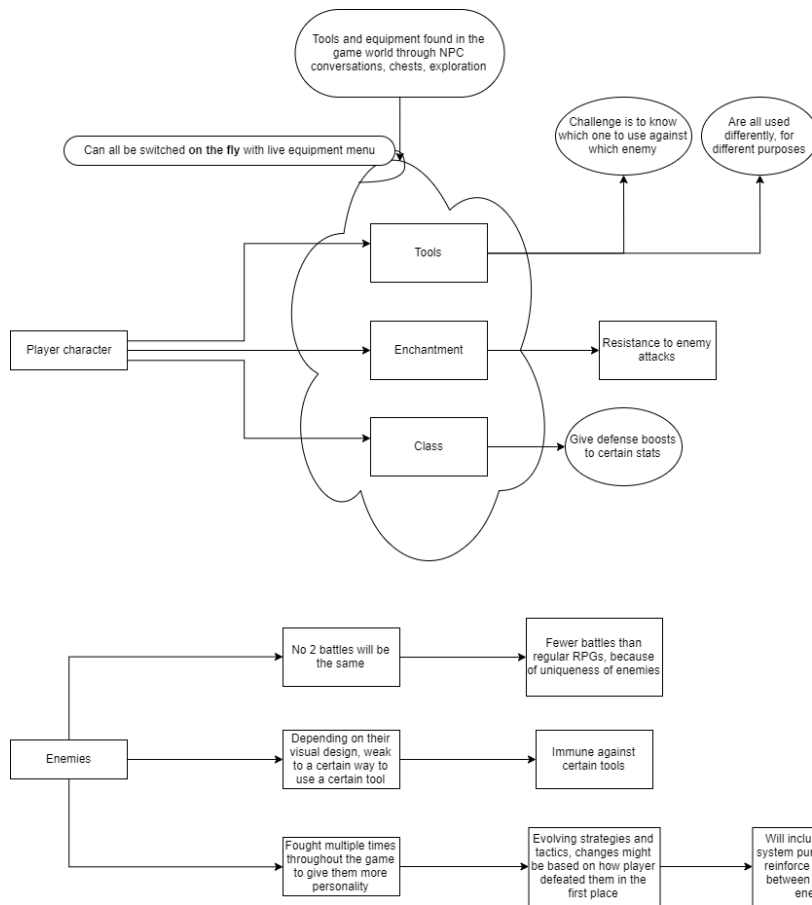
One of the traditional aspects of my game relates to the way you initially obtain the tools/weapons, by simply picking them up in the fields or by opening chests. I think that it would be much more interesting to use the fact that already acquired tools produce sounds to design puzzles centered around music and dissonance. For instance, to unlock the mace, the player would have to play a game similar to "Simon says". When the player touches a stone, a distant melody can be heard in a specific scale. Then, by swinging the tools the player already owns, they try to find the one with the scale corresponding to the stone's scale. Once they hit the stone multiple times with the right weapon, the mace is given to the player.

## Appendix A: Initial design document





## Appendix B: Second iteration's design document



## Appendix C: Second Iteration's Gameplay Loop

