|  |
| --- |
| Dubit’s Coding Challenge |
| Specification Documents |
| Chuan Liu |

# Brief

## Coding Environment

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

## Coding Style

I read the code and…yes, the code is obviously easy to read and 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 focused on:

* Readability, comments;
* Less 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 going to disappear, 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 point 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 are something new added into the original Mole class to help this achieve:

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

## Build in Achievements / Awards (Accomplished)

The application will create an achievement system and pass reference it to the game. The 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 this report 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). It will light on the achievement label -- the one which has been sent by the event.

# Polishing, Embellishing and Extending

## Something New

* A new achievement “Perfect” has been added. It requires the player kill every single mole in a game;
* A new button has been added into score screen. It allows the player back to the introduction screen and re-selects a difficulty level – but the achievements will be reset as well.

# 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()”.