**Test Plan**

for

Centipede Army Checkers

**Version 1.0 approved**

**Prepared by Adrienne Bowman,   
Cynthia Cordova, Rogelio Garza,   
Jacob Gibson, Brent Redmon**

**Centipede Army**

**October 29th 2018**

# Table of Contents

**Table Of Contents 1**

**Revisions 2**

**1 Introduction 3**

1.1 Test Plan Objectives 3

**2 Test Strategy 3**

2.1 System Test 3

2.2 Automated Test 3

2.3 Beta Test 3

2.4 User Acceptance Test 4

**3 Test Environments 4**

**4 Functions to be Tested 5**

4.1 User Interfaces (SRS 3.1) 5

4.2 Normal Starting Position (SRS 2.2.5) 6

4.3 Piece Movement Logic (SRS 2.2.3) 7

4.4 AI Implementation (Single Player Mode - SRS 4.1 and 2.2.2) 8

4.5 “King” Pieces (SRS 2.2.4) 9

4.6 Win/Loss State Determination (SRS 2.2.5) 9

4.7 Multiplayer Mode (SRS 4.2) 10

4.8 Spectator Mode (SRS 4.3) 10

4.9 Real Time SMS Update (Subscribe Mode - SRS 4.4) 11

# Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| 1.0 | Jacob G. | Wrote Remaining Tests and Added Screencaps | 10/28/2018 |
| 0.90 | Rogelio G. | Formatting | 10/28/2018 |
| 0.85 | Brent R. | Wrote Automated Testing | 10/23/2018 |
| 0.75 | Rogelio G. | Detailed Many Different Component Tests | 10/20/2018 |
| 0.5 | Jacob G. | Wrote Parts #1-3 | 10/17/2018 |
| 0.25 | Jacob G. | Foundation of Test Plan and Introduction created | 10/13/2018 |

# Introduction

The purpose of the following document is to provide a comprehensive Test Plan for the Centipede Checkers application, so that we as the developers may confirm that the stated requirements found inside of our SRS and that the intended functionality of the architecture as described in the SDD are all tested to satisfactory standards. After the execution of our Test Plan, the developers will have a clear understanding of whether we have met all of our client’s stated expectations.

* 1. **Test Plan Objectives**
* Define the Tools and Strategies to be used in creating tests for our software
* Determine how our various components will be tested
* Verify and Validate that each high-level requirement that has been delivered to us is met to the expectation of the customer

1. **Test Strategy**
   1. **System Test**

Our System Testing will primarily be composed by testing all functionality component-wise as described in detail in Section #4. Should all functionality be met, then the system’s usability will be at a high and acceptable standard. If any functionality is found to be significantly deficient or non-existent, then the system shall need to be reconfigured and serviced until it is brought to appropriate standard

* 1. **Automated Test**

Automated Testing of our product shall be conducted by executing a custom Python testing script that will check the following:

* Our product has opened up to the correct port (5000)
* Our server’s socket connections are empty upon launch
* Our NGROK connection is alive and reachable

Should any one of these conditions be false, then the test will fail and the script will report the results found.

* 1. **Beta Test**

All Centipede developers will be enlisting the help of their friends, family, and loved ones whom they wish to torment with endless questions about checkers in the process of Beta Testing our software. This will allow us to test our products functionality in the hands of the average user that is not closely affiliated to the inner workings or code base of our project.

* 1. **User Acceptance Test**

By the end of production and time of release, the end-user should be able to use Centipede Checkers as easily and intuitively as if a physical board was placed in front of them. Testing this acceptance with as diverse a range of age and technical prowess in our sample demographics will be useful in ensuring that the simplicity of our design has the broadest appeal possible to our audience.

1. **Test Environments**

Centipede Checkers has been created to run on the following operating systems and must therefore be tested accordingly on each iteration:

* Windows (7, 8, 8.1 10)
* MacOS (10.11, 10.12, 10.13, 10.14)
* Android (7, 8, 9)
* IOS (10, 11, 12)

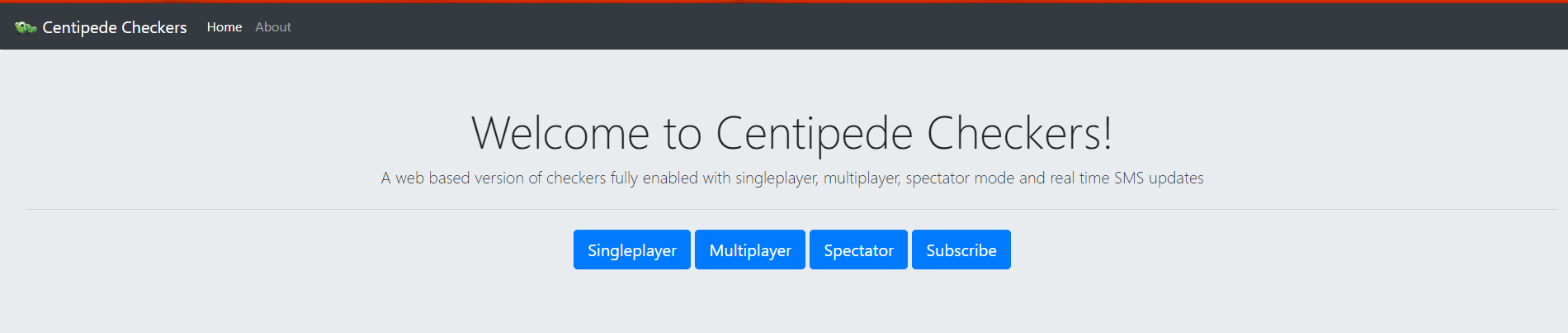
Centipede Checkers should be fully compatible without loss of functionality on each of the following internet browsers:

* Google Chrome version 69.0.4397 or later
* Safari version 11 or later
* Firefox 63 or later
* Microsoft Edge 42.17314 or later

Additional environments may be compatible, but not directly supported or considered, and will therefore remain untested.

1. **Functions To Be Tested**

## User Interfaces (SRS 3.1)



#### Test single player button

*Action: Click Single player button*

*Expected outcome: Start single player game*

*Actual outcome:* Start single player game

#### Test multiplayer button

*Action: Click multiplayer button*

*Expected outcome: Username pop up*

*Actual outcome:* Username pop up

#### Test username popup

* ***Test without a username***

*Action: click enter lobby without a username*

*Expected outcome: “you must specify a username” pop up*

*Actual outcome:* “you must specify a username” pop up

* ***Test with a username***

*Action: Click enter lobby with a username*

*Expected outcome: Start multiplayer game*

*Actual outcome:* Start multiplayer game

#### Test spectator button

*Action: Click spectator button*

*Expected outcome: view multiplayer game*

*Actual outcome:* view multiplayer game

#### Test Subscribe button

*Action: Click subscribe button*

*Expected outcome: load subscribe page*

*Actual outcome:* load subscribe page

#### Test home button

*Action: Click home button*

*Expected outcome: load home screen*

*Actual outcome:* load home screen

#### Test about button

*Action: Click about button*

*Expected outcome: load about screen*

*Actual outcome:* load about screen

## Normal Starting Position (SRS 2.2.5)

#### Test Normal Starting Position in Single Player

### *Action: Start single player game and inspect starting position*

*Expected outcome: Game will start in normal position*

*Actual outcome:* Game starts in normal position

#### Test Normal Starting Position in Multiplayer

### *Action: Start multiplayer game and inspect starting position*

*Expected outcome: Game will start in normal position*

*Actual outcome:* Game starts in normal position

## Piece Movement Logic (SRS 2.2.3)

#### Test select your man

*Action: Click blue man*

*Expected outcome: Select blue man*

*Actual outcome:* Select blue man

#### Test move man that is blocked

*Action: Try to move a blocked man (illegal move)*

*Expected outcome: nothing happens*

*Actual outcome:* nothing happens

#### Test move man backwards

*Action: Try to move a man backwards (illegal move)*

*Expected outcome: nothing happens*

*Actual outcome:* nothing happens

#### Test move man more than one space

*Action: Try to move a man more than one space (illegal move)*

*Expected outcome: nothing happens*

*Actual outcome:* nothing happens

#### Test move man

*Action: Move a man one legal space*

*Expected outcome: Man moves one space*

*Actual outcome:* Man moves one space

#### Test take an opponent’s man

*Action: Jump over an opponent's man*

*Expected outcome: Take opponent’s man*

*Actual outcome:* Take opponent’s man

#### Test take opponent’s 2nd man

*Action: Jump over two opponents men*

*Expected outcome: Take two opponents man*

*Actual outcome:* Take two opponents man

#### Test king me

*Action: Move man to other end*

*Expected outcome: Man becomes a king*

*Actual outcome:* Man becomes a king

#### Test move king backwards

*Action: Move king one legal space backwards*

*Expected outcome: King moves one space*

*Actual outcome:* King moves one space

#### Test move king blocked

*Action: Try to move a blocked king (illegal move)*

*Expected outcome: nothing happens*

*Actual outcome:* nothing happens

#### Test move king (illegal move)

*Action: Try to move king more than one space (illegal move)*

*Expected outcome: nothing happens*

*Actual outcome:* nothing happens

#### Test select opponents man

*Action: Click green man*

*Expected outcome: nothing happens*

*Actual outcome:* nothing happens

#### Test opponent takes a man

*Action: Opponent takes players man*

*Expected outcome: opponent takes players man*

*Actual outcome:* opponent takes players man

## AI Implementation (Single Player Mode - SRS 4.1 and 2.2.2)

#### Test AI moves its pieces

*Action: Watch AI move its pieces*

*Expected outcome: AI moves its man across the board in legal ways*

*Actual outcome:* AI moves its man across the board in legal ways

#### Test AI captures player’s pieces

*Action: Watch AI capture player’s piece*

*Expected outcome: AI uses its piece to capture player’s piece*

*Actual outcome:* AI uses its piece to capture player’s piece

## “King” Pieces (SRS 2.2.4)

#### Test Receiving a King in the King Position

*Action: Move man to king position*

*Expected outcome: Piece becomes a King*

*Actual outcome:* Piece becomes a King

#### Test King Movement Logic

*Action: Move King “backwards” with a King-only move*

*Expected outcome: King moves according to its ruleset*

*Actual outcome:* King moves according to its ruleset

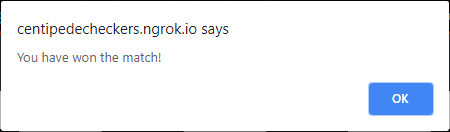
## Win/Loss State Determination (SRS 2.2.5)

#### Win State

*Action: Capture all of the opponent’s pieces*

*Expected outcome: Receive a “congratulations on winning” message*

*Actual outcome:* Receive a “congratulations on winning” message



#### Lose State

*Action: Have all of your pieces captured by the opponent*

*Expected outcome: Receive a “you lost” message*

*Actual outcome:* Receive a “you lost” message

## Multiplayer Mode (SRS 4.2)

#### Test Starting a Multiplayer Game with a Single Player

*Action: Begin a Multiplayer game with your chosen username*

*Expected outcome: Multiplayer game starts and waits for other player*

*Actual outcome:* Multiplayer game starts and waits for other player

#### Test Joining a Started Multiplayer Game

*Action: Enter the username of the multiplayer opponent*

*Expected outcome: Join Multiplayer game against opponent*

*Actual outcome:* Join Multiplayer game against opponent

#### Test Multiplayer Piece Logic

*Action: Play Multiplayer game and test piece logic using same steps*

*Expected outcome: All logic is followed as determined in the ruleset*

*Actual outcome:* All logic is followed as determined in the ruleset

## Spectator Mode (SRS 4.3)

#### Test Observing Game

*Action: start a spectating session*

*Expected outcome: watch the players conduct their game*

*Actual outcome: watch the players conduct their game*

#### Test Interfering with Game

*Action: Click on the pieces and board*

*Expected outcome: nothing happens*

*Actual outcome:* nothing happens

## Real Time SMS Update (Subscribe Mode - SRS 4.4)

#### Test subscribe

*Action: Click “subscribe” with no phone number entered*

*Expected outcome: “you must specify a number” pop up*

*Actual outcome:* “you must specify a number” pop up

*Action: Click “subscribe” with a phone number with too few numbers entered*

*Expected outcome: “please enter a correct phone number” pop up*

*Actual outcome:* Load “Thank you for registering” page

*Action: Click “subscribe” with a phone number too many numbers entered*

*Expected outcome: “please enter a correct phone number” pop up*

*Actual outcome:* Load “Thank you for registering” page

*Action: Click “subscribe” with letters entered*

*Expected outcome: “please enter a correct phone number” pop up*

*Actual outcome:* Load “Thank you for registering” page

*Action: Click “subscribe” with a phone number entered*

*Expected outcome: Load “Thank you for registering” page*

*Actual outcome:* Load “Thank you for registering” page

#### Test unsubscribe

*Action: Click “unsubscribe” with no phone number entered*

*Expected outcome: “you must specify a number” pop up*

*Actual outcome:* “you must specify a number” pop up

*Action: Click “unsubscribe” with a phone number with too few numbers*

*entered*

*Expected outcome: “please enter a correct phone number” pop up*

*Actual outcome:* “You have unsubscribed from future updates” pop up

*Action: Click “unsubscribe” with a phone number with too many numbers*

*entered*

*Expected outcome: “please enter a correct phone number” pop up*

*Actual outcome:* “You have unsubscribed from future updates” pop up

*Action: Click “unsubscribe” with letters entered*

*Expected outcome: “please enter a correct phone number” pop up*

*Actual outcome:* “You have unsubscribed from future updates” pop up

*Action: Click “unsubscribe” with a wrong phone number entered*

*Expected outcome: “You are not currently subscribed” pop up*

*Actual outcome: “*You have unsubscribed from future updates” pop up

*Action: Click “unsubscribe” with a valid phone number entered*

*Expected outcome: “You have unsubscribed from future updates” pop up*

*Actual outcome:* “You have unsubscribed from future updates” pop up