*[Preliminary Reading:]*

1. Make sure you fully understand Quintus (functionality and syntax) before attempting this homework. Consult with the Quintus online guides and documentation. Post onto the class Piazza board with your questions. *(You likely will have questions! Do not be embarrassed to post.)*

*[Get the Code:]*

1. We'll be modifying the "Block Break" game (introduced in class on 10/14/14). The code can be found on our class page, or Pascal Rettig's book [website](http://mh5gd.com/) under Chapter 11.

*[The Bare Minimum: Implement the following extensions to the Block Break game]  
(below tasks are ordered by approximate difficulty level)*

1. Change the headers so that Quintus is loaded from a CDN. (The Quintus CDN was discussed during our in-class programming on 10/16/14.)
2. Add a title screen to the game. Include instructions that the player should use the arrow keys to move.
3. Add a title screen for winning the game or losing the game.
4. Add sounds to the game. There should be at least 3 different sounds: (1) when the ball hits the paddle, (2) when the ball hits a block, (3) when the ball hits the left, right, or top wall. Consult with the Quintus Audio Documentation.
5. Add a couple of UI components, placed above the blocks: (1) score, (2) lives. Scoring should start at 0 when the game begins, and should increase when a ball hits a block. There should be three "lives" for each game, where the player loses a life if the ball goes beyond the bottom of the screen. Consult with the Quintus UI Example and the Quintus Game State object.

*[Extra Credit Tasks (valued up to an additional 100% of the homework value):]*

1. Using git to develop your project. You can: (1) use git on taz.cs.wcupa.edu, or (2) use [GitHub](http://github.com/) (either your own account or our class account.) Email me if you would like a repository on our [class page](https://github.com/WCU-CSC496-F14).
2. Allow the player's paddle to be controlled by the mouse.
3. Follow these [instructions](http://enclavegames.com/blog/2014/02/02/host-your-html5-games-on-github-pages/) for hosting your game on GitHub.
4. Create five different types of blocks, each with their own color. Each type of block should have a different point value associated with it. This also will require modifying the sprite sheet.
5. Additionally, at least one type of block should require multiple collisions before it is "destroyed".
6. Implement the functionality for a "magical, automatically win" block that is colored gold. If the ball hits this block, the player automatically wins. This block should not be static, but rather the block that is magical should cycle every second in a linear pattern. (Thus, the player can try to steer the ball to where the magic block will be in order to automatically win.)
7. An extra, cool modification of your choice. Points will be awarded based on the coolness and level of difficulty of the modification.
   * Write a few sentences about your modification in your submission. Tell me why it should be worth lots of points.

*[Test and Submit:]*

1. Make sure your code works in current versions of Firefox and Chrome. Recall that our preliminary observations in class showed that Firefox's JS engine tends to be pickier and fail soft more often than Chrome's.