

Tarea 1: *Sink a ship* (Hundir un barco).

Versión simple

Fecha de entrega: 1 de Setiembre, 2022

Un juego tipo Battleship. Usuario vs la computadora. La diferencia es que en este juego el usuario **no** posee barcos. En su lugar, el trabajo del usuario será hundir los barcos que posee la computadora (el programa) en el menor número de intentos.

Objetivo

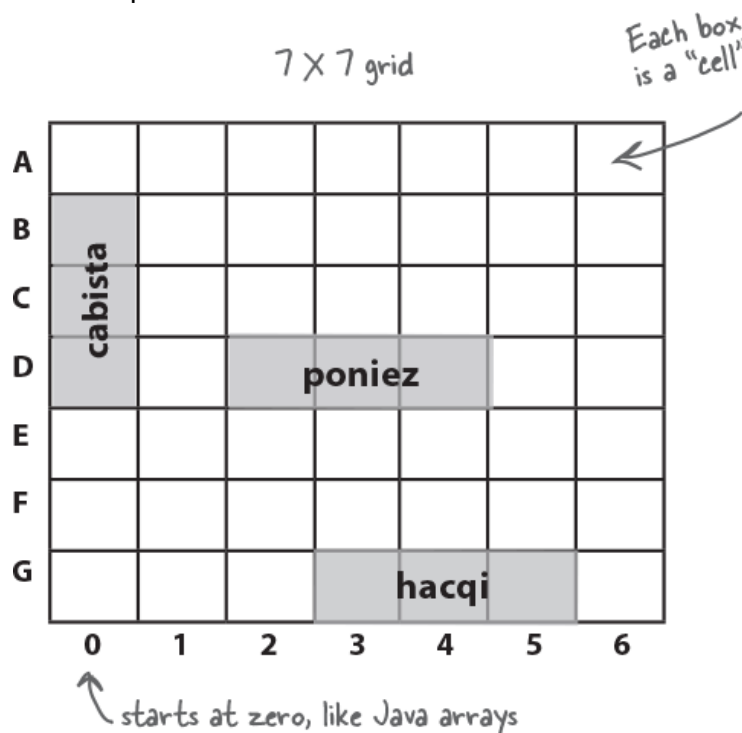
Hundir todos los barcos de la computadora en el menor número de intentos. Se le otorgará una calificación o nivel, según su desempeño.

Configuración

Cuando el programa se ejecute, la computadora ubicará tres barcos en una cuadrícula de 7 x 7. Luego de esto, el juego pregunta por el primer intento.

¿Cómo se juega?

La computadora le pedirá que ingrese una suposición (una celda) que escribirá en la línea de comando como "A3", "C5", etc. En respuesta a su suposición, verá un resultado en la línea de comandos, ya sea "hit", "miss" o "You sunk ship-1". Cuando haya hundido los tres barcos, el juego termina imprimiendo su calificación.



File Edit Window Help Sell

```
%java StartupBust
```

```
Enter a guess  A3
```

```
miss
```

```
Enter a guess  B2
```

```
miss
```

```
Enter a guess  C4
```

```
miss
```

```
Enter a guess  D2
```

```
hit
```

```
Enter a guess  D3
```

```
hit
```

```
Enter a guess  D4
```

```
Ouch! You sunk poniez    : (
```

```
kill
```

```
Enter a guess  G3
```

```
hit
```

```
Enter a guess  G4
```

```
hit
```

```
Enter a guess  G5
```

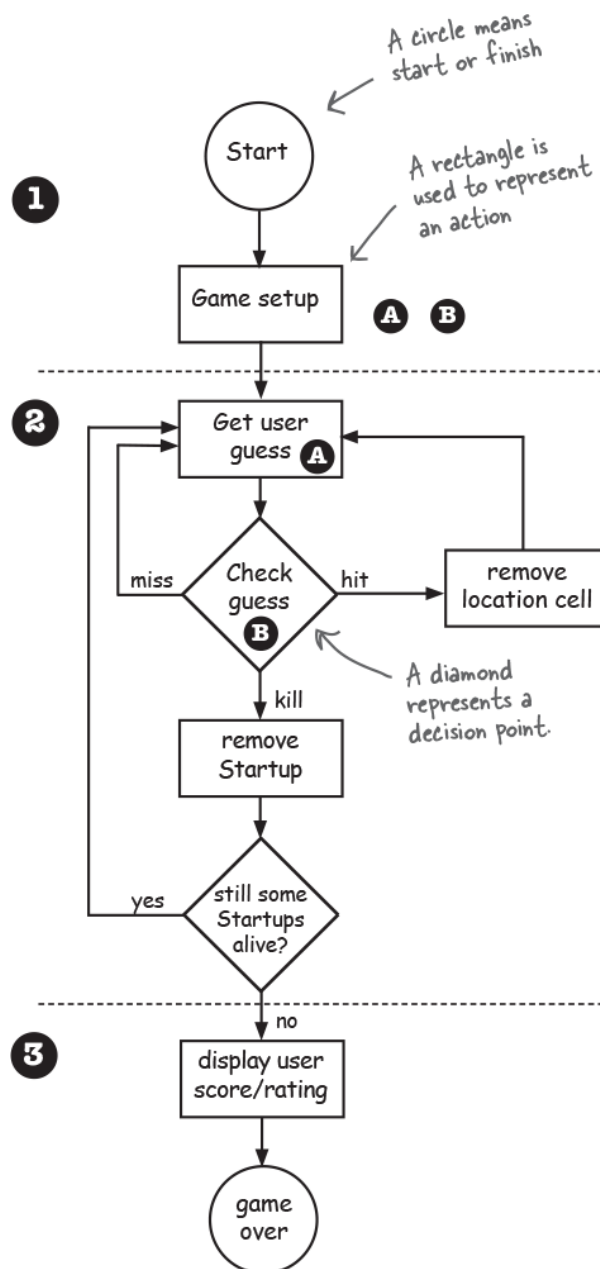
```
Ouch! You sunk hacqi    : (
```

```
All Startups are dead! Your stock  
is now worthless
```

```
Took you long enough. 62 guesses.
```

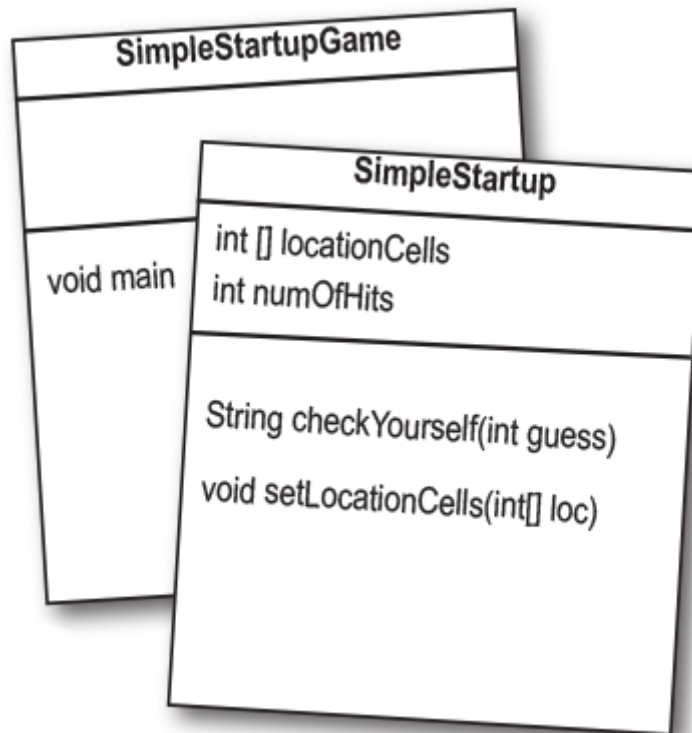
Diseño de alto nivel

1. El usuario inicia el juego
 - a. El juego crea un barco
 - b. El juego pone el barco en la cuadrícula
2. Comienza el juego
 - a. Solicitar al usuario por una suposición ("A2", "C0", "B3", etc)
 - b. Verificar si la suposición del usuario es un "hit" o un "miss". Si un barco se hunde entonces borrarlo de la cuadrícula.
3. El juego finaliza



Versión para la tarea 1: “Simple Battleship Game”

- En lugar de una cuadrícula, se va a esconder el barco en un *array* (una dimensión).
- En lugar de tres barcos, será solo uno.
- Luego de lo anterior, el objetivo del juego es el mismo.



Las clases en la tarea deberán llamarse **BattleshipGame.java** y **Battleship.java**

1. Cuando el juego inicia crea un barco y proporciona una localización en tres de las siete celdas del *array*.

En lugar de “A2” o “C4”, las localizaciones son enteros simples (por ejemplo 1, 2, 3 son las localizaciones de las celdas).



2. Comienza el juego. Se le solicita al usuario una suposición (un intento); luego se verifica si alguna de las celdas del barco fue impactada. Si hay un impacto, incrementar la variable `numOfHits`.

3. El juego finaliza cuando las tres celdas han sido impactadas (el valor de la variable `numOfHits` es igual a 3)

```
File Edit Window Help Destroy
%java SimpleStartupGame
enter a number 2
hit
enter a number 3
hit
enter a number 4
miss
enter a number 1
kill
You took 4 guesses
```

El proyecto se debe desarrollar con Apache Maven.

GroupId: tec.poo.tareas
ArtifactId: simple-battleship
Version: 1.0.0

Incluir el siguiente código debajo de la sección <plugins>

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>exec-maven-plugin</artifactId>
  <version>3.0.0</version>
  <executions>
    <execution>
      <goals>
        <goal>java</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <mainClass>tec.poo.tarea1.BattleshipGame</mainClass>
    <cleanupDaemonThreads>false</cleanupDaemonThreads>
  </configuration>
</plugin>
```

Algunos comandos de ayuda

```
mvn clean compile
```

```
mvn clean package
```

```
m55 //para ejecutar el programa
```

Entrega de la tarea

Las siguientes acciones tienen que ejecutarse tanto a nivel del repositorio local como el repositorio remoto de GitHub.

1. Crear el archivo .gitignore bajo el folder tec-poo e incluir todos los contenidos que se desean ignorar. (ver ejemplo en la sección de Notas). Incluir en el repositorio.
2. Hacer un nuevo branch llamado "dev"
3. Crear un nuevo folder bajo tec-poo con nombre "tareass/battleship"
4. En el folder "battleship" tiene que ir el proyecto Maven.
5. Cuando considere tener lista la tarea:
 - a. Cambie la versión del proyecto de 1.0.0-SNAPSHOT a 1.0.0
 - b. Haga un tag con nombre "tarea-1".
6. Haga "git merge" del código de "dev" con "main"
7. Borre el branch "dev"
8. **Solamente se revisará el código de la tarea que está bajo el tag de "tarea-1"**. Si su tag se llama "v1, 1.0.0, Tarea1, tarea_1, final, version_final", la tarea no será evaluada.
9. No tiene que existir el folder "target" bajo el proyecto en los contenidos del tag.

Notas

1. Leer argumentos desde la terminal:

```
import java.util.Scanner;

public static void main(String.. args) {
    System.out.print("enter a number" + ": ");
    Scanner scanner = new Scanner(System.in);
    int userGuess = scanner.nextInt();

    System.exit(0);
}
```

2. Contenido del archivo .gitignore

```
# Created by
https://www.gitignore.io/api/java,maven,eclipse,intellij+all,visualstudiocode
# Edit at
https://www.gitignore.io/?templates=java,maven,eclipse,intellij+all,visualstudiocode
```

```
.metadata
bin/
tmp/
*.tmp
*.bak
*.swp
*~.nib
local.properties
.settings/
.loadpath
.recommenders
```

```
# External tool builders
.externalToolBuilders/
```

```
# Locally stored "Eclipse launch configurations"
*.launch
```

```
# PyDev specific (Python IDE for Eclipse)
*.pydevproject
```

```
# CDT-specific (C/C++ Development Tooling)
.cproject
```

```
# CDT- autotools
.autotools
```

```
# Java annotation processor (APT)
.factorypath
```

```
# PDT-specific (PHP Development Tools)
.buildpath
```

```
# sbteclipse plugin
.target
```

```
# Tern plugin
.tern-project
```

```
# TeXlipse plugin
```

```
.texlipse

# STS (Spring Tool Suite)
.springBeans

# Code Recommenders
.recommenders/

# Annotation Processing
.apt_generated/

# Scala IDE specific (Scala & Java development for Eclipse)
.cache-main
.scala_dependencies
.worksheet

# Eclipse Core
.project

# JDT-specific (Eclipse Java Development Tools)
.classpath

# Annotation Processing
.apt_generated

.sts4-cache/

# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm,
CLion, Android Studio and WebStorm
# Reference:
https://intellij-support.jetbrains.com/hc/en-us/articles/206544839

# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf

# Generated files
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
.idea/**/dataSources.ids
.idea/**/dataSources.local.xml
.idea/**/sqlDataSources.xml
```



```
.idea/**/dynamic.xml
.idea/**/uiDesigner.xml
.idea/**/dbnavigator.xml

# Gradle
.idea/**/gradle.xml
.idea/**/libraries

# Gradle and Maven with auto-import
# When using Gradle or Maven with auto-import, you should exclude module
files,
# since they will be recreated, and may cause churn. Uncomment if using
# auto-import.
# .idea/modules.xml
# .idea/*.iml
# .idea/modules
# *.iml
# *.ipr

# CMake
cmake-build-*/

# Mongo Explorer plugin
.idea/**/mongoSettings.xml

# File-based project format
*.iws

# IntelliJ
out/

# mpeltonen/sbt-idea plugin
.idea_modules/

# JIRA plugin
atlassian-ide-plugin.xml

# Cursive Clojure plugin
.idea/replstate.xml

# Crashlytics plugin (for Android Studio and IntelliJ)
com_crashlytics_export_strings.xml
crashlytics.properties
crashlytics-build.properties
fabric.properties

# Editor-based Rest Client
```

.idea/httpRequests

Android studio 3.1+ serialized cache file
.idea/caches/build_file_checksums.ser

Ignores the whole .idea folder and all .iml files
See <https://github.com/joeblau/gitignore.io/issues/186> and
<https://github.com/joeblau/gitignore.io/issues/360>

.idea/

Reason:
<https://github.com/joeblau/gitignore.io/issues/186#issuecomment-249601023>

*.iml
modules.xml
.idea/misc.xml
*.ipr

Sonarlint plugin
.idea/sonarlint

Compiled class file
*.class

Log file
*.log

BlueJ files
*.ctxt

Mobile Tools for Java (J2ME)
.mtj.tmp/

Package Files #
*.jar
*.war
*.nar
*.ear
*.zip
*.tar.gz
*.rar

virtual machine crash logs, see
http://www.java.com/en/download/help/error_hotspot.xml
hs_err_pid*

```
target/  
pom.xml.tag  
pom.xml.releaseBackup  
pom.xml.versionsBackup  
pom.xml.next  
release.properties  
dependency-reduced-pom.xml  
buildNumber.properties  
.mvn/timing.properties  
.mvn/wrapper/maven-wrapper.jar  
.flattened-pom.xml
```

```
.vscode/*  
!.vscode/settings.json  
!.vscode/tasks.json  
!.vscode/launch.json  
!.vscode/extensions.json
```

```
# Ignore all local history of files  
.history
```

```
# End of  
https://www.gitignore.io/api/java,maven,eclipse,intellij+all,visualstudioc  
ode
```