CS 5410 – Take Home Test
Mini Game

Introduction

A mini game is a game that lasts only a few seconds to a few minutes per play. Nintendo released an entire "game" made up of mini games called, "WarioWare". Each mini game lasted at most 5 seconds. Actually, these are called micro games and that is much closer to what you are going to do for this take home test.

Mini Game

Several years ago I played a game called Fable II, it was great fun. Within this game there are various ways to make money, through working in some profession: Blacksmith, Bartender, or Woodcutter. Essentially, each of these professions required me to play a mini game. In the mini game, I was presented with a meter of some fixed size, with a shrinking and expanding shaded region within the meter. A small marker moves back and forth across the whole distance of the meter. If I pressed the "A" button when the marker was in the shaded region, I earned some money and then the shaded region shrunk by some size, while the marker continued to move back and forth. If I pressed the "A" button when the marker was in the shaded region, I earned more money, with some multiplication factor from the previous "level". Once I pressed "A" outside the shaded region, the mini game reset and I had to start over from the largest shaded region and lowest level of money. I could keep doing this for as long as I wanted to keep making money. It is based upon this, you are to write the following mini game.

- Create a mini game that mirrors the above description. Refer to the mockups on the last page before reading the rest of these items.
- The marker moves at a constant rate back and forth over the whole meter, taking 1 second to go from one side to the other; independent of the size of the shaded region. If you want to be really cool, make it non-linear...start slower at the side, increase to a max speed by the middle and then slow back down by the other side; a sin wave would work well for this (no extra points, other than the coolness factor you'll feel inside).
- Each time the player successfully hits the button with the marker in the shaded region, provide a score, with some multiplier (between 1.0 and 2.0) for each level, shrink the shaded region and continue. Create a maximum of 6 levels. The first level uses a multiplier of 1.0, and the last 2.0. Once the smallest level is reached, stay at that level for as long as the player is successful.
- When the user successfully hits the button with the marker in the shaded region, create a ***tasteful*** (not too little, not too much) particle effect at that location, and play a sound indicating success.
- When the user hits the button with the marker outside the shaded region, give no score and the mini game is terminated, provide an indicate of "Game Over" and

return to the main menu. Because of this, make it VERY EASY to start over. Mini game ends, I press enter and I'm playing again.

- The player is not required to press a button on each pass. The marker simply moves back and forth as many times as the player wants before pressing the button.
- Show the current score.
- Track the top 5 high scores, storing them to local storage.
- Visual Style
  - Follow the visual style I have shown on the next page. You are welcome to make it look prettier, but only if it doesn't make it easier to code.
  - Black outline around the whole meter, and black outlines on the shaded region and marker.
  - Colors: feel free to use something else, but it must look "better."
    - Blue meter interior.
    - Green shaded region
    - Yellow marker
- Application Features
  - Use the Space-Bar as the button to press.
  - Main Menu (New Game, High Scores, Credits)
  - Create an index.html and then whatever else you need; JavaScript is separate from the html file. Node server not required, not desired, don't use it.

NOTES:

- **Due: Thursday, April 10th by 11:00 pm.**
- When complete submit a .zip file (and only .zip files, not .tar, not .rar, not .7z, just a .zip file) containing your entire project solution directory to Canvas.

Grading Breakdown

- Functionality (30%)
- Particle Effect (10%)
- Sound Effect (10%)
- Visual Appeal (10%)
- Ease of use (10%)
- Abstraction of input (10%) (vs hard-coding the key-press)
- High Scores (10%)
- Game State Management (i.e., menus) (10%)

Visual Appearance

Basic Meter

Meter with largest shaded region

Meter with largest shaded region and the marker

Smaller shaded region, with marker outside

Smallest shaded region