

**Final Game : Design Document**

**Crescendia: Harmony and Dissonance**

# **CRESCENDIA** **-MAGICAL MUSIC GIRLS-**

**Developers:**

Rico Balakit  
Aaron Ayrault  
Jackson De Jong  
Rupen Sakariya  
Jack Demaree  
Daniel Croy  
Saifil Ali

## **1 Introduction**

In Crescendia, the world takes place in a land where magic exists, and where magic and music are one in the same.

## **2 Description**

Where there is music, there is magic in the world of Crescendia. The "science" behind how the magic works and interacts is based upon the rules of music theory too, especially with the interaction between different 'keys' of magic, corresponding with the same relationships between keys in music. The game borrows from the Japanese anime genre of "Magical Girls" too, which primarily feature girls in very adorable battle outfits making use of fantasy-style magic to fight for justice.

The game is a turn-based strategy where your one to four character units are created out of songs you choose from your music library. Songs uploaded from local file storage (or as a stretch goal—retrieved from Soundcloud or Spotify) are analyzed, and characteristics of the songs such as tempo, key, length, and frequency distribution are used to generate statistics for your playable units to fight with. Attacks and Actions are derived from sections of the song (individually analyzed for each section) and the player is given a choice of what portions of the song to use as the attacks. When these attacks or actions are performed, the sample of the song plays.

There are four main types that can be combined together when creating characters, these are offence, defence, status, and heal. When these types are combined there is a total of ten different unit types (Figure 1). In addition each character will have a primary type, this will determine if they will have more special moves of that type than their other (i.e. A Knight with a primary type in defence will have three defence moves and two offensive moves).











Offensive	Defensive	Status	Healer	
<b>Gladiator</b>  00	<b>Knight</b>  0D	<b>Assassin</b>  0S	<b>Crusader</b>  0H	Offensive
	<b>Guard</b>  DD	<b>Warden</b>  DS	<b>Paladin</b>  DH	Defensive
		<b>Trickster</b>  SS	<b>Alchemist</b>  DS	Status
			<b>Priest</b>  HH	Healer

Figure 1  
Combination sheet of all classes based on their types

### 3 Key features

There are two major plotlines within Crescendia, and each plotline defines the two major modes of gameplay. In the Single-player mode, there is "The Darkness" that seeks to steal all the power of music from the world. "The Darkness" is meant to be a mysterious but clearly evil force in the world. After stealing music, The Darkness encounters the player in battles and the idea of the single-player mode is to "liberate" stolen music from The Darkness and return it to the world of Crescendia. The Darkness represents no genre and no movement, it is, if anything, closer to a natural disaster than a scheming person with ulterior goals. This is by design to

prevent the player from empathizing with The Darkness, or becoming confused as to why they should oppose The Darkness.

The other major plotline within Crescendia is the idea of a battle for power. During the initial previews of gameplay and introduction of our game in development, we encountered that while many people thought the gameplay itself was interesting and well developed, the incentive or excitement to play the game was less than we hoped for. However, we conceptualized later the concept of creating an inclusive guild system, where a guild would be created by a player with a significant amount of experience, and any player can join the guild. We called these “Record Labels” to go along with the music theme.

As such, the second major plotline involves Record Labels clashing for power between each other within Crescendia. While we plan to evolve the system from a simple leaderboard to a geographic-based metagame as a possible stretch goal, the overall idea of the plotline is these record labels are battling for power of the world of Crescendia. By allowing these Record Labels to be inclusive (Label owners do not have power to add, approve, or remove players. They only really have the ability to name, rename, and claim to own their labels), players can identify with their preferred music scenes. We figured it would be very exciting to see battles of power between:

- Rock fans and EDM ravers
- One Directioners vs Beliebers
- Meek Mill fans vs Drake Fans
- NuVaporSpaceCore vs Tropical Ambient Shoegaze Rock.

Primarily, the two functions of Record Labels is for overall control of Crescendia, as well as a way for two specific record labels to directly go “at war”.

#### **4 Genre**

The game will be a turn-based strategy that will have a classic steampunk fantasy J-RPG aesthetic. Magic in the game is created from music. The art style will be anime chibi style. Most of the in-game music will be ambient/soundscape tracks,

so as to not clash against the user-selected tracks. A very important part of the game is for the player's musical taste to be shown off, especially considering it is a metaphor for a battle of musical tastes. As such, the player's selected music is at the forefront of the sound aesthetic. Gameplay sounds are meant to be non-intrusive while letting a player know about the status of their attacks and rhythm. The game will not appear gritty as this is a turnoff to many potential players. Vivid colors and effects will occur as the music magic attacks occur.

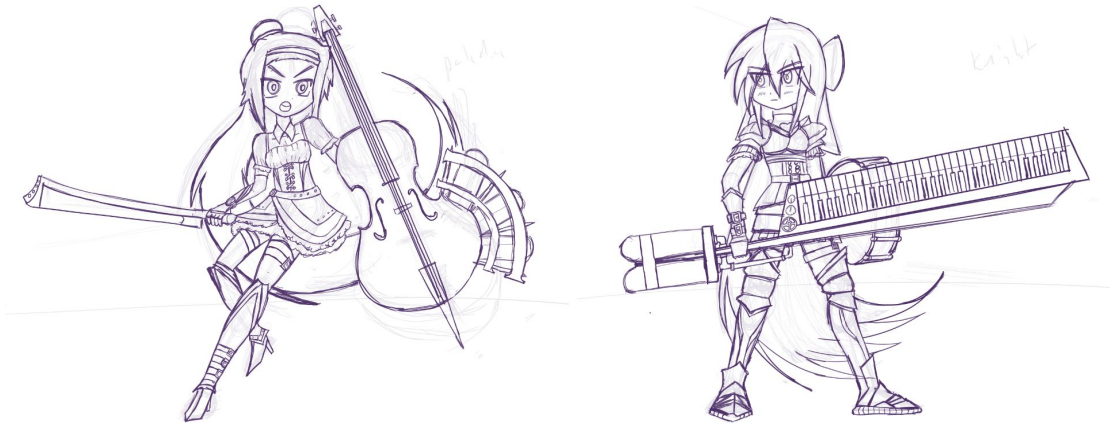
## **5 Platform**

The game will be developed for mobile platforms, Android and iOS, and will require an internet connection in order to retrieve songs, send local song uploads, and retrieve information regarding the analyzed statistics/characteristics of songs. The game will likely require two thumbs to play.

The game will embrace microtransactions as a method for customizing unit aesthetics (such as hairstyle, costumes/clothes) as well as player avatars and name-tags and fonts and other cosmetic features. Experience points may also be purchasable through microtransactions, although by no means should the game become imbalanced, "play to win". The unit level caps should be attainable in a speed that makes manually playing the game to gain experience (and learn how to effectively play the game) more desirable than purchasing experience. In addition, no items which provide actual bonuses in combat shall be purchasable with real currency.

The game will need to rely on a central server holding information about songs, their possible attacks, and their base stats. This is so all units from the same song hold consistent stats and actions without regard to if the song was a local file, from Spotify, from Soundcloud, or is a perfect FLAC file or a 128kbps YouTube rip. This is likely done with metadata matching. The music analysis is also performed server-side rather than mobile-side.

## 6 Concept art



Figures 2 & 3

Concept art of the Paladin class (left) and of the Knight class (right)



Figure 4

Rigging concept art for creating and animating characters in game



Figure 5

Basic layout of in game battle (minus UI features)

## **7 Market Analysis**

According to newzoo, the global video games market currently makes almost 100 billion dollars annually, this number is rising to 106 billion in 2017 [Global Games Market]. Additionally, currently 27% of this market is brought in by smartphone games with its percentage soon to surpass PC and Console game sales and become one third of the overall market. Meaning that smartphone games will generate over 42 billion dollars in 2017. In addition a statistical analysis conducted by BigFish in 2014 shows that 11% of games bought on mobile platforms are Turn-Based RPGS, this means the market for our where Crescendia would be in makes around 4.6 billion dollars a year [Most Popular Mobile Game Genres]. Lastly, the game we far following the design and sales plan of, Love Live! School idol festival, currently makes \$8,294 daily revenue in the US alone [Love Live! School idol festival Revenue] and is frequently in the top 200 in grossing games. It is with the success of other games similar to ours and the Total market share of the Mobile Turn-Based RPGS being over 4.6 billion dollars that makes us confident that Crescendia will a succeed.

## **8 Technical Analysis**

Crescendia will be developed using the Unity engine, available freely from Unity Technologies. The game will be built in 2.5-D—it will look like a 2-Dimensional game for most gameplay, but will sometimes incorporate 3-Dimensional aspects, such as during the battle or character creation phases. We will be using a PC development platform, to be later ported to IOS and Android mobile devices. We expect a development timeline of roughly two and a half months, detailed on a week-by-week basis below. The analysis portion of the game will be done through the use of multiple third-party coding libraries, allowing us to utilize whichever language API each of us is most comfortable with to analyze the player's songs. For song analysis and battle computations, the game will most likely exist in a client-server relationship with a dedicated game server, which will do the computations once, and then store the results for future use in order to speed up performance. Multiplayer functionality will be developed concurrently with

single-player interaction, as each “team” within the game will exist as separate modules without distinguishing whether the team is interacting with an AI or another player.

Communications between the server and client will be handled by a series of RESTful API endpoints. A REST architecture allows virtually any front end system to access our back end via the same type of HTTP requests no matter what language the front-end or back-end is implemented in. The communication is only send-and-receive, however, and is not asynchronous. Our back end will only send data to a client when asked to. Therefore, the game is designed in such a way that this is a boon rather than a problem.

Side Effects and benefits of this system include the ability to request relevant data on a song that has been analyzed before lightning fast and in a relatively easy manner, and also facilitating turn based gameplay that happens upon a preset interval.

We expect to write the back-end in Python, as this language facilitates getting systems up and running quickly. The Flask library for Python will be instrumental in assisting our efforts with a RESTful API.

We will be using the Unity engine to handle our front-end application. With the engine we will implement the graphical aspects of the game from sprites to animations. In addition it will handle client side game logic in C#. The engine will make server calls to update the turn based combat system. To publish our app, Unity offers simple methods for deployment to mobile platforms.

## **9 Cost and Revenue Projections**

Current cost for Crescendia has been limited to voice acting and server upkeep for the game, totalling in \$85, where voice acting costs \$55 and the current server costs \$30. The server so far is mainly for testing and when the app is launched we will need to upgrade to hold the additional data of all the users joining. Meaning that the server costs will be depend totally on how many users are playing the game and how many are users it again after day one. For starting off in the beta we



will cap the number of users to just 100, and then through each iteration of the beta we will increase the cap by a multiple of 10.

According to the Financial-Modeling website, cost and revenue projection is based on User Base-starters, New User, User Loss, User Base-usually ending period. We can forecast(revenue) the user daily playing activity by (app downloaded x new user). We can also gain revenue by other means as well such as keep game store where user can buy coins or some gifts in exchange for some dollars and we can also keep advertisement video in between the games to aware use about our premium app. We also should keep in mind about all types of scenarios.

Lastly, if the project turns into a fully fledged game making positive profit then additional programming staff would then need to be hired to maintain and update the game from time to time. Meaning that for each programmer added would be a cost of around 70k per year.

## **10 References:**

Global Games Market, April 21, 2016. Newzoo

<https://newzoo.com/insights/articles/global-games-market-reaches-99-6-billion-2016-mobile-generating-37/>

Most Popular Mobile Game Genres, 2014

<http://www.bigfishgames.com/blog/stats/most-popular-mobile-game-genres/>

Love Live! School idol festival Revenue, December 9, 2016, Think Gaming

<https://thinkgaming.com/app-sales-data/7510/school-idol-festival/>