

Rhythm Game Development Toolkit For Touchscreen Devices



Members: Philip H. Peng

Advisor: Dr. Stephen H. Lane

CIS 400, Fall 2011, University of Pennsylvania

All images used in this belong to their rightful owners

Problems/Motivation

Touchscreen Devices

- New technology, becoming common
- Touch-driven input paradigm
- Games need to be redesigned

Rhythm Games

- Many kinds, common backend
- Each game designed for specific game mode
- Specialized hardware
- Inaccessible, uncommon
- Few rhythm games for touchscreens (-Beats)

Related Work

Relevant Games

Parappa The Rapper, Beatmania, Pop 'N Music, Dance Dance Revolution, Taiko No Tatsujin, Gitaroo Man, Osu! Tatakae! Ouendan!, Guitar Hero, jubeat, DJMAX Technika, Hatsune Miku: Project DIVA



Simulators

StepMania, BM98, Frets on Fire, osu!, Project – Project DXXX–, Youbeat



Beats

- <http://beatsportable.com>
- Released 1yr ago, for Android phone + tablets
- 350k+ downloads so far, 4.6/5.0 avg rating

Approach

Three targets:

- Usability
Cross-platform, open source, easy to port, modular and easy to extend
- Multi-Mode
Components allow for support of multiple gameplay modes and data formats
- Touch-Driven
Interface and input components designed for touch-driven input paradigm

Approach: Usability

Features

- <http://code.google.com/p/beats2>
- Open source, Modified BSD License
- Unity 3: cross-platform, large dev community
- Toolkit vs Framework
- Modular, independent components
- Three test targets:
 - Samsung Galaxy S: smartphone, Android
 - Samsung Galaxy Tab 10.1: tablet, Android
 - Dell XPS L702X: touchscreen laptop, Win 7/8

Approach: Multi-Mode

Parser Formats

- step/DDR: StepMania .sm, DWI .dwi
- pump/Pump It Up Pro: KIU .ksf
- beat/Beatmania: BM98 .bms/.bme
- pop/Pop 'N Music: Feeling Pomu .pms
- guitar/Guitar Hero: Frets on Fire .ini
- diva/Project DIVA: PPD .ppd
- box/jubeat: Youbeat .txt
- taiko/Taiko no Tatsujin: osu! .osu
- osu/Osu! Tatakae! Ouendan!: osu! .osu
- technika/DJMAX Technika, N/A

Approach: Multi-Mode

Pattern Modes

- Fade/appearing notes
- Scrolling notes sheet
- Pathed-motion notes
- Sliding hitbar

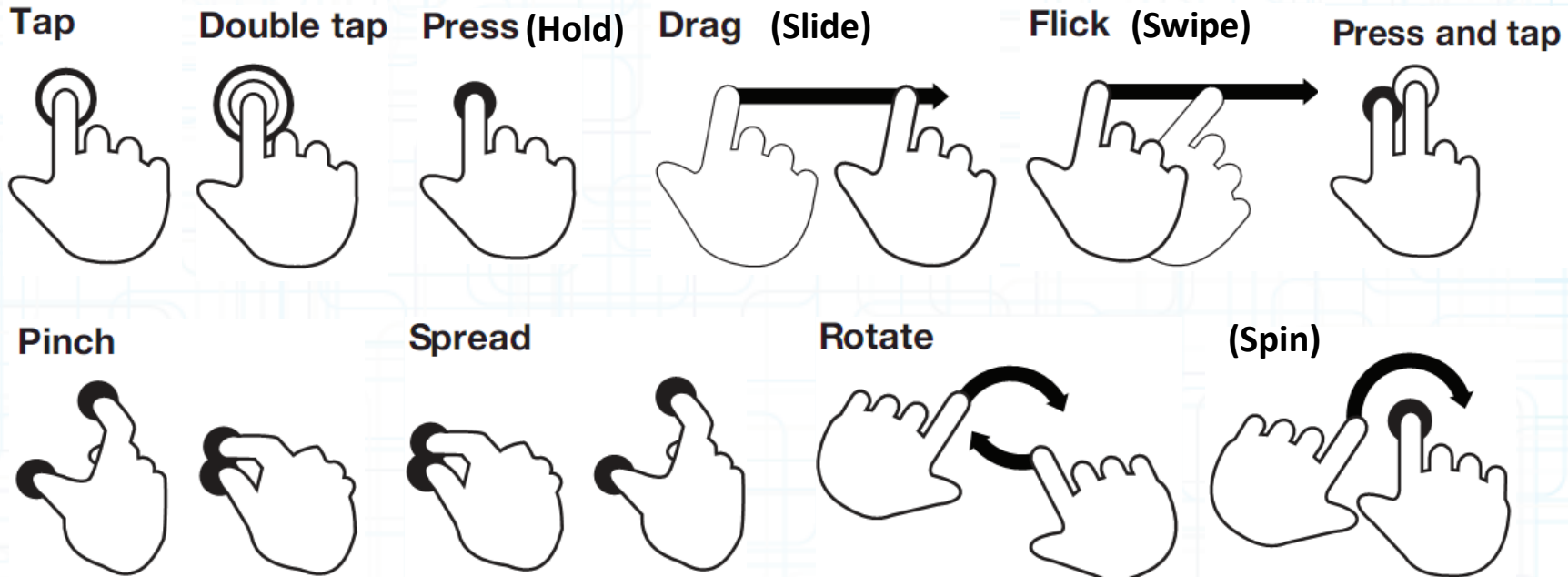
Backend

- Configurable timing/accuracy tracker
- Multiple scoring systems (DDRMAX2, osu, etc)
- Common notes data format
- Converters between data formats

Approach: Touch-Driven

Input Gestures

- From “Touch Gesture Reference Guide”
<http://www.lukew.com/touch/TouchGestureGuide.pdf>



Approach: Touch-Driven

User Interface

- NUI (coverflow, grids, large icons, etc.)
- All gesture-input (not just taps)
- Game modes support touch gestures
- AVH elements (sfx/gfx, haptic feedback)



DJMAX
Technika
menus are
a good
example

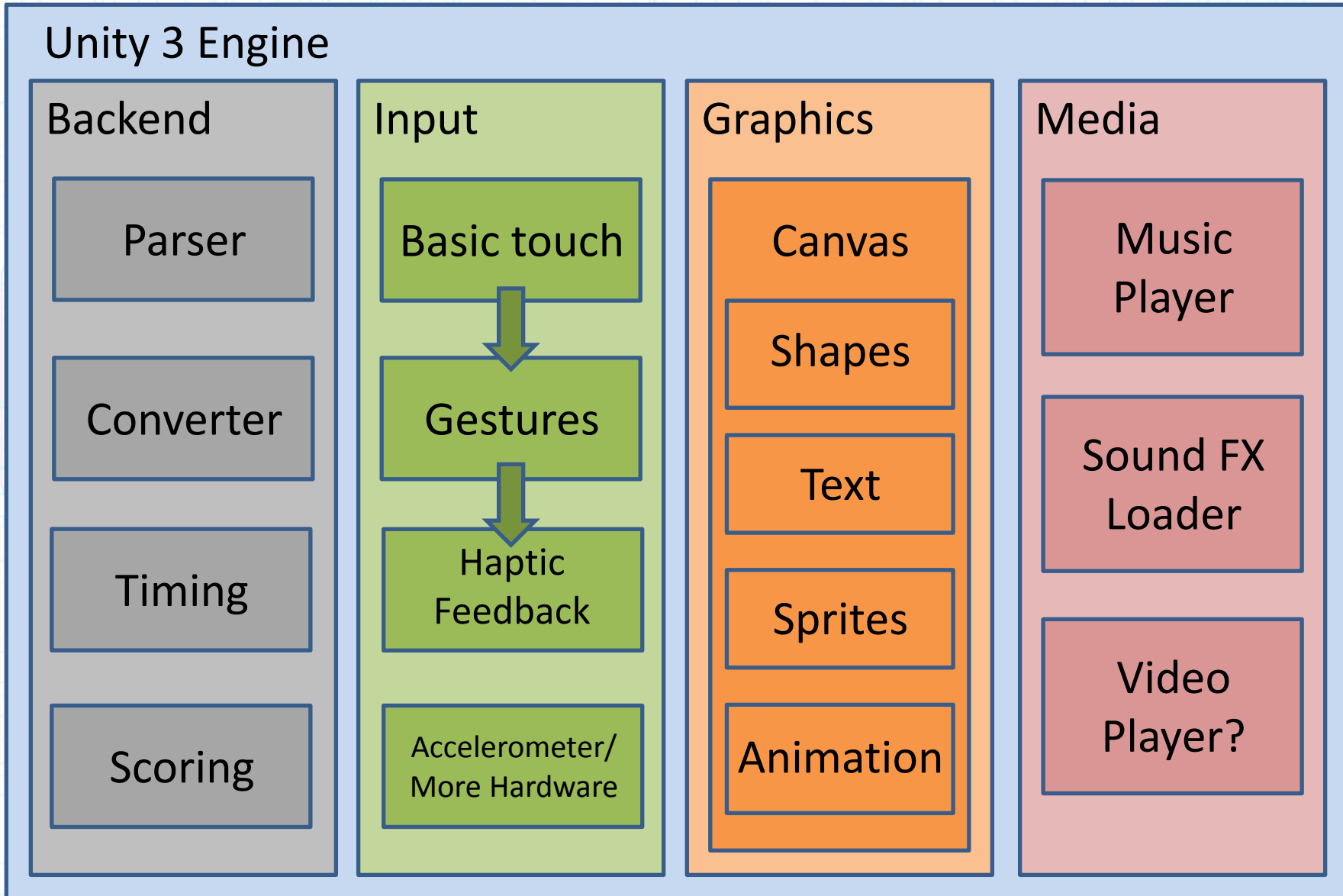
Approach: Technical

Tools/Resources

- Unity 3 Engine
- Android SDK
- Immersion's UHL
- Windows 8 Dev Preview tools
- Additional A/V playback codecs
- Google Code (svn + wiki + tracker)
- Crowdin (crowd-sourced translations)
- Google Analytics (via Unity 3)



Approach: Components



Approach: Demo – Beats2

Prototype

- Multi-mode rhythm game simulator
- Support step, beat, box, technika
- Supports tap, hold, slide, swipe
- Runs on Windows 7 and Android
- Beta published on Android Market

Stretch Goals

- Supports all modes
- Supports all gestures
- Runs on Windows 8
- XBOX Kinect integration
- Community-contributed content and involvement



Timeline

M0 – 3 weeks (Nov 18th)

- Investigate Unity 3 Engine, obtain license
- Set up Google Code/resources

M1 – 3 weeks (Dec 9th)

- Draft toolkit specifications
- Draft Beats2 game design
- Component prototypes

M2 – 5 weeks (Feb 3rd)

- Complete basic Backend components
- Complete basic Input components

M3 – 6 weeks (Mar 16th)

- Complete basic Media components
- Complete basic Graphics components

M4 – 5 weeks (Apr 20th)

- Working prototype of Beats2

M5 - Remainder

- Documentation
- Improve/debugging Beats2
- Stretch Goals