

# CIS400/401 Project Proposal - Designing Rhythm Game Simulators for Touchscreen Devices

Dept. of CIS - Senior Design 2010-2011

Philip H. Peng  
pengp@stwing.upenn.edu  
Univ. of Pennsylvania  
Philadelphia, PA

Stephen H. Lane  
shlane@cis.upenn.edu  
Univ. of Pennsylvania  
Philadelphia, PA

## ABSTRACT

*As tablets and other touchscreen devices become increasingly common, consumers are expecting new interaction models that fit the new touch-driven input paradigm. Application interfaces need to be redesigned such that their methods of interaction must feel "natural" to users as well as contribute to or enhance the apps usage itself. This is particularly important for games and the currently under-represented rhythm game genre. If designed to take advantage of this shift in input paradigm, rhythm games can turn touchscreen technology into an integral part of gameplay, leading to enhanced, more immersive user experience.*

*There are two main challenges in designing rhythm games for touchscreen devices. The first challenge is designing an touch-centric interface that is easy to learn and use without hindering user experience. The second challenge is designing gameplay modes with interaction models that make touch-input feel natural without detracting from the timing element of rhythm games. This project will address these two challenges through the design and production of a touch-based, multi-mode rhythm game simulator designed to run on tablets and other touchscreen devices.*

## 1. INTRODUCTION

Rhythm games are a genre of music-based games in which the player performs actions in response to visual and audio cues. Rhythm games focus on both pattern recognition and fast hand-eye coordination. In most rhythm games, the patterns correspond to the beat or rhythm of songs and the player is rewarded on the timing accuracy of their actions through a point-based system. Players are inclined to repeatedly challenge the same song level to improve their previous scores or the scores of others. Rhythm games are only "complete" when the player has achieved perfection on all song levels on all difficulty levels - a feat that is often made extremely challenging but still with a glimmer of achievability.

The main goal of rhythm games is to provide a highly immersive experience that challenges the player's sense of rhythm. Rhythm games are typically designed to take advantage of hardware in a manner that enhances and heightens the player's senses (hearing, sight and touch) and subsequently their timing accuracy. In traditional cases, this is accomplished by creating rhythm games designed to be played

with specialized hardware, such as custom arcade game machines or video game console peripherals. For most gamers, however, going to an arcade or buying special controllers is an issue of inconvenience or inaccessibility. Touchscreens, however, are becoming increasingly common and will soon be the de facto input mechanism for personal electronics. By designing rhythm games for touchscreen devices, this genre of games will become more widely available to rhythm game enthusiasts as well as casual gamers.

When creating rhythm games for touchscreen devices, the focus is the impact of the touch-driven input paradigm when designing the game menu interfaces and the gameplay modes. A *touchscreen device* is an electronic device that receive user input through usage of a finger or stylus. The *touch-driven input paradigm* is the input model encompassing the various gestures and actions humans will naturally try to use when interacting with a touch-driven device. The *game menu interfaces* of rhythm games are the series of menu screens and their interaction models that support and/or lead up to the main gameplay. The *gameplay modes* of rhythm games are the methods of rhythm pattern representation and their corresponding interaction models. The *pattern representations* are the visual objects that represent the song's rhythm and beats. The *gameplay interaction model* is defined by the combination of audio-visual and haptic (AVH) elements of the gameplay linked to the actions performed by the player.

In this project, an open-source, multi-mode rhythm game simulator will be developed with touch-driven input paradigm in mind. The game menu interfaces and gameplay modes will be designed to take advantage of touchscreen technology as well as other features of such touchscreen devices. The result will be an expandable, cross-platform rhythm game simulator readily available to the increasing population of touchscreen device owners.

## 2. RELATED WORK

If one were to ask an average gamer today to name three "rhythm games" (or rather, the more broader, commonly-used parent genre, "music games"), most would list *Rockband*, *Guitar Hero*, and maybe *Dance Dance Revolution*. Deeper rooted rhythm games such as the long-running *BeatmaniaIIDX* series or even *Osu! Tatakae! Ouendan* (known as *Elite Beat Agents* in North America) are virtually unknown and unheard of by the general western population. In recent years, publishers such as Activision have introduced

a slew of music games that capitalize on the commercial rock and pop music craze in North America, subsequently increasing popularity of this previously lesser-known game genre. The general rhythm game genre, however, can trace its roots back to the late 1990's in Japan.

## 2.1 Rhythm Games in History

One of the best works documenting the various different rhythm games throughout history is Scott Steinberg's book, *Music Games Rock: Rhythm Gaming's Greatest Hits of All Time* [15]. While it also covers numerous music games outside the rhythm game genre (such as vocal pitch-based games) and other games where pattern matching is not the main focus, it gives a very comprehensive look at the trends and commercial products of the rhythm game industry over the years. The following is a collection of rhythm games and their significance in rhythm game history. A very comprehensive list of arcade and console rhythm games up to 2007 can be found on *Dance Dance Revolution Fanatics of Kansas City's* website [10].

### Parappa The Rapper [15]

**From:** Sony Computer Entertainment

**Released:** 1997

**Platform:** PlayStation

**Significance:**

*Parappa The Rapper* is often considered the first true rhythm game as it incorporated the element of rhythm timing accuracy into main gameplay. Upon hearing a sequence of patterns and a "rap challenge", the player must respond by pressing the PSOne controller's buttons to the beat. By pressing the correct sequence with the correct timing, the player is rewarded with a comical, flowing rap counter-response. This positive reinforcement system encourages the player to repeatedly play the same level until perfection is achieved. *Parappa The Rapper* "songs" are all light-hearted raps that directly tie in to the game's plot.

### Beatmania [15]

**From:** Konami

**Released:** 1997

**Platform:** Arcade, PlayStation

**Significance:**

The *Beatmania* series is one of the longest running rhythm game series still in production and played in arcades today. While the one international version ever released received little attention in the US, the *BeatmaniaIIDX* (standing for "Beatmania 2 Deluxe" but pronounced "Beatmania two-dex") franchise in Japan was wildly popular and resulted in the release of numerous arcade machines and PlayStation 2 ports. In the game, the player "mixes" like a faux-DJ, using five or seven keys and a turntable to trigger the playback of musical notes, represented by falling tabs. Hitting the corresponding button with the correct timing will reward the player with playback of a pre-recorded sound that enriches the background music. Missing a however, not only breaks building combos but also decreases the player's health gauge. Hence, passing a level requires both most notes to be hit and be hit accurately. In this case, a negative reinforcement system is used; players will try to improve their accuracy and score to both avoid losing prematurely as well as to hear the complete, properly "mixed" background mu-

sic. Because of this, the game is generally seen to have an extremely high learning curve and requires a lot of practice to master. *Beatmania* arcade machines are dedicated cabinets with large speakers, the five or seven keys arranged in a zig-zag piano layout, and the turntable, while the PS2 ports would require the usage of a specialized controller with the keys and turntable. The *Beatmania* and *BeatmaniaIIDX* series soundtracks feature a wide spread of genres and artists, much of which is original music. Most rhythm games developed following the success of *Beatmania* borrow many rhythm game mechanics and ideas from *Beatmania*, such as Pentavision's *DJMax* series and AmuseWorld's *EZ2DJ*.

### Pop 'N Music [15]

**From:** Konami

**Released:** 1998

**Platform:** Arcade, PlayStation

**Significance:**

The *Pop 'N Music* (abbreviated *PNM*) series is another one of Konami's rhythm games series and is very similar gameplay-wise to *Beatmania*. While given a seemingly children-appealing appearance with its brightly coloured interface and cartoon graphics, the game itself is arguably even more challenging than *Beatmania* due to its usage of nine keys and nearly impossible beat patterns at the highest difficulty levels. Like *Beatmania*, notes fall down the screen and dedicated cabinets or controllers are used. *Pop 'N Music's* soundtrack focuses mostly on upbeat pop, eurobeat and electronica music, mostly produced in-house but more recently including licensed anime songs.

### GuitarFreaks [15]

**From:** Konami

**Released:** 1998

**Platform:** Arcade

**Significance:**

*GuitarFreaks* was the original "guitar" game and made use of a plastic guitar controller with buttons representing frets. The gameplay is nearly identical to the later *Guitar Hero* games produced by Harmonix, but only uses three buttons instead of five. Unfortunately for Konami, the game did not sell well in Japan, probably due to its poor interface and plague of timing-synchronization issues.

### Drummania [15]

**From:** Konami

**Released:** 1999

**Platform:** Arcade

**Significance:**

*Drummania* was the original "drum" game and made use of a plastic five-pad drum kit and fake drum pedal. The gameplay is nearly identical to its partner game, *GuitarFreaks*, with the two arcade cabinets usually linked together. The game suffered from the same problems as *GuitarFreaks* and is rarely found in use today.

### Dance Maniax [15]

**From:** Konami

**Released:** 1999

**Platform:** Arcade

**Significance:**

*Dance Maniax* was Konami's attempt at a dance-focused

rhythm game that makes use of hand motions. The specialized *Dance Maniax* cabinets contain four sets of optical sensors that react to hands waving over them as the equivalent of button presses. Instead of *Beatmania*-style falling notes, *Dance Maniax* notes would scroll up to the top of the screen and all notes would be judged as a "hit" or "miss", disregarding timing accuracy. The game's soundtrack contains songs created in-house, borrowed from other Konami rhythm games, and some licensed from the *Dancemania* series of eurodance music.

### Dance Dance Revolution [15]

**From:** Konami

**Released:** 1999

**Platform:** Arcade, PlayStation

**Significance:**

The *Dance Dance Revolution* (abbreviated *DDR*) series is arguably the best known Konami line of rhythm games outside of Japan. Instead of using a controller or hands, a physical dance pad is used with four large arrow pads meant to be pressed down by the player's foot. Music notes would be in the form of up-scrolling arrows and performance would be based on timing accuracy. Unlike Konami's previous rhythm games, however, the full background music would play in its complete form regardless of arrow-hitting performance, but players would still lose early if their health gauge was depleted from consecutive missed arrows. *DDR* also introduced various new gameplay mechanics and modifiers such as holds, mines, non-stop mode, and more. Due to its highly physical gameplay, *DDR* was popularized not only as a rhythm game but also a form of exercise and sport in western countries, especially with its PlayStation home edition ports and relatively cheap dance pad accessory. Since all *DDR* cabinets were built with two distinct sets of pads and the game itself provided numerous two-player versus modes, the game is a competitive favourite in arcades. The game itself has inspired numerous similar rhythm games and clones, including Andamiro's *Pump It Up* series and Roxor Games's *In the Groove* series (which was later acquired by Konami). The *Dance Dance Revolution* soundtrack focuses mostly on dance music but also includes in-house Konami productions and other licensed music in different genres.

### Vib-Ribbon [15]

**From:** Sony Computer Entertainment

**Released:** 2000

**Platform:** PlayStation

**Significance:**

*Vib-Ribbon* is a unique cross of an action game and rhythm game developed by the creators of *Parappa The Rapper*. Gameplay is similar to side-scrolling action games such as *Super Mario Bros* except that the level designs and obstacles are generated from custom music that the player loads during runtime. *Vib-Ribbon* was one of the first games to transform arbitrary songs into dynamic game data based on the music's rhythm, a concept used in later simple rhythm games such as Dylan Fitterer's PC game *Audiosurf*, Harmonix's iPod game *Phase*, and Sony Computer Entertainment's PSP game *Beats*.

### Keyboardmania [15]

**From:** Konami

**Released:** 2000

**Platform:** Arcade, PlayStation, PC

**Significance:**

*Keyboardmania* is the last in the set alongside *GuitarFreaks* and *Drummania*. Its simulated instrument was a 24-key electric keyboard and its gameplay essentially resembled an expanded *Beatmania* remake for piano music. A PC version was later released but its lack of songs resulted in low popularity.

### Taiko No Tatsujin [15]

**From:** Namco

**Released:** 2001

**Platform:** Arcade, PlayStation, Nintendo platforms, Mobile platforms

**Significance:**

The *Taiko No Tatsujin* series focuses on simulating the Japanese Taiko drumming experience. Two large Taiko drums and drumsticks are used as controllers, with hits in the middle and taps on the rim acting as the two input types. With only these two types of inputs, gameplay is very simple with a stream of two-coloured notes scrolling across the screen to the target hit area. Low timing accuracy and misses will result in a lower score; however, there are no mid-song gameovers nor interrupted music flow due to missing notes. The *Taiko No Tatsujin* soundtracks typically include J-pop and video game music, mostly produced by Namco.

### Guitaroo Man [15]

**From:** Koei

**Released:** 2002

**Platform:** PlayStation

**Significance:**

*Guitaroo Man* was similar to *Parappa The Rapper* in the sense that each song level is introduced consecutively as part of a plot. *Guitaroo Man* was unique in its rhythm game mechanics through its involvement of direction in beat patterns; notes would fly in from all directions and the player would not only have to press the correct controller button at the right timing, but also use the controller's joystick to match the correct angle that the note is flying in from. Because of the notes could fly in from any direction, the player would be constantly engaged in scanning over the edges of the screen for the next note rather than focusing on specific locations of the screen like in previous rhythm games. The addition of spatial orientation to note patterns is used in later rhythm games such as *Osu! Tatakae! Ouendan!* and *Hatsune Miku: Project DIVA*. *Guitaroo Man* had very few songs, all of which were produced in-house.

### Osu! Tatakae! Ouendan! [15]

**From:** iNiS

**Released:** 2005

**Platform:** Nintendo DS

**Significance:**

*Osu! Tatakae! Ouendan!* (abbreviated *Ouendan*), known as *Elite Beat Agents* (abbreviated *EBA*) in North America, is one of the first rhythm games designed to be played on a portable touchscreen device (Nintendo DS). The notes patterns were represented as numbered circles and sliders fading in various locations on the screen. Tapping or sliding along the notes would not only increase score based on

your timing accuracy but would also play one of a handful of sound effect (e.g. drum tap, snare, whistle blow, etc.). In addition, many visual cues help aid the player in recognizing the note patterns, such as shrinking focus circles and guidelines between notes. Because the notes could appear anywhere, players would be constantly engaged in scanning over the entire screen for the next note rather than focusing on a specific locations of the screen like in previous rhythm games. *Ouendan!* only has three levels of timing accuracy but uses a constantly decreasing health bar, that replenishes on each note hit; on higher levels of difficulty, better accuracy is needed to survive the song level. In both *Ouendan!* and *EBA*, the soundtrack consists mainly of pop songs by various artists from the past two decades.

### **Guitar Hero** [15]

**From:** Harmonix

**Released:** 2005

**Platform:** PlayStation, XBOX, Nintendo platforms, Mobile platforms

**Significance:**

While the concepts and gameplay of the *Guitar Hero* series is far from original (predated by many similar rhythm guitar games, most notably *GuitarFreak*), it is best known for its strong cultural impact. The *Guitar Hero* series can be attributed to the boom of the music game craze in North America, starting a massive multi-billion dollar franchise. Rather than introducing new innovating content, however, each new version of the game had used the same gameplay mechanics with a very shallow learning curve (compared to other rhythm games) due to its general lack of variation as well as no notion of timing accuracy. The series' soundtracks were also nothing original; each version would feature songs from popular guitar artists of the era or even focus on one particular band. In addition, the playing of the game required the purchase of expensive accessories, most of which worked only with specific platforms and even sometimes only with specific versions of the game. With the introduction of downloadable content on the PS3, Wii and XBOX platforms, players started preferring downloading more songs over the internet as opposed to purchasing new versions of the game and their subsequent, costly hardware. The game's high costs, combined with the oversaturation of the music game market with numerous *Guitar Hero* versions with little or no variation in gameplay (as well as its competing *Rock Band* series, which used the same overall game mechanics), led to the recent collapse of the console music game market.

### **jubeat** [7]

**From:** Konami

**Released:** 2008

**Platform:** Arcade

**Significance:**

*jubeat* was Konami's attempt at a touch-based rhythm game. The arcade machine consisted of a grid of 16 touchscreen boxes that had to be tapped to the rhythm of the music, similar to "whack-a-mole". While the game was tested in the US and UK, under the name *UBeat*, it never matched the popularity of *Dance Dance Revolution*.

### **DJMAX Technika** [13]

**From:** Pentavision

**Released:** 2009

**Platform:** Arcade

**Significance:**

*DJMAX Technika* was one of the first arcade rhythm games to focus on a full touchscreen as the main input method. In *DJMAX Technika*, touch notes would appear on screen long before their timing point and a moving hitbar would scroll across the screen; users would have to tap/slide/hold notes when the hitbar reached the note itself. Like the rest of the *DJMAX* series, the game will simulate music mixing, with the player roleplaying as a "DJ". The soundtrack consists mainly of songs from the other *DJMAX* rhythm games.

### **Hatsune Miku: Project DIVA** [14]

**From:** Sega

**Released:** 2009

**Platform:** Arcade, PlayStation

**Significance:**

*Hatsune Miku: Project DIVA* was a rhythm game made to capitalize the popularity of the cultural success of Hatsune Miku, a Vocaloid2 character created by *Crypton Future Media*. Unlike most rhythm games, *Hatsune Miku: Project DIVA* first started as a PlayStation Portable game, and was then ported over to arcade machines. The gameplay involves pressing the correct PSP keys when flying notes overlap with countdown hitboxes, similar to *Guitaroo Man* but with the hitboxes appearing in patterns across the screen. The game's soundtrack consists entirely of songs written by Vocaloid2 producers. Unlike most rhythm game companies, Sega strongly favours and often uses community input in the game content, as well as including an in-game song editor.

### **Other Notable Music Games**

#### **The IDOLM@STER:** [15]

Not technically a music game but the inclusion of rhythm game elements in a vastly popular video game series brought the rhythm game genre to casual gamers of Japan.

#### **Lumines:** [15]

A cross of a tetris-style puzzle game with rhythm game elements that became popular with casual gamers. The game was also known for its smooth integration of audio and visual special effects in enhancing gameplay. It brought rhythm games to casual puzzle game players.

#### **Rez:** [15]

A rail shooter game which uses the rhythm of the background music as audio cues to the player. Through the use of AVH elements, the game and its spiritual Kinect successor, *Child of Eden*, explores the theme of synesthesia. It combined action games with rhythm games.

#### **SingStar:** [15]

The *SingStar* series and similar games (*Karaoke Revolution*, *Lips*, etc.) fall under the vocal/pitch games genre rather than rhythm games as the focus is on accurately replicating the melody of songs as opposed to their rhythm.

#### **Rockband:** [15]

*Rockband* can be categorized more as an overall general mu-

sic game. It is one of the most successful attempts at combining guitars, drums, and vocals into one coherent game; however, it doesn't contribute much to the evolution of rhythm games in terms of gameplay.

#### **Dance Central:** [15]

*Dance Central*, *Just Dance*, and similar games are the newest wave of music games in the gaming industry, with players replicating full-body dancing motions captured by the Kinect or PlayStation Move. While the timing element is still there, there is weak association of rhythm and a greater focus on body movements. Hence, these games would fall under a newer dancing game genre as opposed to the traditional rhythm game genre.

## 2.2 Rhythm Games Simulators

With most rhythm games only available in arcades and occasionally with gaming console ports, enjoyment of rhythm games was a costly endeavour. In addition, the song selection for those games are limited to what the creators decided to add; playing new songs would usually involve either hacking the arcade machines themselves or hoping that the next release would include better songs. Over the years, there have been many (mostly open source) rhythm game simulators written for the PC platform that simulate the arcade rhythm game experience with mouse and keyboard. The following are relatively successful rhythm game simulators that have been released, most of which are still actively under development. Work-in-progress information collection about various rhythm game simulators and their format details can be found on the Beats2 wiki [12]. A comprehensive list of rhythm game simulators up to 2003 can be found in the document "bemani and Music Game Simulator File Types", written by AJ 187 [1].

#### **StepMania** [2]

**Lead Developer:** Chris Danford

**Website:** <http://www.stepmania.com>

**License:** MIT License

**Simulates:** Dance Dance Revolution

**Significance:**

*StepMania* is one of the most well known rhythm game simulators for Konami's *Dance Dance Revolution* series. Gameplay is the same as in *DDR* with rising arrows being hit on time via keyboard strokes, but includes many additional modifiers not available in the game. *StepMania* includes a built-in pattern editor that saves note data in its own ".sm" format, but the game also supports simulator *Dance With Intensity's* ".dwi" format, *Kick It Up's* ".ksf" format, and many more. The *StepMania* game engine has been ported to numerous platforms and was used as the basis for the Fun In Motion's *Pump It Up Pro* and Roxor Games' *In the Groove* arcade games. *StepMania* is still actively in development, currently at *StepMania 5*

#### **BM98** [?]

**Lead Developer:** Urao-Yane

**Website:** <http://bm98.yaneu.com/bm98>

**License:** Freeware

**Simulates:** Beatmania

**Significance:**

*BM98* is one of the oldest *Beatmania* simulators written and

generally one of the most forked. The keys and the scratch pad would be mapped to keyboard keys. It originally only supported its own 5-key with scratch Be-Music Script ".bms" format, but newer versions were written to support the 7-key with scratch Be-Music Script Extended ".bme" format. The *BM98* simulator was the source of inspiration for many *Beatmania* and *Dance Dance Revolution* clones. Although there are few active *Beatmania* simulators in development today, *StepMania* supports the format.

#### **Frets on Fire** [8]

**Lead Developer:** Sami Kyostila

**Website:** <http://fretsonfire.sourceforge.net/>

**License:** GNU GPL

**Simulates:** Guitar Hero

**Significance:**

*Frets on Fire* is a *Guitar Hero* simulator originally written for the Assembly '06 demoparty, but continued on development as a full-fledged, cross-playform rhythm game simulator. Players of *Frets on Fire* would usually hold their keyboard like a guitar, with a row of keys and a special extra key representing the frets and pick. The game comes with its own built-in pattern editor and saves note data in its own custom format in plain-text ".ini" files. The game is no longer in development, ending at version 1.3.110, but still has an active player community.

#### **osu!** [5]

**Lead Developer:** Dean Herbert

**Website:** <http://osu.ppy.sh/>

**License:** Freeware

**Simulates:** Osu! Tatakae! Ouendan, Taiko no Tatsujin

**Significance:**

*osu!* is a simulator for *Osu! Tatakae! Ouenda* and *Taiko no Tatsujin*. The game uses a mouse in place of a touchscreen, but also supports tablet PCs as well as keyboard configurations. The game comes with its own built-in pattern editor and saves note data in its own ".osu" format. *osu!* is still actively in development, with *osu!stream* being the next project in line.

#### **Project -Project DXXX-** [6]

**Lead Developer:** KHCMaster

**Website:** <http://projectdxxx.blog9.fc2.com/>

**License:** Freeware

**Simulates:** Hatsune Miku: Project DIVA

**Significance:**

*Project -Project DXXX-* is the first and currently only simulator for *Hatsune Miku: Project DIVA*. The game uses the keyboard mapped to PSP equivalent buttons, but also adds in buttons for the arrow keys and shoulder buttons not in the original. The game comes with an external pattern editor and saves note data in its own ".ppd" format. The game is still in development, currently at version 0.5.2.0.

#### **Youbeat** [3]

**Lead Developer:** DAZ-Y3

**Website:** <http://youbeat.datael.co.uk/>

**License:** Freeware

**Simulates:** jubeat

**Significance:**

*Youbeat* is the most widely used simulator for *jubeat*. The

game uses the keyboard mapped to the 16 *jubeat* squares. The game reads note data in its own custom format in plain-text ".txt" files. The has stopped development due to lack of time by its creator and is currently at version GL38.

### 3. PROJECT PROPOSAL

In this project, a rhythm game simulator called *Beats2* will be developed. The game will try to meet the following three goals:

- **Open:** Code should be cross platform, open source, easy to port, and maintainable.
- **Multi-Mode:** Game engine supports multiple note data formats and gameplay modes.
- **Touch-Driven :** User interfaces are designed with the touch-driven input paradigm.

The final product will be released at the end of the project as an open beta at <http://beatsportable.com> with code and documentation hosted at <http://code.google.com/p/beats2/>

#### 3.1 Anticipated Approach

##### Beats, Advanced Rhythm Game

*Beats2* will be the spiritual successor of the rhythm game simulator, *Beats, Advanced Rhythm Game*, shortened as *Beats*. *Beats* was originally written for PennApps 2010, a 48-hour hackathon held at the University of Pennsylvania [9]. The game was released as closed source freeware for the Android platform and initially only supported simulation of *StepMania's* ".sm" format. Through continued development, the game expanded to also support *Dance With Intensity's* ".dwi" format as well as partial support for *osu!'s* gameplay and ".osu" format. The game was written for small, touch-screen Android phones only, but simple support for Android tablets was added later. [11]

Even though the game was incomplete and very buggy, development of the game ended at version 1.7b and the game was open sourced under a Modified BSD License. Due to the hack nature of the source code, however, *Beats* cannot be easily ported to other platforms and addition of multiple modes would require extensive modifications. [11]

*Beats2* will not be a rewrite of *Beats*; instead, it should be considered a new project that draws upon the lessons learned in the development of *Beats*. *Beats2* will not use any of the original *Beats* source code but it will take into consideration the vast amount of user feedback of *Beats* in its game design.

##### Open

*Beats2* will be developed to be as accessible to as large an audience as possible. The source code will be open source under a Modified BSD License and will be hosted on Google code at <http://code.google.com/p/beats2/>. The game will be written with the open source, cross-platform *libgdx* game development library, allowing it to run on Windows, Linux, Mac OSX, and Android [4]. Details for the touchscreen devices that will be targetted in this project are as follows:

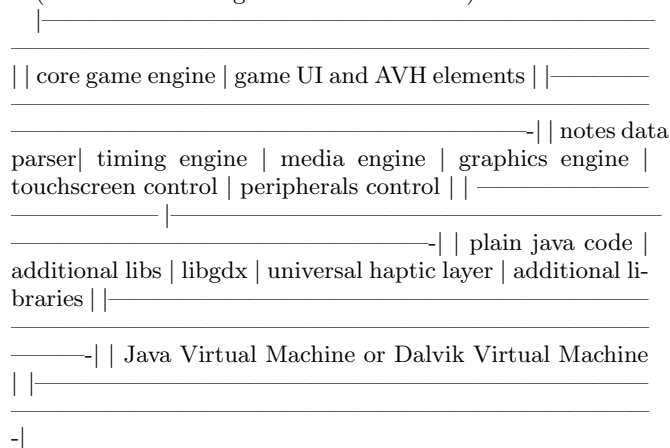
- **Samsung Galaxy S:** smartphone, 1GHz processor, 4-inch capacitive touchscreen, 16 multi-touch points, built-in vibrating motor, Android 2.3

- **Samsung Galaxy Tab 10.1:** tablet, 1GHz dual-core processor, 10.1-inch capacitive touchscreen, 10 multi-touch points, built-in vibrating motor, Android 3.0
- **Dell XPS L702X ,** laptop, 2.2GHz quad-core processor, 17-inch capacitive touchscreen, 4 multi-touch points, Windows 7 Professional

These devices are represent the three typical consumer touch-screen device categories: phones, tablets, and laptops. While they are relatively high-end products at the moment, their hardware specifications will most likely be standard within a few years as touchscreen devices becomes increasingly common. Linux and Mac OSX compatibility will not be tested for but will be targetted for as a stretch goal.

The source code itself will also be designed to be very modular and easy to maintain, port, and even expand on. With the exception of media playback, most platform-specific implementation requirements will be abstracted away through the use of *libgdx*. All game menus and gameplay modes will be written using a common graphics engine and touchscreen control framework for user interaction, but rely on a core game engine for the rhythm aspect of the game. The gameplay modes will be written independently of each other to allow for modularity and easy expandability of the game. The format-specific parsers will also be written independent of each other but save notes data in a common internal format. The general abstraction layers will be as follows:

(TODO: Make diagram based on below:)



The notes data parser and timing engine will be implemented in pure Java, making it completely platform-independent. The media engine will be a mix of what's supported by *libgdx* and additional, platform-specific media-playback libraries based on availability. The graphics engine will rely on *libgdx* (which wraps the cross-platform OpenGL ES library). Touchscreen control will rely on *libgdx* for multi-touch input and Immersion's Universal Haptic Layer for haptic feedback on Android devices. Any additional peripheral support (such as the XBOX Kinect, Guitar Hero guitars, or any other specialized input devices) will be stretch goals and implemented through additional libraries.

##### Multi-Mode

*Beats2's* goal of supporting simulation of multiple gameplay mode is to shift the focus toward the underlying timing and rhythm aspect of the gameplay rather than the visual representation. Notes data from various different rhythm games will be parsed by format-specific parsers but saved in

a shared internal data structure which group together similar fields. This common data structure will allow notes data for one particular rhythm game to be played under various other gameplay modes, thus freeing the player from any format-gameplay mode restrictions. Thus, the player has the ability to choose whatever gameplay mode they wish for the notes data to be played in. This allows rhythm game players to explore what gameplay mode they best prefer or is most suitable for challenging their sense of timing and rhythm.

The following are planned gameplay modes for *Beats2*, in priority order:

- **step**: 4 falling arrows, based on *Dance Dance Revolution*, *StepMania*'s ".sm" format and *Dance With Intensity*'s ".dwi" formats
- **pump**: 4 falling arrows and a middle block, based on *Pump It Up Pro*, use *Kick It Up*'s ".ksf" format
- **beat**: 7 keys + scratch, based on *Beatmania IIDX*, use *BM98*'s ".bms" and ".bme" formats
- **pop**: 9 keys, based on *Pop 'N Music*, use *Feeling Pomu*'s ".pms" format
- **guitar**: 5 tabs + pick, based on *Guitar*, use *Frets on Fire*'s ".ini" format
- **diva**: 10 buttons, based on *Miku Hatsune: Project DIVA*, use *Project -Project Dxxx-*'s ".ppd" format
- **box**: 16 boxes, based on *jubeat*, use *Youbeat*'s ".txt" format
- **taiko**: 2 buttons, based on *Taiko no Tatsujin*, use *osu!*'s ".osu" format
- **osu**: (x,y) taps, based on *Osu! Tatakae! Ouendan!*, use *osu!*'s ".osu" format
- **technika**: (x,y) taps, based on *DJMAX Technika*, create a new format (no prominent simulator exists at the moment)

## Touch-Driven

*Beats2* will be designed based on the touch-driven input paradigm. This means that traditional keyboard keystroke and mouse point-and-click navigation interaction models will not be used in either the game menus or the gameplay modes themselves. For example, scroll-based lists will be replaced with a swiping cover-flow-like lists, and touch-interactive elements will be oriented according to expected hand locations instead of relative to each other in set layouts.

Touchscreen interactions currently include elementary taps, holds, and slide actions, as well as more advanced gestures such as pinch, swipe/flick, scroll, and pivot/rotate. Determining which interaction to use in place of traditional keystrokes or clicks will depend on what feels most natural for the user to achieve a goal with the user interface. Hence, the game menu interfaces and gameplay interactions must be designed such that which touch action to use should be very obvious and intuitive to the user. In addition, the response of the UI itself should make sense to the user based on the action performed. The following are areas in which the touch-driven input paradigm will be used.

## **Home Menu:**

Because *Beats2* will be a multi-mode rhythm game simulator, what the user will see should vary depending on the gameplay mode. The game menu graphics should attune to the selected mode and the songs that appear will de-

pend on the notes data compatibility. Hence, the gameplay mode should be chosen at the start of the game, with the home page acting as a mode selector. A left-right swipe selector could be used, with corner-based icons can represent the other traditional "main menu" items, such as Settings, Scoreboard, Website, and Exit. A "Touch screen to start" message should be displayed, telling the user to tap the centre of the screen to select the game mode and enter the Song selection screen.

## **Songs Selector:**

Without any clear up/down arrow notion in touch interactions, traditional scrolling song lists are unintuitive for song selection. For large screens such as those on tablets, a grid-display of song image art could be used. For small screens such as those on phones, a cover-flow selector could be used. Tapping on the song image art would bring up an overlay displaying song details as well as icons for changing difficulty or other gameplay settings. Tapping the song image art in the overlay would then start the game.

## **Settings Page:**

Due to the purpose of displaying and accessing lots of information at once, it makes most sense for the settings page to be displayed as a traditional list in "setting -> state" format. The settings categories, however, can be displayed as a swipe-rotatable carousel at the top. The setting states can also be represented as tap-toggles or finger-adjustable sliders.

## **Results Screen:**

The results screen is usually just a screen displaying information in a graphical way. The change to a touch-driven paradigm does not affect the way non-interactive information should be displayed, so *Beats2* will probably use a results screen layout based on a mix of layouts from other rhythm games. The results screen, however, does usually include navigational buttons for actions such as retry, return, or score submission. These buttons should be large and easy to tap but also not placed in the way of the results screen information.

## **Gameplay Actions:**

A rhythm game gameplay mode is defined by the way the notes are displayed and the way the player interacts with those notes. Since *Beats2* is a rhythm game simulator, the way notes are displayed depends mostly on the game that is being simulated. There is acceptable and expected flexibility, however, in how the player interacts with the notes, allowing for deviated from the original game for input mechanisms. For example, in *Dance Dance Revolution*, pads are stepped on with the feet when arrows fall into a hitbox area. In *Beats2*, this pad stepping will be replaced with the finger tapping of the hitbox itself. Because of the usage of a common notes data structure, previously non-existent actions can be introduced to one gameplay mode from another. For example, holds can be added to the *osu* and *taiko* modes, and holds can be added to the *beats* and *pop* modes. Mode mixing experiments such as replacing tap-based timing in modes such as *step* with overlap-sliding-based timing can also be done.

## **Gameplay AVH Elements:**

When the player performs an action on a note, an immediate and perceivable reaction is expected. The most basic reaction in rhythm games is the disappearance of the note, indicating that the note has been hit or acted on and thus has been removed from the pool of active notes. The addition of more drastic reactions, however, can increase the immersive experience of the player. AVH elements that provide cues reflecting timing accuracy can drastically improve subsequent performance and the player's sense of rhythm. Audio elements such as audible clicks or other short sound effects alert whether or not the player hit the note or missed. Visual elements such as coloured explosions of the note and flashing combo counts both excite the player as well as give an indicator on current timing accuracy and overall performance. Haptic feedback such as note-specific haptic vibrations create the illusion of actually pressing distinct buttons instead of solid glass, as well as give the player an immediate indicator of whether or not their action was registered and recognized by the game.

### 3.2 Evaluation Criteria

Suppose you have implemented your approach, and it is functioning. Now how are you going to convince readers your approach is better than what exists? In the factorization example, you could just compare run-times between algorithms run on the same input. The image recognition example might use a percentage of accurate classifications. Other fields may have established testing benchmarks.

No matter the case, you need to prove you have contributed to the field. This will be easier for some than others. In particular, those with 'sensory' projects involving visual or sonic elements need to think this point through – objectivity measures are always better than subjective ones.

## 4. RESEARCH TIMELINE

Finally, we would like you to speculate about the pace of your research progress. This section need not be lengthy, we would just like you to specify some milestones so we can gauge your progress during our intermediate interviews. Let us follow through with our image recognition example:

- ALREADY COMPLETED: Preliminary reading. Began implementation of image-recognition algorithm.
- PRIOR-TO THANKSGIVING : Photograph buildings for DB. Make algorithm more efficient, tune parameters.
- PRIOR-TO CHRISTMAS : Create server-MMS interface. Expand tagged DB collection.
- COMPLETION TASKS : Verify implementation is bug-free. Conduct accuracy testing. Complete write-up.
- IF THERE'S TIME : Investigate image pre-processing techniques to improve accuracy.

## 5. REFERENCES

- [1] AJ 187. bemani and music game simulator file types. [http://www.ajworld.net/simfile\\_types.txt](http://www.ajworld.net/simfile_types.txt).
- [2] Chris Danford. Stepmania: Free dance game and songs. <http://www.stepmania.com/>.
- [3] DAZ-Y3. Youbeat by daz-y3. <http://youbeat.datael.co.uk/>.
- [4] Badlogic Games. libgdx. <http://code.google.com/p/libgdx/>.
- [5] Dean Herbert. osu! - rhythm is just a click away. <http://osu.ppy.sh/>.
- [6] KHCMASTER. Ppd means project -project dxxx-. <http://projectdxxx.blog9.fc2.com/>.
- [7] Konami. jubeat knit -super site-. <http://www.konami.jp/bemani/jubeat>.
- [8] Sami Kyostila. Frets on fire. <http://fretsonfire.sourceforge.net/>.
- [9] Justin Meltzer. Interview w/ beats creator from pennapps 2010. <http://pennapps.com/spring2011/2011/01/interview-w-beats-creator-pa10/>.
- [10] Dance Dance Revolution Fanatics of Kansas City. Ddrkc. <http://www.ddrkc.com>.
- [11] Philip Peng. Beats portable, advanced rhythm game for android. <http://beatsportable.com/>.
- [12] Philip Peng. beats2. <http://code.google.com/p/beats2/>.
- [13] Pentavision. Djmax technika - beyond the future. <http://djmax.co.kr/technika/>.
- [14] Sega. Hatsune miku -project diva- official site. <http://miku.sega.jp/>.
- [15] Scott Steinberg. Music games rock: Rhythm gaming's greatest hits of all time. <http://www.musicgamesrock.com/Music Games Rock.pdf>.