

# Apply Basic Math Skills to Create Scoring Schemes in a Simple Game

“PLEASE EXCUSE My Dear Aunt Sally.” What does Aunt Sally have to do with video gaming, and what did she do that needs excusing? If you learn the mathematical order of operations, you just may create a knockout scoring scheme.



## Objectives:



1. Describe the mathematical order of operations.
2. Define terms commonly used in game scoring schemes.
3. Write mathematical equations to satisfy scoring requirements for a simple game.

## Key Terms:



arithmetic operators  
decrement  
design document  
health  
increment  
lives  
order of operations  
power  
score

## Understanding the Order of Operations and Scoring

Most games have winners and losers and some way of distinguishing the two. Typically, some type of score is maintained. If you have played games and kept score, then you have

applied basic math principles to maintain scores. Board games (e.g., Checkers, Scrabble®, Dominoes, Battleship®, and Trivial Pursuit®) and Internet-based games use scoring schemes. The same basic math principles apply to video gaming.



FIGURE 1. Even a simple board game can use a type of scoring scheme.

## MATHEMATICAL ORDER OF OPERATIONS

**Order of operations** is the mathematical rule that states that arithmetic equations are evaluated from left to right in a particular order. Some **arithmetic operators** (symbols that indicate the type of calculation to occur) take higher precedence than others. The saying “Please Excuse My Dear Aunt Sally” is often used to help students remember the order of operations. Without the order of operations in place, incorrect answers would result.

### Rules

Certain rules must be followed in a very specific order with no exceptions:

- ◆ Rule 1—It is necessary to perform any calculations inside parenthesis (P).
- ◆ Rule 2—It is essential to perform any exponential (E) calculations, working from left to right.
- ◆ Rule 3—It is necessary to perform multiplication (M) and division (D) calculations, working from left to right.
- ◆ Rule 4—It is essential to perform addition (A) and subtraction (S) calculations, working from left to right.

## COMMON ARITHMETIC OPERATORS USED IN PROGRAM CODE

Many arithmetic operators are used in program code.

- ◆ The plus sign (+) is used for addition.
- ◆ The minus sign (–) is used for subtraction.
- ◆ The asterisk (\*) is used for multiplication.
- ◆ The forward slash (/) is used for division.
- ◆ The percent sign (%) is known as the modulus operator. It is used by some programming codes, such as C++, to return the remainder of an integer division.

# Terms Commonly Used in Game Scoring Schemes

Many terms are used in gaming. These basic terms allow gamers to communicate clearly and quickly with each other. This is especially important when programming as a team.

## DECREMENT

---

**Decrement** is to take away or to reduce. In a war game, each time a player destroys an enemy tank, the number of enemy tanks the player has to battle is decreased or decremented by one.

## HEALTH

---

**Health** is the level of strength or power that the player has available to use in game challenges. For instance, in a boxing game, the player may begin with a high level of health. However, after time goes by and a few punches are taken, health will decrease. If the player is knocked out, health drops to zero.

## INCREMENT

---

An **increment** is an addition or increase. In a hunting game, the player score is increased each time a game animal is shot.

## LIVES

---

**Lives** are the number of chances a player has to accomplish the game objectives. Games commonly allow a player to earn extra lives by reaching a particular score or by successfully accomplishing a difficult challenge. “Lives” and “health” sometimes may be interchanged, depending upon the design and terms used in a particular game. Many games depict the number of lives or the level of health with red heart icons.

## POWER

---

**Power** is a term with at least three different meanings in gaming: health, number of weapons, or advantage. Power can be compared to health because it may indicate the amount of strength the player has available to use in a game challenge. Power may indicate the amount of ammunition or number of weapons a player has available to use in a game, or it may describe an advantage or strength that one player or character has over another (e.g., the ability to fly).

## SCORE

**Score** is a term that typically takes on two meanings in gaming: points earned and the act of earning points. Scoring (verb) points increases the player's score (noun). Score, used as a noun, is the record kept to indicate the number of points a player has earned in the game. Score, used as a verb, is the act of earning points or accomplishing a task.

## Mathematical Equations

To write the appropriate mathematical equations, you must first thoroughly understand how the scoring is designed. Once you better understand its organization and functioning, you can code the equations.

## DESIGN DOCUMENT

It is necessary to read the game **design document**, which is a written item that details



FIGURE 2. Games that contain scores use basic math principles.



## UNDER INVESTIGATION...

### LAB CONNECTION: Reinforcing Scoring Concepts in Gaming

Read the following design document and interpret the scoring requirements. Then write the appropriate equations to meet those requirements. Begin by reading each situation. Then write a math problem to calculate the score and the lives at that point in the game. In each game, Number 1 assumes the game has just begun; so there are three lives available, and the score is zero. Use the results of situation Number 1 to calculate the “score” and “lives” results from situation Number 2 and so on. Determine the number of lives available and the score at the end of each game.

The objective of this snowboarding game is to snowboard as many mountain ski runs as possible before running out of lives. Each time a run is successfully “boarded,” the player earns 1,000 points and moves to a higher run. While snowboarding, the player must attempt to jump several obstacles on the way down each run. The player has three chances to make it down as many mountain runs as possible. If the player runs into a tree, the player loses a life (chance). For every obstacle successfully jumped, the player earns 50 points. If the player attempts to jump an obstacle and falls, the player loses a life. If the player misses a ski lift, the player takes the next lift but loses 50 points. If the player makes it to the bottom of the highest run while

maintaining all three lives, the player score is doubled after points are awarded for reaching the bottom. The player starts each game with three lives and a score of zero.

**Game 1:**

1. The player successfully jumps an obstacle on the first run.
    - Score =  $0 + 50 = 50$
    - Lives = 3
  2. The player successfully jumps another obstacle.
    - Score =
    - Lives =
  3. The player attempts to jump an obstacle but falls.
    - Score =
    - Lives =
  4. The player successfully jumps an obstacle.
    - Score =
    - Lives =
  5. The player reaches the bottom of the first run.
    - Score =
    - Lives =
  6. The player successfully jumps an obstacle on the second run.
    - Score =
    - Lives =
  7. The player runs into a tree.
    - Score =
    - Lives =
  8. The player successfully jumps an obstacle.
    - Score =
    - Lives =
  9. The player reaches the bottom of the second run.
    - Score =
    - Lives =
  10. The player misses the ski lift.
    - Score =
    - Lives =
  11. The player successfully jumps four obstacles on the way to the bottom of the third run.
    - Score =
    - Lives =
  12. The player reaches the bottom of the third run.
    - Score =
    - Lives =
  13. The player successfully jumps an obstacle on the fourth run.
    - Score =
    - Lives =
  14. The player runs into a tree.
    - Score =
    - Lives =
- Final score =  
Lives =

every aspect of the game to be created (e.g., game play, characters, and the environment). This document is used to guide the team creating the game through the development process. It is essential to simulate the game play. If possible, programmers create a non-electronic version of the game and play it several times to boost their understanding of how scoring should work.

## CODING EQUATIONS

Coding of equations can begin when the design document is thoroughly understood. Programmers follow the order of operations to choose the appropriate arithmetic operators. The proper language syntax makes it all come together.

### Summary:



Basic math operations and video game scoring schemes are closely related. Most games have winners and losers and some way of distinguishing the two. The order of operations is the mathematical rule that states that arithmetic equations are evaluated from left to right in a particular order. This rule is foundational when calculating scores.

Many arithmetic operators are used in program code. These need to be carefully memorized. Also, there are many terms used in gaming. These basic terms allow gamers to communicate clearly and quickly with each other. To write the appropriate mathematical equations, you must first thoroughly understand how the scoring is designed. Once you better understand its organization and functioning, you can code the necessary equations.



**FIGURE 3.** Even a snowboarding game requires basic math if you want to keep score.

### Checking Your Knowledge:



1. Briefly explain the difference between the gaming terms “increment” and “decrement.”
2. Describe the three common definitions of power used in gaming.
3. Define and explain the order of operations.
4. Define the term “score” as a noun and as a verb.
5. An advantage or ability that one player or character has over another is known as what?

## Expanding Your Knowledge:



For further practice with scoring schemes, try another version of the snowboarding game (introduced above in the sidebar). In this game, the score is updated only when the player reaches the bottom of the run. Write equations accordingly.

1. The player successfully jumps all 10 obstacles on the way to the bottom of the first run.
  - Score =
  - Lives =
2. The player successfully jumps all 12 obstacles on the way to the bottom of the second run.
  - Score =
  - Lives =
3. The player successfully jumps only four obstacles but makes it to the bottom of the third run. The player had attempted only five obstacles.
  - Score =
  - Lives =
4. The player runs into a tree but manages to successfully jump all 20 obstacles on the fourth run.
  - Score =
  - Lives =
5. The player successfully jumps all 25 obstacles on the way to the bottom of the fifth and final run.
  - Score =
  - Lives =

Final score =

Lives =

## Web Links:



### Game Programmer

[http://en.wikipedia.org/wiki/Game\\_programmer#Game\\_engine\\_programmer](http://en.wikipedia.org/wiki/Game_programmer#Game_engine_programmer)

### Physics in Video Games

[http://wiki.answers.com/Q/How\\_physics\\_relates\\_to\\_video\\_games](http://wiki.answers.com/Q/How_physics_relates_to_video_games)

### Mathematics in Videogames

[http://www.gamasutra.com/view/feature/3197/mathematics\\_in\\_videogames.php](http://www.gamasutra.com/view/feature/3197/mathematics_in_videogames.php)