Interactive Game Design with Greenfoot—YEAR 2
Greenfoot Single-player Interactive Game

Level: High School. Open to all competitors. **Required** for competitors who participated in the 2015 Competition.

Type of Contest: Team
Composition of Team: 2 – 4 students per team
Number of Teams: 10 entries per Center

The year 2 MESA Virtual Computer Science Competition is designed to extend and advance the knowledge of the Greenfoot programming environment and the Java programming language. The year 2 competition will require students to incorporate some advance concepts of Greenfoot programming. Students participating in the year 2 competition will create a project at their home school sites and, subsequently, submit it to the MESA California Statewide Office for judging. The MESA Statewide Office may determine at its discretion that the competition is to be a timed, proctored competition at the school sites.

**OVERVIEW**

Develop an educational and creative interactive game using Greenfoot in accordance to the specifications set forth in this document. Entries will be submitted to a statewide URL for judging by 5:00 p.m. **February 22, 2016**.

All entries must be designed and created by California MESA students. Software packages and their sources are as follows:

<table>
<thead>
<tr>
<th>Software Package</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greenfoot 2.3 or above (2.4.2 now available)</td>
<td><a href="http://www.greenfoot.org/download">www.greenfoot.org/download</a></td>
</tr>
</tbody>
</table>

Also, it is *highly recommended* that each school obtain at least one copy of the official Greenfoot instructional book:

The first edition is suitable; used copies can be purchased from Amazon. The second edition is now available but is expensive.

Judges reserve the right to contact contestants prior/during judging for verification of student work. Student work should be verified by the teacher/mentor prior to submission; this might help to eliminate copying and missing content.

**GAME DETAILS**

1. **Skill Levels**: Each game must have three levels which increase in difficulty. Difficulty is defined as an increase in skill level.
2. **Time Limit**: Each level should take a minimum of one minute to play.
3. **Start and Return to Menu**: All levels must have the following buttons that will function while the level is executing:
   1. Restart
   2. Return to main menu
4. **Timer**: The game must include a timer.
5. **Scoreboard**: The game must include a scoreboard.
6. **Sound**: The game must include sound effects.
7. **Color**: Use of the color library for at least one class.
8. **Anatomical Animation**: The game must include animations. Animation characters can be obtained from the Greenfoot images library or imported from other sources. They must have anatomical functions.
9. **Data Input**: The game must prompt the user to input data in the form of text or numbers at least once per level.
10. **Instructions**: The game must contain clear and informative in-game instructions relevant to the targeted age and skill level.
11. **Starting Configuration**: Game pieces should be initially configured automatically.
12. **Disappearing**: Some objects should be programmed to **disappear** either randomly or based on a game event.
13. **Appearing**: Some objects should be programmed to **appear** either randomly or based on a game event.
14. **Wrapping**: Some object should wrap the world. This means that if they exit one side of the world moving in a particular direction, they should return to the world moving in the same direction.
15. **Rebounding**: Some object(s) should rebound off of a side of the world or a fixture within the world using proper geometric principles (Example: If an moving object contacts a fixed object at a 30° angle, it should rebound at a 30° angle).
16. **Constructors**: The game should employ the use of one or more constructors.
17. **Random Motion**: One or more objects should contain random motion.
18. **Actor 1-Intermediate Actor-Actor 2 Contact**: An event driven object should be generated by one actor and delivered to a second actor causing some resultant action. The following are examples:
   a. Upon key entry, a tank produces an artillery shell that intersects with building and causes it to disappear or explode (incrementing the score).
   b. Upon key entry, a rescue plane produces a care package and drops it to a hurricane victim resulting in some game response (incrementing the score).

19. **While or For Loop**: The game should include either a while loop or a for loop.

20. **If-then Decision**: The game should include an if-then loop or an if-then-else loop.

Games can be built from one of the Greenfoot templates available on the MESA website. (Only these templates may be used.)

**Documentation:**
Students should produce concise in-code comments explaining the function of each line, or related series of lines, of code. Documentation should serve two purposes:
   a. The comments should be sufficient for a reader with a basic understanding of Greenfoot to understand the function of the code.
   b. The comments should also serve to identify for the judges that a specific game element (see “Rules”) has been utilized.

**JUDGING**
Games will be judged in the following categories (see attached rubric):
   1. Game Components
   2. Technology Fluency
   3. In-code Comments
   4. Creative Design
   5. Questions

It benefits contestants to make it as easy as possible for judges to follow your code and comments.

**SCORING**
Judges will score games individually. The average score of all judges’ results will be totaled to determine the score for each team.

**AWARDS**
Medals will be awarded for first, second, and third place teams overall and by region. Center Directors will be notified on **March 8, 2016** of local and regional winners.
## 2016 MESA Interactive Game Design Competition with Greenfoot Year 2 Rubric

<table>
<thead>
<tr>
<th>Points Possible</th>
<th>Advanced</th>
<th>Developing</th>
<th>Emerging</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Game Components</strong></td>
<td>25</td>
<td>One point will be awarded for each of twenty (20) game elements included in the game (see “Rules”). An additional five (5) points can be awarded for the three levels of increasing difficulty, animation, and sound effects.</td>
<td>One point will be awarded for each of twenty (20) game elements included in the game (see “Rules”). Fifteen or fewer of the game elements and two levels of increasing difficulty.</td>
</tr>
<tr>
<td><strong>Technology Fluency</strong></td>
<td>25</td>
<td>The game works as designed with no errors due to programming or design. Game is particularly well organized, logical, and debugged. Good programming structure is used. Methods are written below the act method and then called when needed.</td>
<td>The game works completely from beginning to end but may have minor flaws in the way it flows. Good programming structure is sometimes used. Methods should not be written in the act method. Rather methods are called to the act method when needed.</td>
</tr>
<tr>
<td><strong>In-code Comments</strong></td>
<td>25</td>
<td>Comments explain the function of individual lines of code or a related series of lines of code. The explanations should be understandable to a beginning Greenfoot user. Comments also completely identify the completion of any of the 20 game elements for judges’ reference. Minimum 90 percent complete.</td>
<td>Comments mostly explain (greater than or equal to 50 percent) the individual lines or a related series of lines. Comments do not completely identify the game elements used.</td>
</tr>
<tr>
<td><strong>Creative Design</strong></td>
<td>15</td>
<td>Game concept is highly creative. More than a shooter or chase game. Elements of complex game responses to events within the game or user driven events are in evidence.</td>
<td>Game concept shows some creativity. This would include shooter or chase games with embellishments. A classic Pacman game would be an example.</td>
</tr>
<tr>
<td><strong>Answers to Questions</strong></td>
<td>10</td>
<td>One point for each correct answer.</td>
<td>One point for each correct answer.</td>
</tr>
</tbody>
</table>
Year 2: Questions about Java

1. What is a Java Virtual Machine (JVM)?

2. What is a Java class?

3. What is an object? How are objects and classes related?

4. Can different instances of an object of the same class have different characteristics? Explain.

5. What is a Constructor?

6. What is a “call” and how are calls used in the act method in Greenfoot?

7. Name the seven Greenfoot classes and what function they perform.

8. What does Java API stand for and where can Java API be accessed from Greenfoot?

9. Name three advantages of Java.

10. Who developed Java and for what company? What company now owns Java?