Melissa Jones, Nicholas Benyo, Scott Moser
Seattle University – Professor McKee
CPSC5210-01
June 6, 2019

Milestone 2: Build & Regression Test Scripts

I. buildTestSuite.sh (source)

```
#!/bin/bash
#================================================================================
# Team D'Buggers (Team 7)
# Scott Moser, Nicholas Benyo, Melissa Jones
# Professor McKee
# CPSC 5200-01
# 6 June 2019
#                               Milestone #2
#
# File: buildTestSuite.sh
#
# Description:
# This shell script builds the Battleship project and associated J-Unit unit
# tests.
#
#   DEPENDENCIES, LIMITATIONS, & DESIGN NOTES:
#       Dependencies :
#           1. JDK 1.8.0_211 must be installed at $JDK_PATH.
#       Design Notes :
#           1. All source files are built, then all J-Unit test files are built.
#           2. Artifacts are placed in the $OUT_DIR directory.
#       Limitations :
#           1. Due to memory limitations on SU's CS1 server, the $CS1_HACK
#              variable is used to limit the memory used by the JVM.
#
#   Example Usage:
#   "./buildTestSuite.sh"
#================================================================================

#set -o errexit
#set -o pipefail
set -o nounset
#set -o xtrace

#================================================================================
# Constants
#================================================================================
OUT_DIR="out"
PROJECT_ROOT=".."
JDK_PATH="$PROJECT_ROOT/../jdk1.8.0_211/bin"
JUNIT_JAR="junit/junit-platform-console-standalone-1.5.0-M1.jar"

# Hack required due to memory limitations on CS1
CS1_HACK="-J-Xmx512m"  # Limit heap to 512 MB

# NOTE: This list needs to be updated if any additional source files are added
#       to the project!
sourceList=(
```

```
    Battleship
    Grid
    Location
    Randomizer
    Ship
)

# NOTE: This list needs to be updated if any additional test files are added
#       to the project!
testList=(
    LocationTest
    ShipTest
    GridTest
)

#==============================================================================
# Script
#==============================================================================
echo "Building test suite..."

echo "Creating directory for build artifacts..."
mkdir -p $PROJECT_ROOT/$OUT_DIR

echo "Removing stale artifacts..."
rm -f -v $PROJECT_ROOT/$OUT_DIR/*.class

echo "Printing javac version..."
$JDK_PATH/javac $CS1_HACK -version

echo "Building source code..."
for i in ${sourceList[@]}; do
    echo "Building: $i.java --> $i.class"
    $JDK_PATH/javac $CS1_HACK -d $PROJECT_ROOT/$OUT_DIR -cp $PROJECT_ROOT
$PROJECT_ROOT/$i.java
    if [ $? != 0 ]; then
        echo "ERROR: Unable to build $i.java! Aborting build..."
        exit 1
    fi
done

echo "Building unit tests..."
for i in ${testList[@]}; do
    echo "Building: $i.java --> $i.class"
    $JDK_PATH/javac $CS1_HACK -d $PROJECT_ROOT/$OUT_DIR -cp
$PROJECT_ROOT/$OUT_DIR:$PROJECT_ROOT/$JUNIT_JAR $PROJECT_ROOT/$i.java
    if [ $? != 0 ]; then
        echo "ERROR: Unable to build $i.java! Aborting build..."
        exit 2
    fi
done

echo "Build success!"
exit 0
```

II. buildTestSuite.sh (output)

The following output was obtained by running the script on SU's CS1 server.

```
[mosers1@cs1 scripts]$ pwd
```

```
/home/st/mosers1/cpsc5210/buildSystem/Java-Battleship/scripts
[mosers1@cs1 scripts]$
[mosers1@cs1 scripts]$ ./buildTestSuite.sh
Building test suite...
Creating directory for build artifacts...
Removing stale artifacts...
removed '../out/Battleship.class'
removed '../out/Grid.class'
removed '../out/Location.class'
removed '../out/LocationTest.class'
removed '../out/Randomizer.class'
removed '../out/Ship.class'
removed '../out/ShipTest.class'
Printing javac version...
javac 1.8.0_211
Building source code...
Building: Battleship.java --> Battleship.class
Building: Grid.java --> Grid.class
Building: Location.java --> Location.class
Building: Randomizer.java --> Randomizer.class
Building: Ship.java --> Ship.class
Building unit tests...
Building: LocationTest.java --> LocationTest.class
Building: ShipTest.java --> ShipTest.class
Building: GridTest.java --> GridTest.class
Build success!
```

III. runTestSuite.sh (source)

```
#!/bin/bash
#==============================================================================
# Team D'Buggers (Team 7)
# Scott Moser, Nicholas Benyo, Melissa Jones
# Professor McKee
# CPSC 5200-01
# 6 June 2019
#                              Milestone #2
#
# File: runTestSuite.sh
#
# Description:
# This shell script executes all J-Unit test cases in the $OUT_DIR directory
# a specificed number of times and e-mails the results to an e-mail address, if
# provided.
#
#   DEPENDENCIES, LIMITATIONS, & DESIGN NOTES:
#       Dependencies :
#           1. JDK 1.8.0_211 must be installed at $JDK_PATH.
#       Design Notes :
#           1. All J-Unit test cases in the $OUT_DIR directory are run.
#           2. Console output is logged to $LOG_FILE.
#           3. Caller can specify the number of test iterations.
#           4. Caller can optionally specify an e-mail address.
#           5. If an e-mail is provided, the test results are set and the
#              console output is attached to the e-mail.
#           6. Test results and statistics are calculated and displayed.
#       Limitations :
#           1. Due to memory limitations on SU's CS1 server, the $CS1_HACK
```

```
#              variable is used to limit the memory used by the JVM.
#
#   Example Usage:
#   1. Executes the test suite one time
#         ./runTestSuite.sh 1
#   2. Executes the test suite twice and e-mails results to test@gmail.com.
#         ./runTestSuite.sh 2 test@gmail.com
#   3. Executes the test suite twice and e-mails the results to multiple
#      recipients.
#         ./runTestSuite.sh 2 "test1@gmail.com test2@gmail.com test3@gmail.com"
#===============================================================================

#set -o errexit
set -o pipefail
set -o nounset
#set -o xtrace

#===============================================================================
# Constants
#===============================================================================
MIN_ARGS=1
MAX_ARGS=2

OUT_DIR="out"
PROJECT_ROOT=".."
JDK_PATH="$PROJECT_ROOT/../jdk1.8.0_211/bin"
JUNIT_JAR="junit/junit-platform-console-standalone-1.5.0-M1.jar"

# Hack required due to memory limitations on CS1
CS1_HACK="-Xmx512m"  # Limit heap to 512 MB

LOG_FILE="runTestSuiteLog.txt"

#===============================================================================
# Script
#===============================================================================

# Check number of arguments
if [ "$#" -lt "$MIN_ARGS" ] || [ "$#" -gt "$MAX_ARGS" ]; then
    echo "ERROR: Invalid number of command-line arguments!"
    echo "Usage:"
    echo "   ./runTestSuite <numIter> [emailRecipient]"
    echo "     - <numIter> - Number of times to run the test suite. Range:
[1,10000)"
    echo "     - [emailRecipient] - Optional e-mail address to notify with test
results"
    exit 1
fi

# Basic input validation for number of iterations
if [ "$1" -le "0" ]; then
    echo "ERROR: Caller must enter numIter of 1 or more!"
    exit 2
fi

# Remove previous log files
echo Cleaning up stale log files...
rm -f -v $LOG_FILE
```

```
# Capture system-level information for future debug-ability
echo -e "
================================================================================
Environment Variables
================================================================================
$(env)" >> $LOG_FILE

echo -e "
================================================================================
Java Version
================================================================================
" | tee -a $LOG_FILE
echo $($JDK_PATH/java $CS1_HACK -version 2>&1) | tee -a $LOG_FILE
echo JUnit jar file: $JUNIT_JAR | tee -a $LOG_FILE

echo -e "
================================================================================
Test Suite Execution
================================================================================
" | tee -a $LOG_FILE

# Run the test suite
numIter=$1
numPass=0
numFail=0
echo "Start date/time:" $(date)  | tee -a $LOG_FILE
start=$SECONDS
for i in `seq 1 $numIter`; do
    echo "************************************************" | tee -a $LOG_FILE
    echo "Executing test run $i of $1..." | tee -a $LOG_FILE
    echo "************************************************" | tee -a $LOG_FILE
    $JDK_PATH/java $CS1_HACK -jar $PROJECT_ROOT/$JUNIT_JAR --class-path
$PROJECT_ROOT/$OUT_DIR --scan-class-path | tee -a $LOG_FILE
    rv=$?
    echo "Tests completed with return code: $rv" | tee -a $LOG_FILE
    if [ $rv == 0 ]; then
        let "numPass++"
    else
        let "numFail++"
    fi
done
stop=$SECONDS
echo "Stop date/time:" $(date) | tee -a $LOG_FILE

# Calculate statistics
  # Total execution time
duration=$(( $stop - $start ))
  # Search log file for the number of test cases per test suite run
numCases=$(grep -i -m1 "tests found" $LOG_FILE | grep -o -E '[0-9]+')
  # Multiply test cases per run times the number of runs
numCases=$((numCases * numIter))
  # Search the log file for the number of failing test cases
numCaseFail=$(grep -i -c "AssertionFailedError" $LOG_FILE)
  # The remainder are passing tests
numCasePass=$((numCases - numCaseFail))
  # Calculate the individual test case passing rate
testCasePassRate=$(bc -l <<< "scale=2; $numCasePass/$numCases*100")
```

```
  # Calculate the test suite passing rate
testSuitePassRate=$(bc -l <<< "scale=2; $numPass/$numIter*100")

# Determine overall result
rval=3
status="FAIL"
if [ "$numPass" -eq "$numIter" ]; then
    rval=0
    status="PASS"
fi

# Report statistics
echo ""
| tee -a $LOG_FILE
echo
"================================================================================"
| tee -a $LOG_FILE
echo "Results & Statistics"
| tee -a $LOG_FILE
echo
"================================================================================"
| tee -a $LOG_FILE
echo "Overall test suite result:  $status"
| tee -a $LOG_FILE
echo "Execution time:             $duration [seconds]"
| tee -a $LOG_FILE
echo ""
| tee -a $LOG_FILE
echo "# of test suite runs:       $numIter"
| tee -a $LOG_FILE
echo "# of passing runs:          $numPass"
| tee -a $LOG_FILE
echo "# of failing runs:          $numFail"
| tee -a $LOG_FILE
echo "Test suite passing rate:    $testSuitePassRate%"
| tee -a $LOG_FILE
echo ""
| tee -a $LOG_FILE
echo "# of test cases run:        $numCases"
| tee -a $LOG_FILE
echo "# of passing test cases:    $numCasePass"
| tee -a $LOG_FILE
echo "# of failing test cases:    $numCaseFail"
| tee -a $LOG_FILE
echo "Test case passing rate:     $testCasePassRate%"
| tee -a $LOG_FILE
echo
"================================================================================"
| tee -a $LOG_FILE
echo "Please see the log file for the full console output: $LOG_FILE"
echo ""

# Notify e-mail recipient if one was provided by caller
if [ $# == $MAX_ARGS ]; then
    echo "Sending e-mail report to $2..."
    messageBody="Please view the attached file for full console output of the test
run."
```

```
    echo $messageBody | mail -s "[$USER] Test Suite Run -- $status!" -a $LOG_FILE
$2
    if [ $? != 0 ]; then
        echo "ERROR: Unable to send e-mail! rv = $?"
        exit 4
    fi
fi

# Return status code
exit $rval
```

IV. runTestSuite.sh (output)

The following output was obtained by running the script on SU's CS1 server.

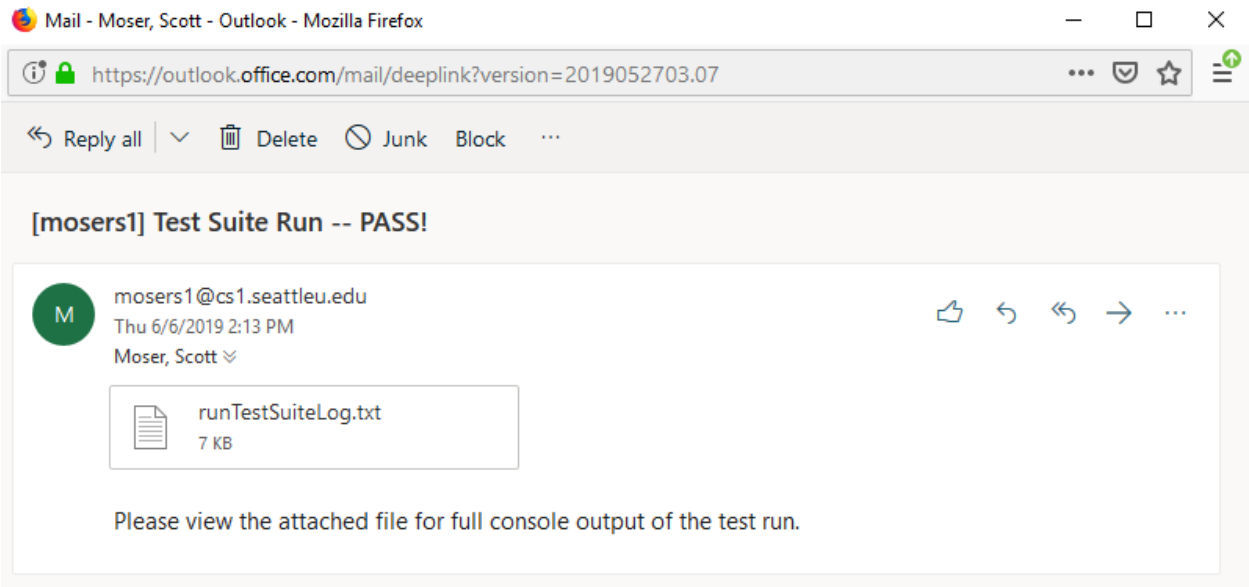
```
[mosers1@cs1 scripts]$ pwd
/home/st/mosers1/cpsc5210/buildSystem/Java-Battleship/scripts
[mosers1@cs1 scripts]$
[mosers1@cs1 scripts]$ ./buildTestSuite.sh
Building test suite...
Creating directory for build artifacts...
Removing stale artifacts...
removed '../out/Battleship.class'
removed '../out/Grid.class'
removed '../out/Location.class'
removed '../out/LocationTest.class'
removed '../out/Randomizer.class'
removed '../out/Ship.class'
removed '../out/ShipTest.class'
Printing javac version...
javac 1.8.0_211
Building source code...
Building: Battleship.java --> Battleship.class
Building: Grid.java --> Grid.class
Building: Location.java --> Location.class
Building: Randomizer.java --> Randomizer.class
Building: Ship.java --> Ship.class
Building unit tests...
Building: LocationTest.java --> LocationTest.class
Building: ShipTest.java --> ShipTest.class
Building: GridTest.java --> GridTest.class
Build success!
[mosers1@cs1 scripts]$ ^C
[mosers1@cs1 scripts]$ ^C
[mosers1@cs1 scripts]$ clear
[mosers1@cs1 scripts]$ pwd
/home/st/mosers1/cpsc5210/buildSystem/Java-Battleship/scripts
[mosers1@cs1 scripts]$
[mosers1@cs1 scripts]$ ./runTestSuite.sh 1 mosers1@seattleu.edu
Cleaning up stale log files...

================================================================================
Java Version
================================================================================

java version "1.8.0_211" Java(TM) SE Runtime Environment (build 1.8.0_211-b12) Java
HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)
JUnit jar file: junit/junit-platform-console-standalone-1.5.0-M1.jar
```

```
================================================================================
Test Suite Execution
================================================================================

Start date/time: Thu Jun 6 14:13:52 PDT 2019
************************************************
Executing test run 1 of 1...
************************************************

Thanks for using JUnit! Support its development at https://junit.org/sponsoring

╷
├─ JUnit Jupiter ✔
│  ├─ LocationTest ✔
│  │  ├─ testCheckHit() ✔
│  │  ├─ testIsUnguessed() ✔
│  │  ├─ testGetSetDirectionOfShip() ✔
│  │  ├─ testLocation() ✔
│  │  ├─ testSetHasShip() ✔
│  │  ├─ testGetSetLengthOfShip() ✔
│  │  ├─ testMarkHit() ✔
│  │  ├─ testMarkMiss() ✔
│  │  ├─ testCheckMiss() ✔
│  │  └─ testGetSetStatus() ✔
│  ├─ ShipTest ✔
│  │  ├─ testShip() ✔
│  │  ├─ testSetLocation() ✔
│  │  ├─ testToString() ✔
│  │  ├─ testIsDirectionSet() ✔
│  │  ├─ testGetDirection() ✔
│  │  ├─ testIsLocationSet() ✔
│  │  ├─ testGetLength() ✔
│  │  ├─ testGetCol() ✔
│  │  ├─ testGetRow() ✔
│  │  └─ testSetDirection() ✔
│  └─ GridTest ✔
│     ├─ testAddShip() ✔
│     ├─ testGet() ✔
│     ├─ testGrid() ✔
│     ├─ testswitchCounterToIntegerForArray() ✔
│     ├─ testHasLost() ✔
│     ├─ testHasShip() ✔
│     ├─ testMarkHit() ✔
│     ├─ testAlreadyGuessed() ✔
│     ├─ testMarkMiss() ✔
│     ├─ testSetStatus() ✔
│     ├─ testGetStatus() ✔
│     ├─ testNumCol() ✔
│     ├─ testSetShip() ✔
```

```
|     └─ testNumRows() ✔
└─ JUnit Vintage ✔

Test run finished after 107 ms
[         5 containers found      ]
[         0 containers skipped    ]
[         5 containers started    ]
[         0 containers aborted    ]
[         5 containers successful ]
[         0 containers failed     ]
[        34 tests found           ]
[         0 tests skipped         ]
[        34 tests started         ]
[         0 tests aborted         ]
[        34 tests successful      ]
[         0 tests failed          ]

Tests completed with return code: 0
Stop date/time: Thu Jun 6 14:13:52 PDT 2019

================================================================================
Results & Statistics
================================================================================
Overall test suite result:  PASS
Execution time:             0 [seconds]

# of test suite runs:       1
# of passing runs:          1
# of failing runs:          0
Test suite passing rate:    100.00%

# of test cases run:        34
# of passing test cases:    34
# of failing test cases:    0
Test case passing rate:     100.00%
================================================================================
Please see the log file for the full console output: runTestSuiteLog.txt

Sending e-mail report to mosers1@seattleu.edu...
```

V. E-mail Content

The following e-mail was an actual e-mail sent at the end of our regression test script.



Below is the content of the runTestSuiteLog.txt file that was attached to the e-mail:

```
================================================================================
Environment Variables
================================================================================
XDG_SESSION_ID=53668
HOSTNAME=cs1.seattleu.edu
SELINUX_ROLE_REQUESTED=
SHELL=/bin/bash
TERM=xterm
HISTSIZE=1000
SSH_CLIENT=67.160.11.198 62457 22
SELINUX_USE_CURRENT_RANGE=
SSH_TTY=/dev/pts/5
USER=mosers1
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33
;01:or=40;31;01:mi=01;05;37;41:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:e
x=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=
01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=0
1;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz
=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01
;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;3
1:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:
*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.
xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.
pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.
ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wm
v=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=
01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;3
5:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=01;36:*.au=01;36:*.flac=01;36:
*.mid=01;36:*.midi=01;36:*.mka=01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.
wav=01;36:*.axa=01;36:*.oga=01;36:*.spx=01;36:*.xspf=01;36:
PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/st/mosers1/.local/bin:/ho
me/st/mosers1/bin:/home/st/mosers1/mpich-
install/bin:/home/st/mosers1/.local/bin:/home/st/mosers1/bin
MAIL=/var/spool/mail/mosers1
PWD=/home/st/mosers1/cpsc5210/buildSystem/Java-Battleship/scripts
```

```
LANG=en_US.utf8
SELINUX_LEVEL_REQUESTED=
KRB5CCNAME=KEYRING:persistent:2850
HISTCONTROL=ignoredups
HOME=/home/st/mosers1
SHLVL=2
LOGNAME=mosers1
CVS_RSH=ssh
SSH_CONNECTION=67.160.11.198 62457 10.124.72.20 22
XDG_DATA_DIRS=/home/st/mosers1/.local/share/flatpak/exports/share:/var/lib/flatpak/exp
orts/share:/usr/local/share:/usr/share
LESSOPEN=||/usr/bin/lesspipe.sh %s
XDG_RUNTIME_DIR=/run/user/2850
_=/usr/bin/env

================================================================================
Java Version
================================================================================

java version "1.8.0_211" Java(TM) SE Runtime Environment (build 1.8.0_211-b12) Java
HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)
JUnit jar file: junit/junit-platform-console-standalone-1.5.0-M1.jar

================================================================================
Test Suite Execution
================================================================================

Start date/time: Thu Jun 6 14:13:52 PDT 2019
************************************************
Executing test run 1 of 1...
************************************************

Thanks for using JUnit! Support its development at https://junit.org/sponsoring

□[36mâ•·□[0m
□[36mâ”œâ”€□[0m □[36mJUnit Jupiter□[0m □[32mâœ”□[0m
□[36mâ”‚  â”œâ”€□[0m □[36mLocationTest□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestCheckHit()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestIsUnguessed()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestGetSetDirectionOfShip()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestLocation()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestSetHasShip()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestGetSetLengthOfShip()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestMarkHit()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestMarkMiss()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestCheckMiss()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€â”€□[0m □[34mtestGetSetStatus()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”œâ”€□[0m □[36mShipTest□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestShip()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestSetLocation()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestToString()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestIsDirectionSet()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestGetDirection()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestIsLocationSet()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestGetLength()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestGetCol()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€□[0m □[34mtestGetRow()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚  â”œâ”€â”€□[0m □[34mtestSetDirection()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”œâ”€â”€□[0m □[36mGridTest□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚     â”œâ”€□[0m □[34mtestAddShip()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚     â”œâ”€□[0m □[34mtestGet()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚     â”œâ”€□[0m □[34mtestGrid()□[0m □[32mâœ”□[0m
□[36mâ”‚  â”‚     â”œâ”€□[0m □[34mtestswitchCounterToIntegerForArray()□[0m □[32mâœ”□[0m
```

```
□[36mâ”‚     â”œâ”€□[0m □[34mtestHasLost()□[0m □[32mâœ”□[0m
□[36mâ”‚     â”œâ”€□[0m □[34mtestHasShip()□[0m □[32mâœ”□[0m
□[36mâ”‚     â”œâ”€□[0m □[34mtestMarkHit()□[0m □[32mâœ”□[0m
□[36mâ”‚     â”œâ”€□[0m □[34mtestAlreadyGuessed()□[0m □[32mâœ”□[0m
□[36mâ”‚     â”œâ”€□[0m □[34mtestMarkMiss()□[0m □[32mâœ”□[0m
□[36mâ”‚     â”œâ”€□[0m □[34mtestSetStatus()□[0m □[32mâœ”□[0m
□[36mâ”‚     â”œâ”€□[0m □[34mtestGetStatus()□[0m □[32mâœ”□[0m
□[36mâ”‚     â”œâ”€□[0m □[34mtestNumCol()□[0m □[32mâœ”□[0m
□[36mâ”‚     â”œâ”€□[0m □[34mtestSetShip()□[0m □[32mâœ”□[0m
□[36mâ”‚     â””â”€□[0m □[34mtestNumRows()□[0m □[32mâœ”□[0m
□[36mâ””â”€□[0m □[36mJUnit Vintage□[0m □[32mâœ”□[0m

Test run finished after 107 ms
[         5 containers found      ]
[         0 containers skipped    ]
[         5 containers started    ]
[         0 containers aborted    ]
[         5 containers successful ]
[         0 containers failed     ]
[        34 tests found           ]
[         0 tests skipped         ]
[        34 tests started         ]
[         0 tests aborted         ]
[        34 tests successful      ]
[         0 tests failed          ]

Tests completed with return code: 0
Stop date/time: Thu Jun 6 14:13:52 PDT 2019

================================================================================
Results & Statistics
================================================================================
Overall test suite result:  PASS
Execution time:             0 [seconds]

# of test suite runs:       1
# of passing runs:          1
# of failing runs:          0
Test suite passing rate:    100.00%

# of test cases run:        34
# of passing test cases:    34
# of failing test cases:    0
Test case passing rate:     100.00%
================================================================================
```