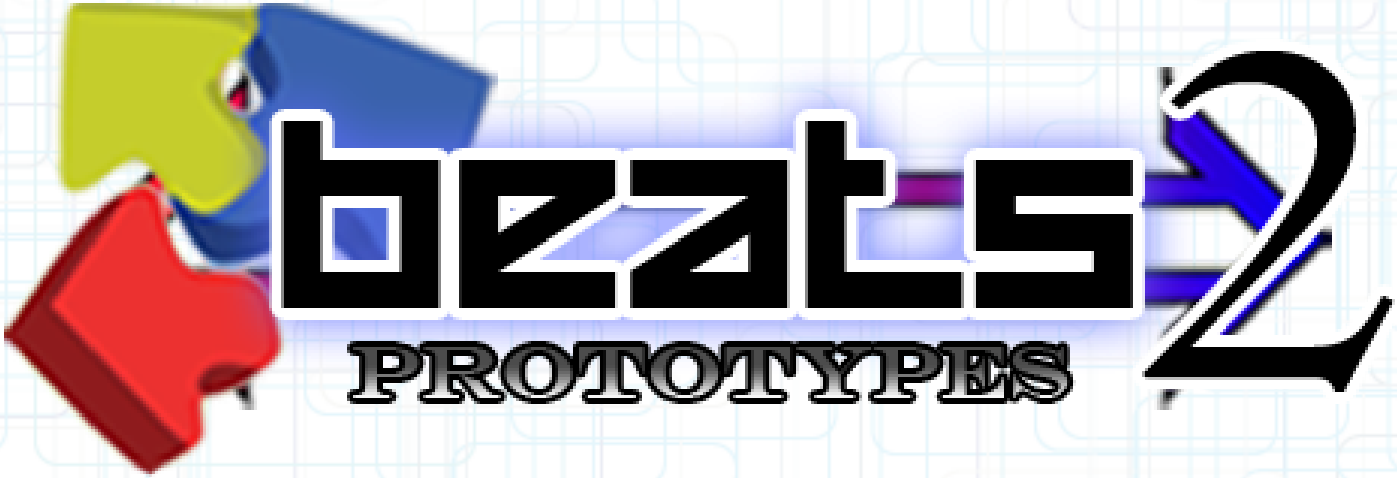
Designing Rhythm Games for Touchscreen Devices



Project Progress Report

Members: Philip H. Peng
Advisor: Dr. Stephen H. Lane
CIS 401, Fall 2011, University of Pennsylvania

Penn Engineering

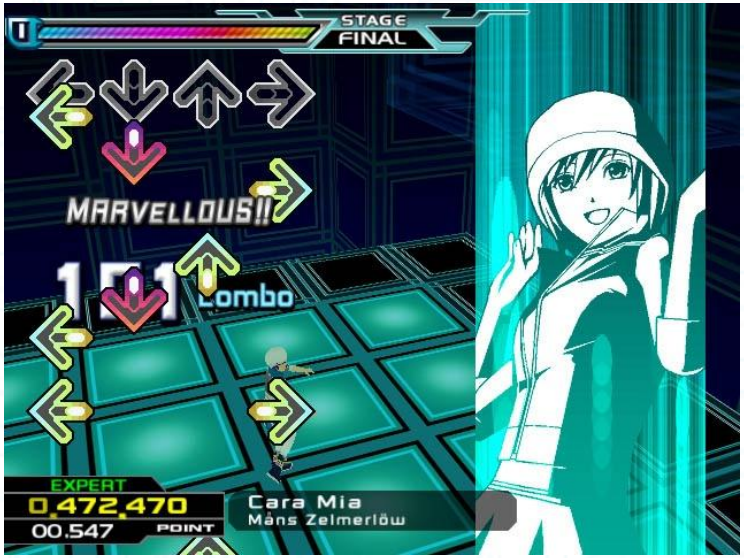
Presentation Overview

1) Summary
2) Project Proposal
3) Related Work
4) Project Outline
5) Progress
6) Demo
7) Results

Penn Engineering

Summary

- Rhythm game: time critical, response-based
- Touchscreen: new input method
- Rhythm game for touchscreen: how to design the interface for highly reactive gameplay?



Penn Engineering

Project Proposal

Goal:

Design, prototype, and evaluate different rhythm games interfaces for touchscreen devices.

Approach:

Create a rhythm game prototype for Android tablets that demos various game interfaces and collects usage data to evaluate their effectiveness.

Related Work

Wiimote + Dance Game

"Understanding Visual Interfaces for the Next Generation of Dance-Based Rhythm Video Games" – University of Central Florida, Orlando, FL

External Multi-touch Panel + Turn-Based Strategy Game

"A Study on Multi-Touch Interface for Game" – Chung-Ang University, Seoul, Korea

Overlayed Multi-touch Screen + Real-Time Strategy Game

"One-handed Interface for Multi-Touch Enabled Real-Time Strategy Games" – University of California, Santa Cruz, CA

Penn Engineering

Project Outline

1) Design – Draft

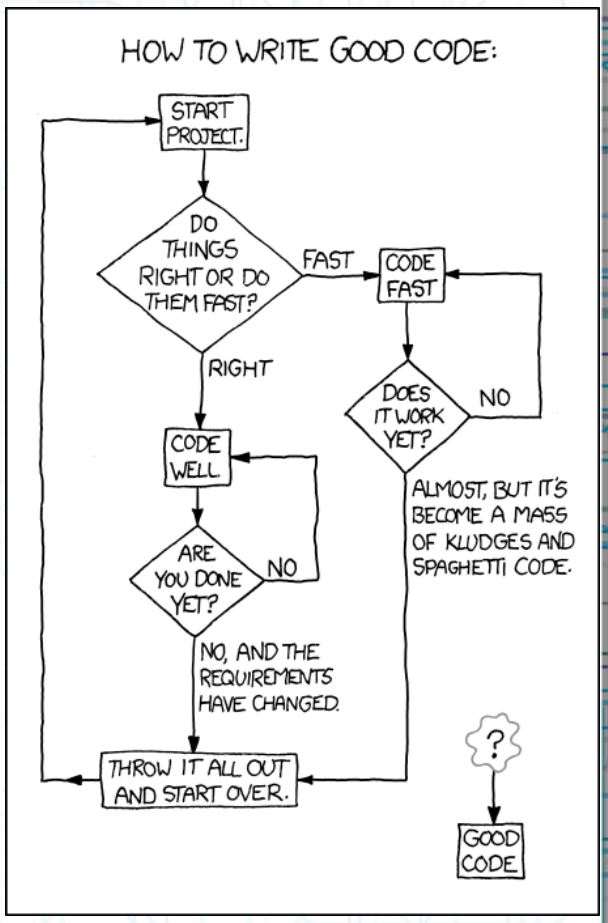
- Investigate existing interface designs
- Draft designs and evaluation metrics

2) Prototype – Code

- Implement these designs via Android
- Use common backbone to reduce non-relevant factors

3) Evaluation - Data

- Release to public with data analytics plugin
- Analyze collected data

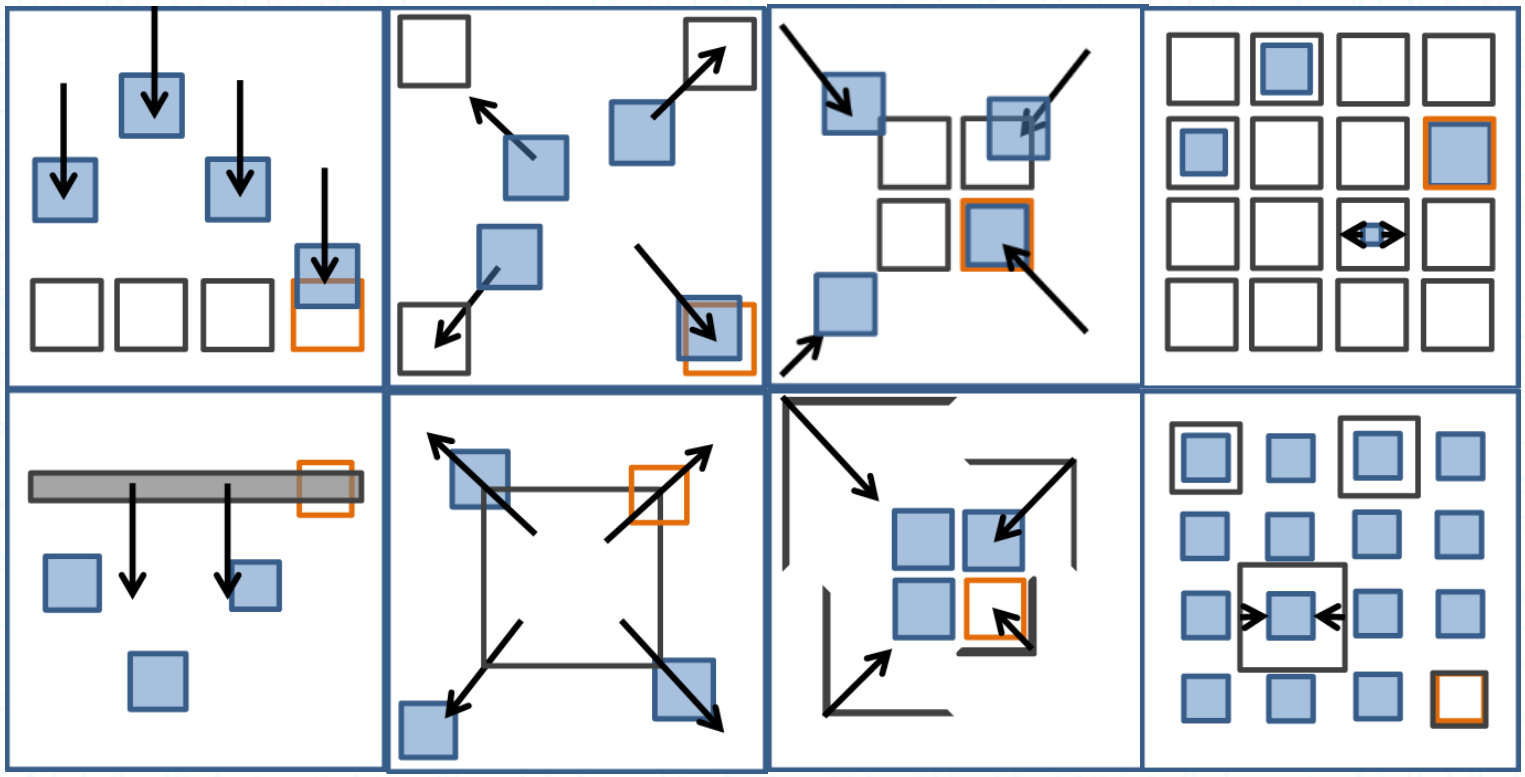


Penn Engineering

Progress

Design – Designs

- Interface designs finalized:



Progress

Design – Metrics

- Test song: smooooch (Beatmania IIDX)
  - High note frequency and 177 BPM
  - Strong audible baseline (good for rhythm)
  - Auto-generated stepfile (Dancing Monkeys)
- Metrics per design:
  - Total accuracy percent (data)
  - Missed note count (data)
  - Ranked enjoyability relative to other designs (feedback)
- Full timing chart will also be collected for overall trend analysis (if there are any)

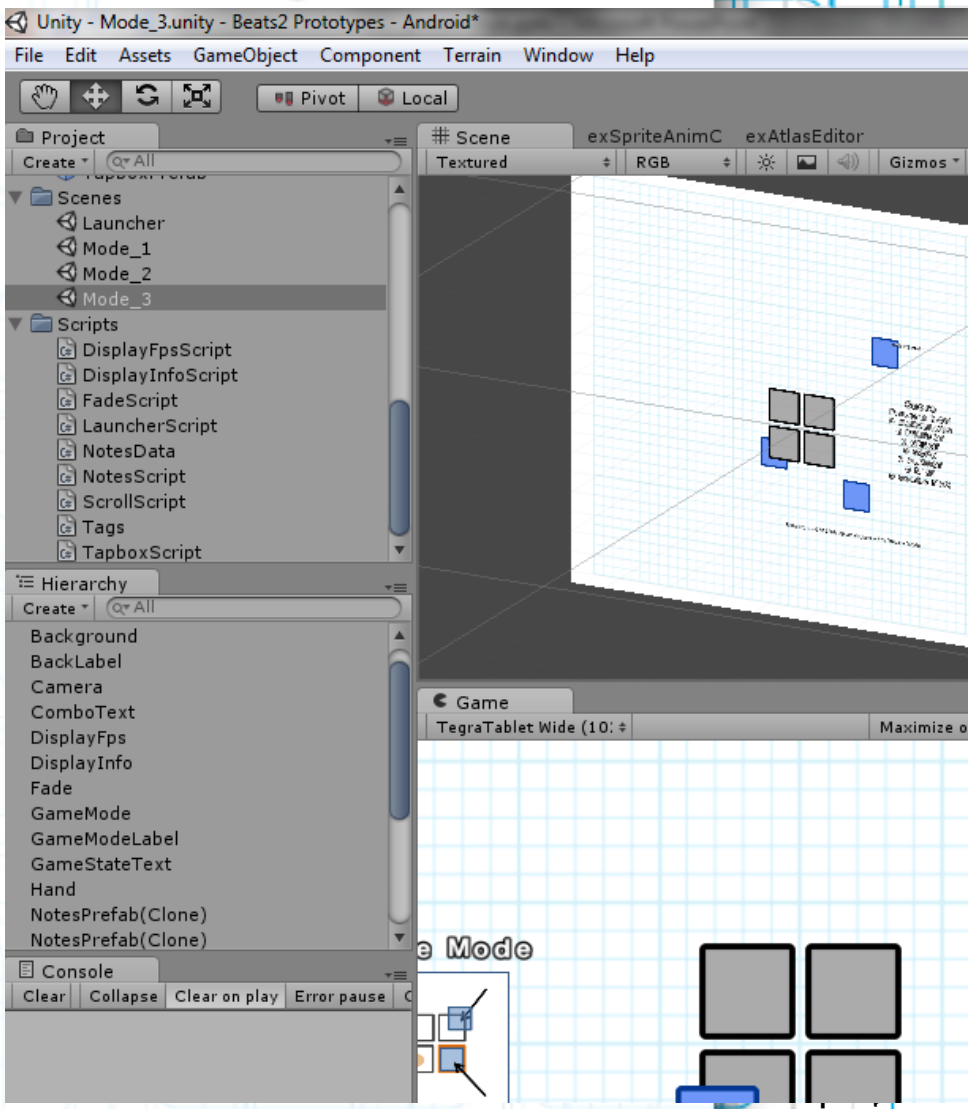
Penn Engineering

Progress

Prototype – Unity

- Biggest challenge of project
  - Game designer vs. Programmer
  - Unity = GameObject driven workflow
  - New to C#

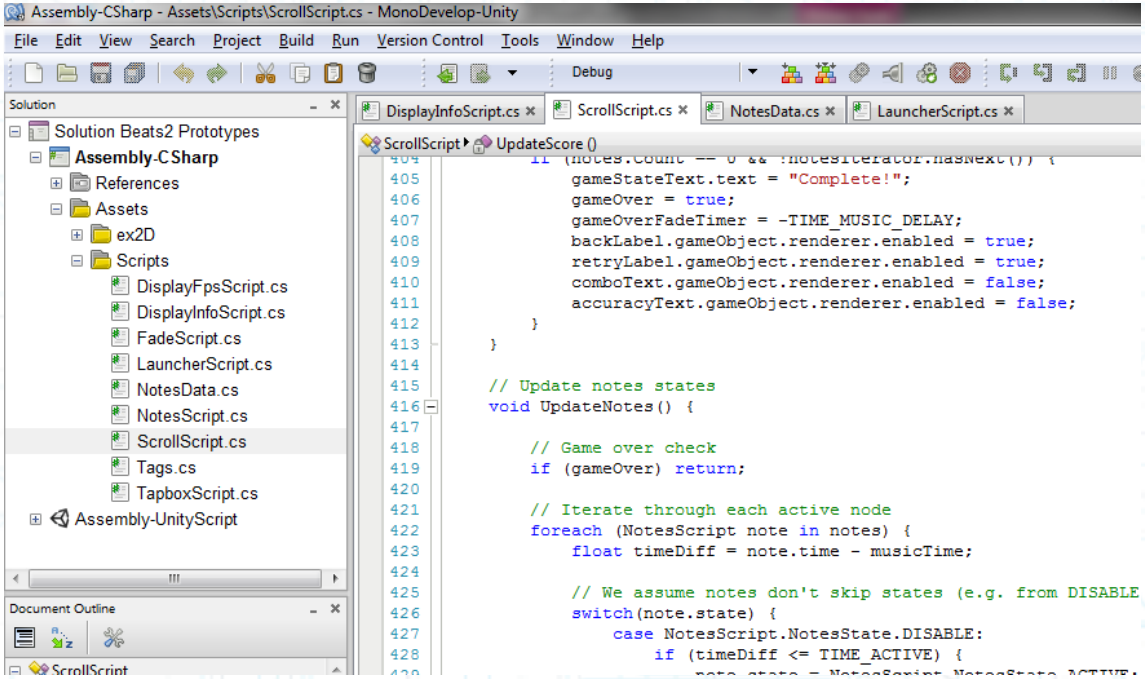
- Game Engine Experimentation
  - Android + iOS license
  - Othello2D vs ex2D
  - TouchGestures overcomplicated
  - MonoDevelop minimalistic



Progress

Prototype – Coding

- Flexible game engine
  - Timer, audio-synchronized
  - Dynamic object generator, memory efficient
  - Score tracking, based on timing accuracy
  - Runs on Android tablet and Windows
  - Adding new modes (interfaces) easy

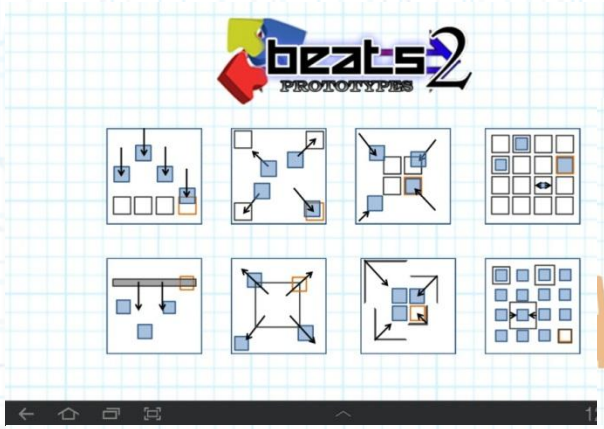


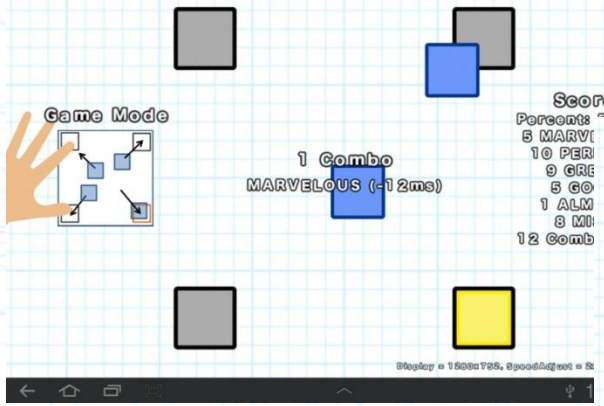
Progress

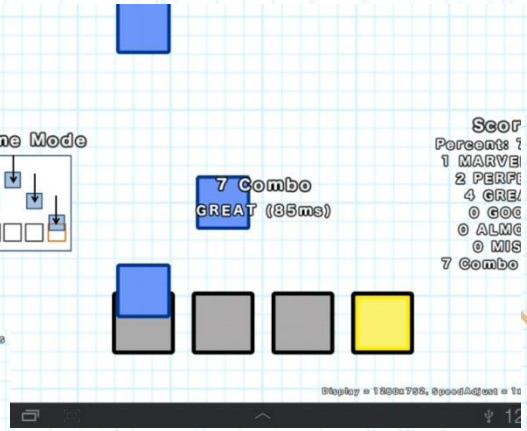
Evaluation – Setup

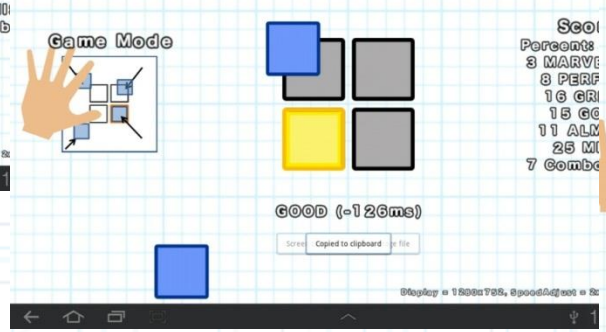
- Tweaking stage
  - Change object placements based on feedback
  - Tweak timing parameters
  - Improve graphics?
- Google Analytics vs own server
  - Analytics: generate graphs but limited info collection
  - Server: custom information but have to set up
- Mass release on Android Market
  - Use Beats' update notifier to advertise (100k+ active users)

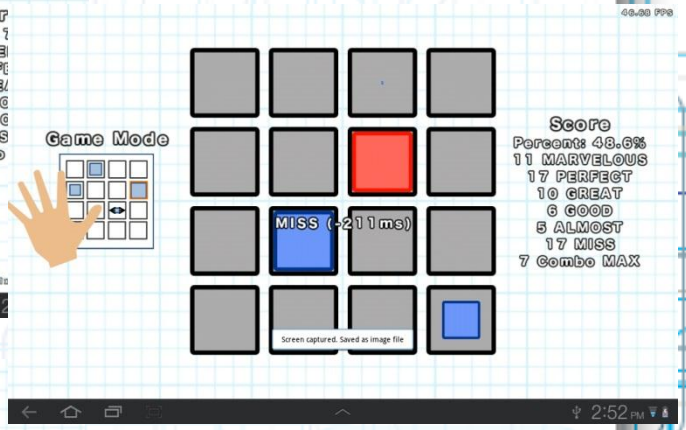
Demo

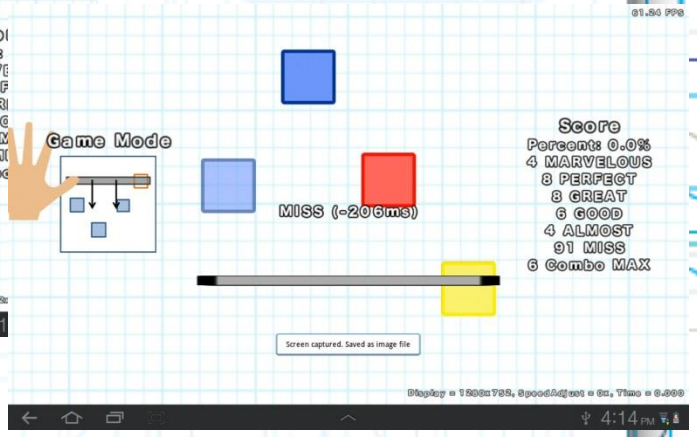












Penn Engineering

Results

Programming:

- Unity hard to learn but very flexible and worthwhile - write once, deploy everywhere
- ex2D very useful but very buggy and crashes
- Framerate very good (consistent 60fps) but timing window a bit too big (~15ns update rate)
- Code currently in one main script – need to learn more about Unity coding conventions (static global object?)

Results

Informal surveys:

- Testers used to DDR style scrolling
- Speed needs to be tweaked (some tester just had slow visual processing times)
- Tapbox placement definitely a factor (proximity to each other, spread of focus)
- Enjoyment factor independent from performance
- May try two songs to eliminate song familiarity from evaluation

Questions?


MercurialMadnessMan 767 points 13 days ago [-]
Sorry, this isn't really a question