

# Testing Document

Team Members	
Name	Email
Damiano Carrioli	carrioli@usc.edu
Chris Pack	cpack@usc.edu
Eric Yi-han Chen	ericyihc@usc.edu
Anant Chandra	anantcha@usc.edu

## Signing Up and Logging In:

- Testcase 1: (Username no longer than 20 chars)
  - Username: aaaaaaaaaaaaaaaaaaaaaa
  - Password: abc
  - Output: Username cannot be longer than 20 characters
- Testcase 2: (Password no longer than 20 chars)
  - Username: abc
  - Password: aaaaaaaaaaaaaaaaaaaaaa
  - Output: Password cannot be longer than 20 characters.
- Testcase 3: (Username cannot be empty)
  - Username: ""
  - Password: "abc"
  - Output: Username cannot be empty
- Testcase 4: (Password cannot be empty)
  - Username: "abc"
  - Password: ""
  - Output: Password cannot be empty
- Testcase 5: (no duplicate, username case insensitive, password case sensitive)
  - Username: abc
  - Password: abc
  - Username: aBc
  - Password: abc
  - Output: Username is taken, try again
- Testcase 6: New user makes an account
  - Sign up:
    - Username: abc
    - Password: abc

- Output: Username abc has not been used yet, so print “You have been signed up”, and then forward to gameplay
- Testcase 7: User from testcase 5 later tries to log in
  - Login:
    - Username: abc
    - Password: abc
    - Output: “You are logged in” and then to gameplay. (After the program checks username and password match a user in the database this is printed)
- Testcase 8: User tries to login to a nonexistent account
  - Login:
    - Username: ttrojan
    - Password: fighton
    - Output: Sorry, this user does not exist, please check your username or create a new account.
- Testcase 9: User tries to login to an account with incorrect password
  - Signup
    - Username: abc
    - Password: fighton
    - Output: Sorry, this password is incorrect, please try again
- Testcase 10:
  - User tries to sign in while still logged in
  - Output: Sorry: You are already logged in, if you would like to sign into a different account, please log out and then sign in to your other account.

Key pressed:

- Testcase 11: (Pressing keys other than right or left should have no effect):
  - Press any and as many keys as you want (except the right and left key)
  - Output: The plane/rocket should fall straight down
- Testcase 12: Moving the rocket to the right
  - Press the right key as the rocket falls
  - Output: The rocket should move to the right as it falls
- Testcase 13: Moving the rocket to the left
  - Press the left key as the rocket falls
  - Output: The rocket should move to the left as it falls
- Testcase 14: (Press ESC should go back to the main page, login status persists)
  - While in game screen, press escape key
  - Output: Loads homepage of game with the user maintaining their login status
- Testcase 15: (Pressing the left and right arrow keys simultaneously)

- If the user presses both the left and right keys simultaneously, the rocket should not change its falling position.

#### Graphics:

- Testcase 16: (rocket is halfway out of the platform)
  - (Change: This is counted as a landing)
  - This should be counted as a miss. The user can attempt the level again.
- Testcase 17: (Rocket can't go past bottom edge of screen)
  - (Change: Rocket quickly falls off screen in case of a crash)
  - As the rocket falls from the top of the screen, try to make it go all the way to the right as far as possible.
  - Output: The rocket should stop when it gets to the side of the screen and continue to fall.
- Testcase 18: (Rocket can't go past left edge of screen)
  - (Change: Rocket wraps around screen)
  - As the rocket falls from the top of the screen, try to make it go all the way to the left as far as possible.
  - Output: The rocket should stop when it gets to the side of the screen and continue to fall.
- Testcase 19: (Rocket can't go past right edge of screen)
  - (Change: Rocket wraps around screen)
  - As the rocket falls from the top of the screen, try to make it go all the way to the right as far as possible.
  - Output: The rocket should stop when it gets to the side of the screen and continue to fall.
- Testcase 20: (Rocket shouldn't go past the bottom of screen)
  - Let the rocket fall straight down without any input
  - Output: If it was lucky enough to spawn above the landing pad it should land on it. If it was unlucky it should crash when it hits the ground and stop.
- Testcase 21:
  - (Change: The text output)
  - Rocket lands on the landing pad after being steered by the user.
  - Output: Congratulations! You landed the Rocket! Proceed to the next level. The rocket is falling faster this time!
- Testcase 22: (Rocket lands with effect)
  - (Change: There is a sound effect, but no visual sign besides the congratulations statement)
  - Rocket lands on the landing pad after being steered by the user.
  - Output: A landing sound and visual effect like fire will be showed

- Testcase 23:
  - (Change: Rocket initialized at center of screen above the top of screen)
  - Rocket initialized at a random x position but a steady y position
  - Output: Start the game several times, and the rocket should start at different x position, but same y position
- Testcase 24:
  - Rocket initialized with increasing y velocity based on level.
  - Output: Start the game several times, and the rocket should start with different y velocity, based on the level of difficulty
- Testcase 25:
  - Rocket x position controlled by key pressed, each press results in a steady move of x position.
  - Output: start the game several times, and each time just press the left key once, and see if everytime the rocket lands/crashes at the same place
- Testcase 26:
  - The user steers the rocket, but missed the landing pad.
  - Output: Oops, you crashed the rocket! Better luck next time.

#### Database:

- Testcase 27: (If database is blank, show blank slots of scores in its place)
  - Testcase: Start server with no data and then have a user connect and login and try to view the leaderboard.
  - Output: There should be no entries, just the numbers 1-5 and an empty spot for the high scores after they are made.
- Testcase 28: (Update highest Score)
  - Score 1: 5
  - Score 2: 4
  - Score 3: 2
  - Score 4: 3
  - Score 5: 1
  - (Highest score should be 5 up to this point)
  - Output:
    - i. Username: 5
    - ii. Username: 4
    - iii. Username: 3
    - iv. Username: 2
    - v. Username: 1
  - Then another score is entered
  - Score 6: 7
  - (Highest score should be 7 and rest are shifted down and last is dropped)

- Output:
  - High Scores: ("Username" is replaced by the actual username)
  - i. Username: 7
  - ii. Username: 5
  - iii. Username: 4
  - iv. Username: 3
  - v. Username: 2
  - vi. Username: 1