# CIS400/401 Project Proposal - The Effectiveness of Using Touch-Input Gestures in Rhythm Games

Dept. of CIS - Senior Design 2011-2012

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 touchscreen devices become increasingly popular, new software applications are expected to support touch-focused interfaces for user interaction. This project focuses on the evaluation of the effectiveness of various touch-input gestures in designing rhythm games. This will be accomplished through the development of a touch-input gesture recognition toolkit, the development of rhythm game prototypes using that toolkit, and finally a quantative evaluation of the effectiveness of the touch-input gestures in each prototype through an Android demo app.*

## 1. INTRODUCTION

Over the past few years, touchscreen devices have become increasingly common in the consumer market. According to a report in the March 2010 publication of *Information Display*, consumer-device manufacturers are rapidly adopting touch, with revenues increasing *10x* and unit production *3x* faster than the display industry [13]. With the adoption of this new input technology comes the natural expectation of increased software support for new touch-focused interfaces, allowing for more natural, intuitive, and powerful human-device interactions [3].

One area that touchscreen support can be leveraged is in the design of rhythm games. A *touchscreen* is a specialized display that receives user input through physical contact from a finger or stylus. *Rhythm games* are a genre of music-based games in which the player performs specific actions in response to audio and visual cues. Rhythm games often focus the the player's beat recognition abilities and consequently their timing accuracy, aided through visual patterns that match the rhythm of the song. These visual patterns consist of a series of *note* objects that appear or move across the screen in a manner referred to in this project as the *gameplay mode*. Interaction with these notes would typically involve hand actions occuring in the *hitbox* area, a pre-defined area for interaction. In non-touchscreen rhythm games, such actions may be the pressing of a button; in a touchscreen scenario, such actions would be either *soft button* touches, defined as virtual buttons interacted by through tap, or *touch-input gestures*, defined as touch events with predefined timing and path properties.

Experimental analysis of touchscreen gesture designs show that gestures can offer significantly increased reaction times and reduced attention load over soft buttons, especially under circumstances where the physical interaction is not the centre of the user's attention [1]. In rhythm games, lowering the attention load required for executing the associated note action can result in increased focus on the song's rhythm. Hence, it can be assumed that designing rhythm games to use touch gestures that users find most effective can lead to improved timing accuracy. This project address the question of: when designing rhythm games, which touch-input gestures are the most effective for which gameplay mode.

## 2. PROJECT PROPOSAL

This project will compare touch-input gestures for rhythm game design through the following three stages:

1. **Development** of a touch-input gesture recognition toolkit for rhythm games
2. **Prototyping** of various rhythm games created using the toolkit
3. **Evaluation** of the prototypes through collecting user feedback

These three stages will answer the question of which touch-input gestures are most effective in rhythm game design, allowing for future development of new rhythm games. All code and documentation for this project will be hosted at `http://code.google.com/p/beats2`.

### 2.1 Development

In the **Development** stage of the project, a touch-input gesture recognition toolkit will be developed for extending the *Unity 3* game development tools. As of the moment, the *Unity 3* scripts handles only simple raw touch events [12]. The proposed toolkit will extend the *Unity 3* scripts with the following goals:

- Track multiple touch events as either independent events or gestures
- Support for pre-defined single-touch gestures including taps, holds, slides, and swipes
- Support for pre-defined multi-touch gestures including press-and-tap, pinch, spread, rotate, and spin
- Provide an interface for associating gestures with *Unity 3* game objects

## 2.2 Prototyping

In the **Prototyping** stage of the project, the touch-input gesture recognition toolkit will be used to develop rhythm game prototypes. There will be prototypes featuring different gestures for each of the following generalized gameplay modes:

- Scrolling sheet of notes with a set hitbox region, *e.g. Dance Dance Revolution, Beatmania, Guitar Hero*
- Scrolling hitbox region with fixed note regions, *e.g. DJMAX Technika, Lumines*
- Fullscreen note region with fixed hitbox regions, *e.g. Guitaroo Man, jubeat, Hatsune Miku: Project DIVA*
- Fullscreen note region with fullscreen hitbox region, *e.g. Osu! Tatakae! Ouendan!*

## 2.3 Evaluation

In the **Evaluation** stage of the project, the rhythm game prototypes will be packaged together as a playable demo with a built-in user feedback system. This playable demo will be built for the Android 3.0 OS due to the facts that 1) Android 3.0 targets tablets, which usually feature large multi-touch displays [4], 2) Android tablet use is on the rise [14], and 3) *Unity 3* supports Android 2.1 and higher [10]. This playable demo will be tested by random samples of UPenn students as well as published online through the Android Market. The prototypes will be evaluated on the efficiency of the touch-input gestures with relation to the gameplay modes. The metrics used for measuring efficiency will be a 5-star rating scale on the following properties:

- Natural: Do the touch-input gestures feel intuitive for the gameplay mode?
- Reliable: Are the touch-input gestures easy to replicate and repeat?
- Reactive: Do the touch-input gestures allow for fast reaction times?
- Non-Distracting: Are the touch-input gestures non-distracting to gameplay?

The results of these comparisons may potentially lead to the future development of the more popular prototypes as full rhythm games.

# 3. TECHNICAL RESOURCES

**Test Device**
The target touchscreen device for testing in this project will be the Samsung Galaxy Tab 10.1. The Galaxy Tab is an Android 3.0 tablet running on a 1GHz dual-core processor and has a 10.1-inch capacitive touchscreen supporting up to 10 multi-touch points. It also has a built-in vibrating motor, allowing for additional haptic-feedback support by the toolkit.

**Unity 3**
`http://unity3d.com/unity/publishing/android.html`
The *Unity 3* development tools consists of the *editor*, the series of tools for developing games, and the *game engine*, the software backend that allows the developed games to run on target platforms. *Unity 3* was chosen due to its cross-platform support of other touchscreen-supporting platforms and large community and professional support base. In this project, the toolkit will be developed as script extentions for the editor, while the prototypes will be run using the game engine. The *Unity 3* license required for this project will be the regular *Unity 3* package with the additional Android add-on to allow for development of Android apps.

**uniTUIO**
`http://xtuio.com/index.php/projectsmain/utuiomain`
*uniTUIO* is a set of *Unity 3* scripts that add support for TUIO based multi-touch input. TUIO is an open framework that defines the common protocol and API for interacting with tangible multi-touch surfaces [5]. Although *Unity 3* uses a different protocol and has its own set of APIs for interacting with the Android touchscreen, the *uniTUIO* scripts may be a good starting point or reference material for developing the toolkit in this project.

**Android SDK**
`http://developer.android.com/sdk/`
The *Android SDK* is the set of development tools and core libraries required for developing Android applications. The *Unity 3* engine builds on top of the *Android SDK*.

**Eclipse**
`http://www.eclipse.org/`
*Eclipse* is the standard IDE (Integrated Development Environment) for developing Android apps. Both the *Android SDK* and the *Unity 3* Android add-on are designed to integrate with *Eclipse*.

**Google Code**
`http://code.google.com/p/beats2/`
*Google Code* is a group of online resources, tools, and hosting for project development. In this project, *Google Code* will be used for hosting the SVN source code, as well as bug tracking and wiki hosting.

**Google Analytics**
`http://www.google.com/analytics/`
*Google Analytics* is a free online service for tracking usage of features in applications. It will be used in this project to track the user feedback during the **Evaluation** stage.

# 4. WORK PLAN

The following is a rough timeline of the various milestones that need to be achieved for this project, along with their estimated completion dates. See the Appendix for Figure 1 for the Gantt chart for the fall semester timeline, Figure 2 for the Gantt chart for the spring semester timeline, and Figure 3 for the Gantt chart for the complete project timeline.

**1) Setup** - *Nov. 11th*
- Obtain Unity 3 license with Android add-on
- Set up Google Code project
- Set up Google Analytics
- Set up Eclipse with Android SDK and Unity 3
- Create a "Hello World" Android app through Unity 3

**2) Investigation** - *Nov. 25th*
- Investigate the uniTUIO framework
- Create a touch-input tracking Android app through Unity 3

- Experiment with tracking touch-input features and characteristics

**3) Drafting** - *Dec 9th*
- Draft specifications for touch gestures
- Draft specifications for toolkit components
- Create a simple prototype demoing one or two gesture recognitions

**4) Development** - *Feb 17th*
- Develop touch-input gesture recognition toolkit
- Create prototypes demoing each gesture recognitions

**5) Prototyping** - *Mar 16th*
- Develop simple rhythm game engine based on *Beats* [7]
- Create rhythm game prototypes for each gameplay mode using different touch-input gestures

**6) Evaluation** - *Apr 13th*
- Create an app showcasing the rhythm game prototype
- Collect user feedback on each prototype through random surveys and market publishing
- Evaluate the effectiveness of the different touch-input gestures through quantitive analysis of feedback

**7) Report** - *Apr 20th*
- Draw conclusions based on results of feedback analysis
- Write report summarizing findings

# 5. RELATED WORK

## 5.1 Touch-Input Rhythm Games

**Beats, Advanced Rhythm Game**

**From:** Philip Peng, 2010 [7]

**Platform:** Android
**Description:**
*Beats, Advanced Rhythm game* (*Beats*) is a *Dance Dance Revolution* (*DDR*) simulator for Android devices. *Beats* uses fixed hitbox regions that simulate the *DDR* dance mat, and a scrolling sheet of notes representing *DDR* arrows. *Beats* uses the tap and hold gestures.

**DJMAX Technika**

**From:** Pentavision, 2009 [8]
**Platform:** Arcade
**Description:**
*DJMAX Technika* is an arcade rhythm game on a machine with a large touchscreen display. Touch notes would appear on screen in fixed regions and a moving hitbar would scroll across the screen. The game uses taps, holds and slides gestures.

**jubeat**

**From:** Konami, 2008 [6]
**Platform:** Arcade
**Description:**
*jubeat* is an arcade rhythm game on a 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". All 16 touchscreen boxes were used, each with individual hitboxes that recognized the tap gesture.

**Osu! Tatakae! Ouendan!**

**From:** iNiS, 2005 [9]

**Platform:** Nintendo DS
**Description:**
*Osu! Tatakae! Ouendan!* (*Ouendan*), known as *Elite Beat Agents* in North America, is a rhythm game where the player must use the Nintendo DS stylus to interact with note objects that appear on the screen. *Ouendan* uses the full screen for note appearances and uses the tap, slide, and spin gestures.

**Taiko No Tatsujin**

**From:** Namco, 2001 [9]

**Platform:** Nintendo DS, iOS
**Description:**
The *Taiku No Tatsujin* series focuses on simulating the Japanes Taiko drumming experience. The game has fixed hitbox regions representing two large Taiko drums and uses a scrolling sheet of notes that move toward the hitbox region. It only uses the tap gesture.

## 5.2 Touch-Input Toolkits

**MT4j**

`http://www.mt4j.org`
*MT4j* (*Multitouch for Java*) is an open source Java framework for designing multi-touch applications. It is a complete framework rather than a toolkit or library, but it can be used as reference. It includes a flexible, customizable multi-touch gesture system that may be adapted in this project. In addition, it includes support for multi-touch in Windows 7, which *Unity 3* currently does not natively support and may be something this project's toolkit may add.

**GestureToolkit**

`http://gesturetoolkit.codeplex.com`
*GestureToolkit* is a toolkit for developing and testing multi-touch applications running on Windows Presentation Foundation or Silverlight. It includes predefined gestures as well as a language for defining new gestures in multi-step, multi-user, or multi-touch scenarios. The toolkit also includes a test framework for validating touch interactions and gesture definitions. The gestures expected to be used with this toolkit are far more complex than those expected to be used in rhythm games, but a test framework may be something that is worth developing alongside this project.

**PyMT**

`http://pymt.eu/`
*PyMT* is an open source pure Python library for developing cross-platform multi-touch applications. It does not, however, support Android or iOS as neither natively support Python. *Unity 3* does, however, support Boo [11], a dialect of Python, so some code snippets may modified for use in this project.

## 5.3 Touch-Input Research

**NUI Group Community Book**

`http://nuicode.com/projects/wiki-book/`
The *NUI Group* is a global research group focusing on the *natural user interface* concept, under which touch-input interfaces is included. The *Community Book*, titled *Multitouch Technologies*, is an online e-book summarizing the cummulative research of the *NUI Group* community. While it focuses

mainly on vision recognition of objects and interpretting hand gestures, Chapters 2.2 and 2.3 focus on multi-touch tracking and gesture recognition. The book describes the use of the Kalman filter and a k-Nearest Neighbour (k-NN) approach for tracking movements and connecting previous states for gestures, as well as provides guidelines for the gesture recognition techniques. [2]. This book will be used as a reputable design reference guide throughout this project.

**Experimental Analysis of Touch-Screen Gesture Designs in Mobile Environments**

http://yangl.org/pdf/gesturestudy-chi2011.pdf
In their research paper "Experimental Analysis of Touch-Screen Gesture Designs in Mobile Environments", Andrew Bragdon et al. study the impact of environmental distractions on touchscreen interaction. In particular, the paper focuses on different design factors for touch-input gestures compared to soft buttons. The results of their analysis was that gestures can offer significant performance gains and reduced attention load compared to soft buttons, with bezel-initiated gestures offering faster performance and mark-based gestures offering better accuracy [1]. The results of Andrew Bragdon et al.'s paper can be used for selecting and defining the gestures being tested, while some of the evaluation techniques used in the paper can be used in the Evaluation stage of this project.

# 6. EVALUATION CRITERIA

The project will be evaluated based on 1) the usability of the toolkit and its ease of integration with *Unity 3*, 2) the interest level of users trying out the rhythm game prototypes, and 3) the conclusiveness of the gesture comparison results. For 1), the final touch-input gesture recognition toolkit should allow for easy addition of gestures, easy integration with rhythm game engines, and provide an interface similar to those of *Unity 3*. For 2), users should find the rhythm game prototypes distinct, clearly demonstrative of the touch-input gestures that they compare, and fun to try out. For 3), the results should ideally conclude with certain touch-input gestures being predominantly more effective with particular gameplay modes (as opposed to the tap gesture being the most effective for all gameplay modes).

# 7. APPENDIX

*See figures on following pages.*

# 8. REFERENCES


[1] Andrew Bragdon, Eugene Nelson, Yang Li, and Ken Hinckley. Experimental analysis of touch-screen gesture designs in mobile environments. http://yangl.org/pdf/gesturestudy-chi2011.pdf.

[2] NUI Group. Multi-touch technologies v1.01. http://nuicode.com/attachments/download/115/.

[3] Bradley H. Hayes. Software driven multi-touch input display as an improved, intuitive, and practical interaction device. http://www.bradhayes.info/thesis.pdf.

[4] Google Inc. Android 3.0 platform highlights. http://developer.android.com/sdk/android-3.0-highlights.html.

[5] Martin Kaltenbrunner. Tuio.org. http://www.tuio.org.

[6] Konami. jubeat knit -super site-. http://www.konami.jp/bemani/jubeat.

[7] Philip Peng. Beats, advanced rhythm game. http"//beatsportable.com.

[8] Pentavision. Djmax technika - beyond the future. http://djmax.co.kr/technika/.

[9] Scott Steinberg. Music games rock: Rhythm gaming's greatest hits of all time. http://www.musicgamesrock.com/Music Games Rock.pdf.

[10] Unity Technologies. Unity - devices that we have tested on. http://unity3d.com/support/documentation/Manual/Android-Devices That Have Been Tested.html.

[11] Unity Technologies. Unity: Features - scripting. http://unity3d.com/unity/features/scripting.

[12] Unity Technologies. Unity script reference - input. http://unity3d.com/support/documentation/ScriptReferenc

[13] Geoff Walker. The best of times. http://www.informationdisplay.org/issues/2010/03/art3/a March 2010.

[14] Jun Yang. Android tablets gained on ipad in third quarter. http://www.bloomberg.com/news/2011-10-21/android-tablets-gained-on-ipad-in-third-quarter-researcher-says.html.

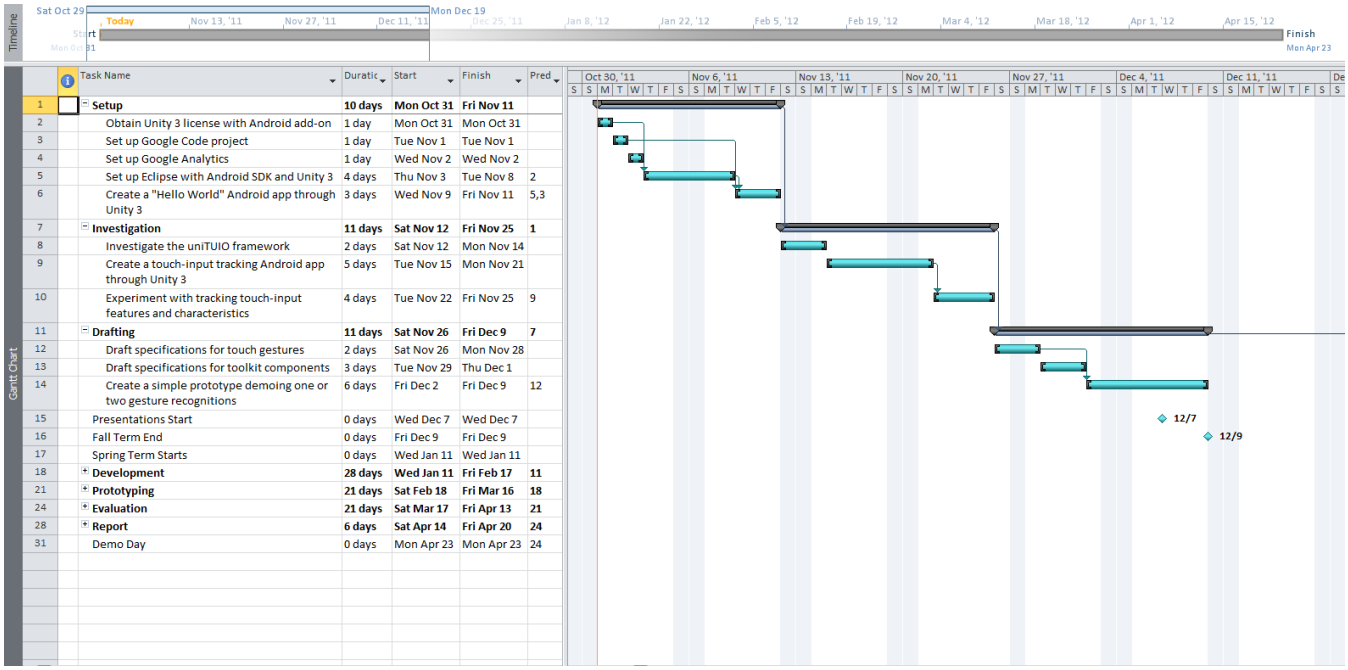


**Figure 1: Fall semester timeline**

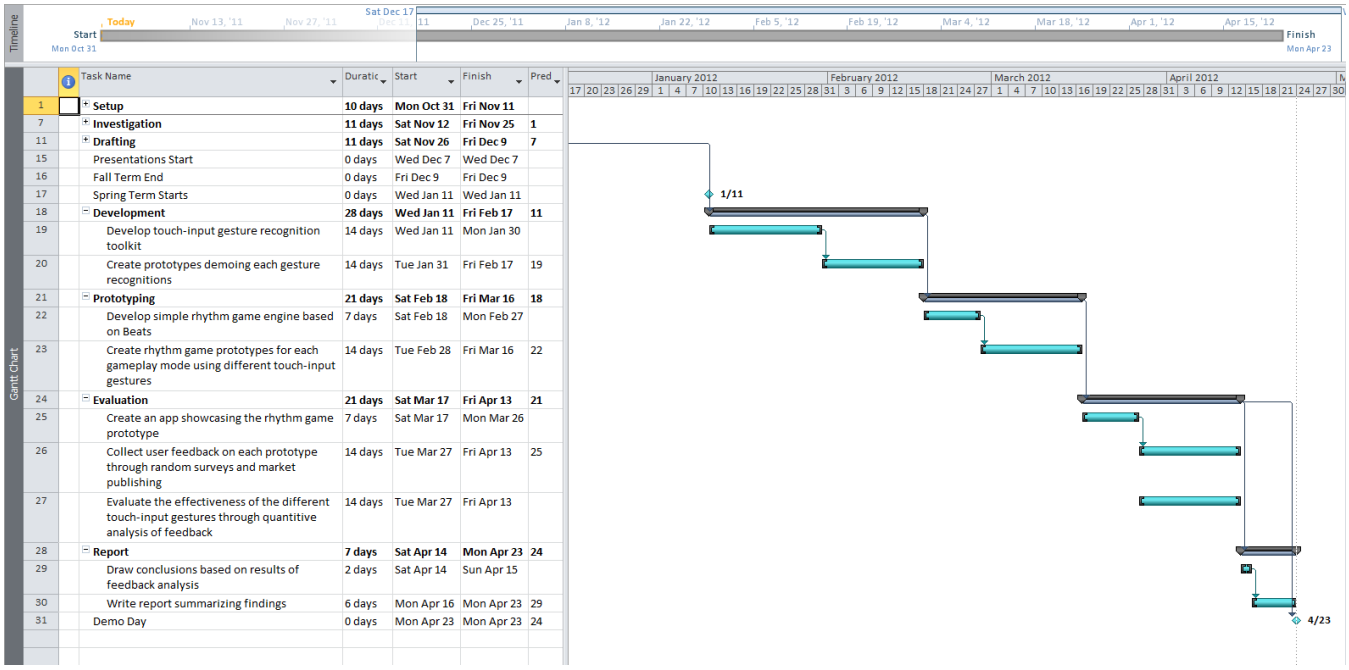


**Figure 2: Spring semester timeline**

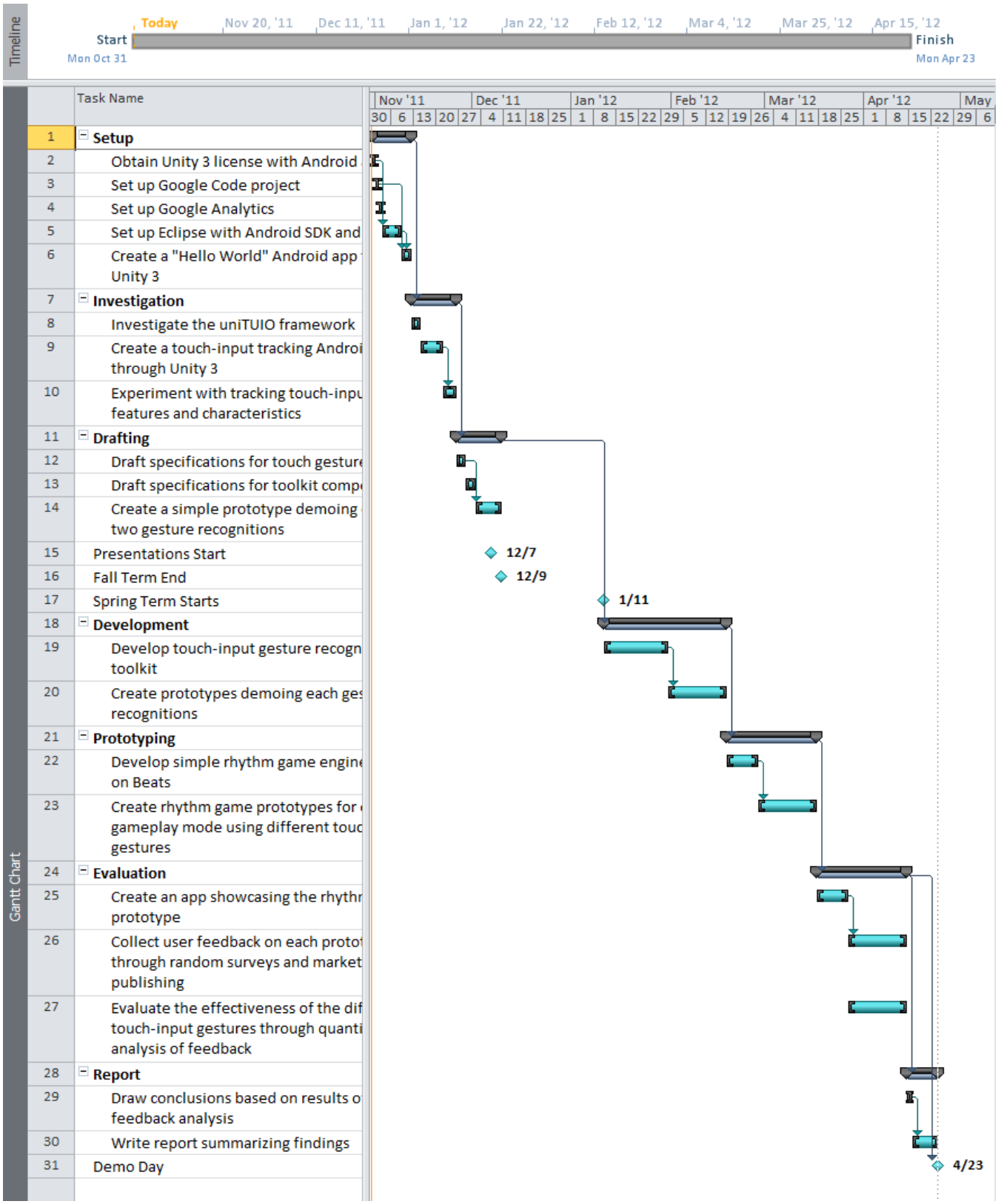


**Figure 3: Complete project timeline**