Dubit's Coding Challenge

Specification Documents

Chuan Liu
ChuanLiu87@gmail.com
RolloLiu.com

Brief

Coding Environment

- OS: Windows 7
- Tools: Adobe Flash Builder 4.6 + Adobe Photoshop CS5 + IE9
- Programming Languages: AS3 + Flex 4
- Version Control: GitHub

Coding Style

I read the code and the code is obviously easy to read, easy to understand. The existing code comes with MVC. So I just keep and follow the original coding style and going to expand more features.

Basically, my code based on the following principles:

- Readability, logics with comments;
- Do only necessary changes for existing functions and variables;
- Similar design/logics for classes, functions, modules;

Tasks Status

Add Game Over / Score Screen (Accomplished)

After the game timer completed, the mole holes will be disappeared, and "Time's Up" label will shows up. Then it goes fade out, and "Score" shows up.

There is also a "Play Again!" button. It will bring mole holes back and restart the game. All game state except awarded achievement s will be reset.

Add Difficulty Levels (Accomplished)

3 different difficulty levels has been set-up: Easy, Medium and Hard. The difference between levels described in the table shown below:

Difficulty	Time Interval (ms)	Show Time Delay (ms)	Total Moles	Bonus Points
Easy	450 ~ 550	0 ~ 500	50	0
Medium	350 ~ 450	-200 ~ 200	60	10
Hard	250 ~ 350	-400 ~ -100	70	20

- Time Interval: how long to wait for the next mole shows up
- Show Time Delay: offset the time of how long does the mole stay at a hole
- Total Moles: how many moles going to show up
- Bonus Points: bonus points per successful kill

Add New Types of Mole (Accomplished)

The “Fire Mole” and the “Zombie Mole” have been successfully added into this game. Either Fire Mole or Zombie Mole is derived from original Mole class.

In addition, there have something new added into the original Mole class to help this achieve:

1. Added a new attribute “Type”:
    a. The renderer will know which texture use on it;
    b. Holes will know players killed which type of the mole.
2. Added a new attribute “Hit Points” for different types of mole. The player requires reaching enough hits to kill a mole.

Build in Achievements / Awards (Accomplished)

The application will create an achievement system and pass the reference to the Mole Game. The Mole Game will just hold a reference of it.

When the player killed or missed a mole, the Mole Hole will report this to its owner – the Mole Game. The Mole Game will then process the result and report it to the achievement system. There is no data has been stored so far.

Everything just happed inside the achievement system: it will count how many moles (and what type) has been killed. If the record matches any of the achievement’s criteria, it will dispatch an Achievement Event.

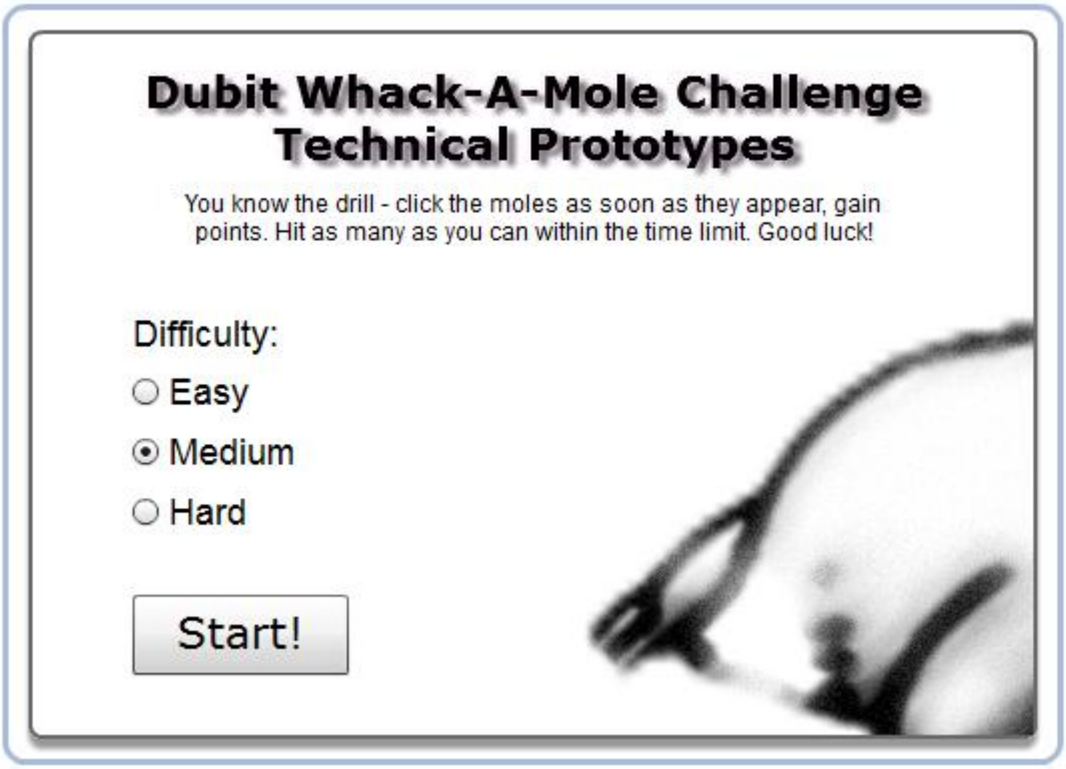
The Achievement Event will then handle by the Mole Game Viewer (controller). The controller will light on the certain achievement label -- the one which has been sent by the achievement event.

Extending and Embellishing

Something New

Re-designed Introduction View

The Introduction view now has a brand new layout and background (see Figure 1).



Dubit Whack-A-Mole Challenge
Technical Prototypes

You know the drill - click the moles as soon as they appear, gain points. Hit as many as you can within the time limit. Good luck!

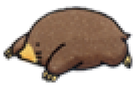
Difficulty:

○ Easy
● Medium
○ Hard

Figure 1: New Backgrounds for Introduction View

Re-designed Moles

There are new images for each type of moles:

Type	Idle	Hit Once	Hit Twice	Dead
Normal		N/A	N/A	
Fire			N/A	
Zombie				

In addition, there are some new methods for the Mole Hole Item Renderer. Now the renderer keeps update each frame, checks the status of its mole holes, and shows the correct image for it.

Also two new images "moleHit" and "moleSkull" added into the game, as a feedback for players: one for successful hit and one for successful kill.

Achievement "Perfect"

A new achievement "Perfect" has been added. It requires the player kill every single mole in a game.

Button "Continue"

A new button "Continue!" has been added into the score screen. It allows the player back to the introduction screen and re-selects a difficulty level. However, the achievements will be reset as well.

Sound Effects

Several sound effects have been added into the game:

- "Hit" sound effect for a successful hit by the player;
- 4 different "mole dead" sound effects will randomly picked and play for a successful kill.

Something Polished

- The Introduction view and the Mole Game view now has a number of animations for transition between scenes;
- Tooltips for "Easy", "Normal", "Hard" radio buttons;
- Tooltips for every achievement in the game;
- Award achievements' animations.

Reflection

Actually, this project is very interesting. I have a very, very similar project implemented for iOS when I was in the university (there is the link of the project in my website: http://rolloliu.com/?project=imole).

There was some kind of issues. I just listed some of them which I think is much more important.

The biggest problem I believe is the design for responds. There is only a skull image as a graphical respond for the player, told them that they have successfully killed a mole. It is just fine until the new types of moles joined into the stage ---- they have hit points!

The problem is, during the gameplay, the player totally have no idea with:

- How much life left with the mole?
- Did I actually hit the mole?

- How long will it disappear?
- Did I kill the mole successfully? (it actually happens when the player killed it in the last few frames, before it disappear)
- When did this game finish and how much moles left?

So I made some new graphics and sound effects (described in the section Re-designed Moles and Sound Effects above) to solve those problems (but not all of them):

- The hit graphical and sound effect respond to the player, told them they actually got a successful hit;
- Different images represent moles' life state;
- An evident death animation (the mole made a rotation and also a skull fly out of it) with 4 different sound effects, told the player they killed the mole.

The program structure is also not that… easy to design. I haven't created that much new stuff but still – if this game getting bigger and bigger, it obviously needs to build on a game engine.

I hope that I can develop this project much longer. It was very fun. Still got many things wait to learn. The last time I use ActionScript was about 10 years ago – Adobe Flash was the Macromedia Flash 5.0. It is really changed a lot ;)

Bug Fixed

Player's Score Logical Issue

The "final score" has been sent with the "game over" event actually before the game finished. The reason is, when the timer counts end, the game is still on running (maybe the last 3 moles wait to be hit).

The bug has been fixed by a new logic. Now the game will check the mole holes after the timer finished its job. If all holes are empty, then report the "game over" event.

Missing Semicolon

MoleGameViewBase.as →function "startAnimationEnd()"→line "_moleGame.start()".