Coliseum – Testing Document   
Jun Yen Leung - [junyenle@usc.edu](mailto:junyenle@usc.edu)  
Ethan Barker - [esbarker@usc.edu](mailto:esbarker@usc.edu)  
Jake Leventhal - [jglevent@usc.edu](mailto:jglevent@usc.edu)  
Uri Rahimi - [urahimi@usc.edu](mailto:urahimi@usc.edu)  
Dylan Appleton - [dappleto@usc.edu](mailto:dappleto@usc.edu)

**White Box Tests**

Login and Registration

* Test: Attempt to login without any text in either input field.  
  Expected Behavior: Error message should display to notify user that the username/password combination is invalid. User should be able to attempt to login again.
* Test: Attempt to login without a password.  
  Expected Behavior: Error message should display to notify user that the username/password combination is invalid. User should be able to attempt to login again.
* Test: Attempt to login without a username.  
  Expected Behavior: Error message should display to notify user that the username/password combination is invalid. User should be able to attempt to login again.
* Test: Attempt to login with the wrong password.  
  Expected Behavior: Error message should display to notify user that the username/password combination is invalid. User should be able to attempt to login again.
* Test: Attempt to login with a nonexistent username.  
  Expected Behavior: Error message should display to notify user that the username/password combination is invalid. User should be able to attempt to login again.
* Test: Attempt to login with a nonexistent password *and* nonexistent username.  
  Expected Behavior: Error message should display to notify user that the username/password combination is invalid. User should be able to attempt to login again.
* Test: Attempt to login with a valid username and password.  
  Expected Behavior: User should be logged in successfully and game should be playable by logged-in user.
* Test: Attempt to register without any text in either input field.  
  Expected Behavior: Error message should display to notify user that a username and password are required. User should be able to attempt to register again.
* Test: Attempt to register with a username over 16 characters.  
  Expected Behavior: Error message should display to notify user that usernames can only be up to 16 characters long. User should be able to attempt to register again.
* Test: Attempt to register without any text in the username input field.  
  Expected Behavior: Error message should display to notify user that a username is required. User should be able to attempt to register again.
* Test: Attempt to register without any text in the password input field.  
  Expected Behavior: Error message should display to notify user that a password is required. User should be able to attempt to register again.
* Test: Attempt to register with an existing username.  
  Expected Behavior: Error message should display to notify user that the supplied username is already taken. User should be able to attempt to register again.
* Test: Attempt to register with a unique username and a password.  
  Expected Behavior: User should be successfully registered. User should be redirected to the Login screen.

Highscores

* Test: Attempt to check the highscores list from the START screen.  
  Expected Behavior: User should be shown the current top 10 highscores.
* Test: Beat the boss with a new highscore.  
  Expected Behavior: User’s name and score should be saved to the highscores list.
* Test: Beat the boss to beat an old highscore by the same user.  
  Expected Behavior: User’s name and score should be saved to the highscores list (you may have more than one score on the list).

Gameplay

* Test: press the right arrow key button while fighting the BOSS.  
  Expected Behavior: The PC should move to the right.
* Test: press the left arrow key button while fighting the BOSS.  
  Expected Behavior: The PC should move to the left.
* Test: press the space button while fighting the BOSS **and** on the ground.  
  Expected Behavior: The PC should jump and should land on the ground afterwards.
* Test: press the space button while fighting the BOSS **and** in the air.  
  Expected Behavior: Nothing should happen. The PC should continue their original jump.
* Test: press left/right plus the right shift button while fighting the BOSS.  
  Expected Behavior: The PC should dash in the specified direction and be invulnerable during the dash.
* Test: press the z button while fighting the BOSS.  
  Expected Behavior: The PC should perform an attack. If in the air, the PC should perform an air attack.
* Test: attempt to walk off screen to the left and right.  
  Expected Behavior: The PC should collide with an invisible wall and be unable to walk off screen.
* Test: Let the PC get hit by the BOSS.  
  Expected Behavior: The PC’s health should decrease by one. Furthermore, the PC’s flinch animation should play.
* Test: Let the PC hit the BOSS.  
  Expected Behavior: The BOSS’ flinch animation should play.
* Test: Press escape while fighting the BOSS.  
  Expected Behavior: The game should pause.
* Test: Press escape while paused.  
  Expected Behavior: The game should resume.
* Test: Start the level.  
  Expected Behavior: The BOSS should start attacking the PC, the PC should be able to start moving and attacking, and the soundtrack should start to play.
* Test: (SOUND) “Miss” the BOSS with a PC attack.  
  Expected Behavior: The miss sound effect should play.
* Test: (SOUND) “Hit” the BOSS with a PC attack.  
  Expected Behavior: The hit sound effect should play.
* Test: (SOUND) “Hit” the PC with a BOSS attack.  
  Expected Behavior: The damage taken sound effect should play.
* Test: (SOUND) Execute a BOSS attack.  
  Expected Behavior: The BOSS’ attack sound effect should play.

Post-Gameplay

* Test: kill the player.  
  Expected Behavior: The PC should play its death animation and the screen should fade to the GAME OVER screen. The GAME OVER screen should fade back to the start screen.
* Test: kill the BOSS.  
  Expected Behavior: The BOSS should play its death animation and the screen should fade to the VICTORY screen. The VICTORY screen should show the player’s score and the list of highscores and then fade back to the start screen.

**Black Box Tests**

Login and Registration

* Test: Attempt to register with control characters in the username (e.g. \t, \n).  
  Expected Behavior: The game should not interpret these characters and instead register the user with those characters as plaintext (e.g. “Dave\nMiller”). The database should not have any problems storing this username.
* Test: Attempt to register with control characters in the password (e.g. \t, \n).  
  Expected Behavior: The game should not interpret these characters and instead register the user with those characters as plaintext (e.g. “Dave\tPW”). The database should not have any problems storing this password.
* Test: Attempt to register with foreign characters in the username (e.g. \t, \n).  
  Expected Behavior: An error message should be displayed if any non-UTF8 character is used. Otherwise, user should be successfully registered.
* Test: Attempt to register with foreign characters in the password (e.g. \t, \n).  
  Expected Behavior: An error message should be displayed if any non-UTF8 character is used. Otherwise, user should be successfully registered.
* Test: Hit the BOSS with the PC.  
  Expected Behavior: The BOSS’ health should decrease (invisible to player, but can be printed on console).
* Test: Let visual elements that expire do so (e.g. BOSS projectile attacks).  
  Expected Behavior: The projectile should no longer exist and thus should no longer affect what happens on-screen. However, effects from before the projectile’s “death” should persist (e.g. health lost on PC).

Highscores

* Test: Save a highscore for a player with control / foreign characters in their username.  
  Expected Behavior: highscore saving should function as normal.
* Test: Override an old highscore (beat the old score).  
  Expected Behavior: The old highscore should still be in the database. Query results will simply be sorted and limited (e.g. SELECT \* FROM highscores ORDER BY score DESC LIMIT 10).

Gameplay

* Test: Check that hitboxes and animations line up properly.  
  Expected Behavior: Hitboxes and animations should line up in an intuitive manner for smooth gameplay experience.
* Test: Observe BOSS attack choice (AI).  
  Expected Behavior: BOSS attack direction and choice of attack should depend (intuitively) on the PC’s position relative to the BOSS.

Post-Gameplay

* Test: kill the BOSS and PC at the same time.  
  Expected Behavior: This should count as GAME OVER (if the hero doesn’t survive, what’s the point?). Both death animations should play, but the screen should fade to the GAME OVER screen.

**Stress Tests**

Login and Registration

* Test: Attempt to register multiple users at the same time.  
  Expected Behavior: All new users should be registered successfully.
* Test: Attempt to register 100 users.  
  Expected Behavior: All new users should be registered successfully.
* Test: Attempt to Login multiple users at the same time.  
  Expected Behavior: All users should be logged in successfully.
* Test: Attempt to Login 100 users at the same time.  
  Expected Behavior: All new users should be logged in successfully.
* Test: Attempt to Login a user while registering a user.  
  Expected Behavior: The new user should be registered successfully and the logged in user should be logged in successfully.
* Test: Attempt to Login or register rapidly (e.g. mashing the Login button).  
  Expected Behavior: Only one request should be served per second.

Highscores

* Test: Attempt to view highscores while saving a highscore (on different users).  
  Expected Behavior: The highscores should be displayed to the user who asked to view them and the highscores list should update with the new user’s highscore. Which happens first depends on who clicked their button first.

Gameplay

* Test: Leave the game running for an hour.  
  Expected Behavior: Game should still run without problems (note that there will need to be someone playing the game for that hour or the BOSS will just kill the PC).
* Test: Run the game alongside other applications (web browser, etc.)  
  Expected Behavior: Game should still run smoothly.

Post-Gameplay

* N/A

**Unit Tests** (broadly categorized for organization – specific tests mostly covered in White / Black Box sections)

Login and Registration

* Test: Login Validation Tests – All white box tests on invalid login inputs.  
  Expected Behavior: As specified in the white box tests.
* Test: Registration Validation Tests – All white box tests on invalid registration inputs.  
  Expected Behavior: As specified in the white box tests.
* Test: Data Flow Tests – All white box and black box tests on inserting and retrieving data.  
  Expected Behavior: As specified in the white and black box tests.

Highscores

* Test: Highscore Saving Tests – All white box tests on inserting new highscores.  
  Expected behavior: As specified in white box tests.
* Test: Highscore Query Tests – All white box tests on pulling the highscore list.  
  Expected behavior: The most recent highscore list should be displayed.

Gameplay

* Test: Movement Functionality Tests – All white box tests on PC movement.  
  Expected behavior: As specified in white box tests.
* Test: Damage Functionality Tests – All white / black box tests on PC / BOSS damage.   
  Expected behavior: As specified in white / black box tests.
* Animation and Sound Tests – Test all gameplay actions (BOSS attack, BOSS gets hit, PC attack, PC gets hit, all PC movements, all BOSS attacks).  
  Expected Behavior: The proper animations and sound effects should play.

Post-Gameplay

* Appropriate Screen Test – Ensure that the proper screen displays for VICTORY or GAME OVER, and that the VICTORY screen has the appropriate data on it.  
  Expected Behavior: The proper screen should display based on whether the BOSS or PC dies.