

TROY GILBERT

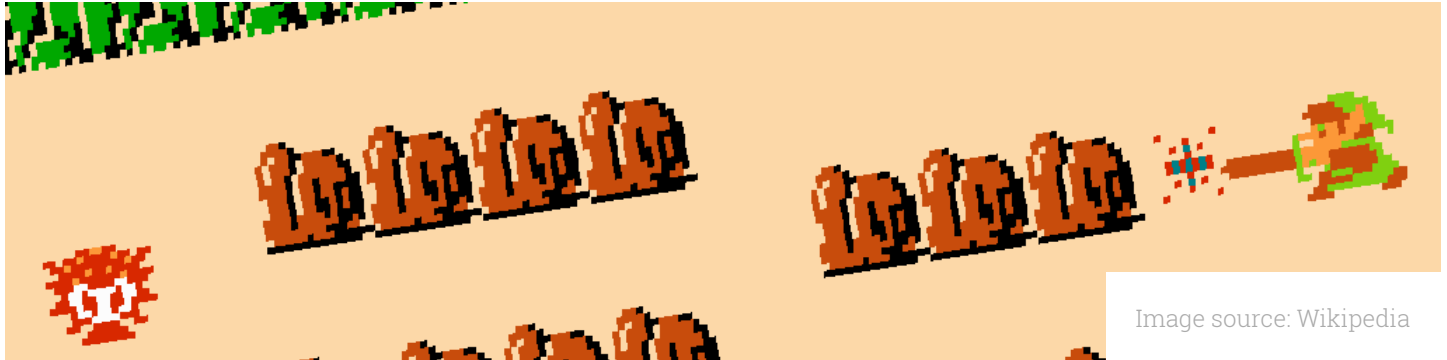


Image source: Wikipedia

[HOME](#) / [DECONSTRUCTING ZELDA](#) /

Movement Mechanics

Let's start our deconstruction with the most basic feature of the player's character, Link: his movement. Link's movement displays a surprising amount of subtlety. If you play *The Legend of Zelda* for a while, you'll notice that Link doesn't ever get hung up on obstacles due to unexpected collisions, nor do you ever just miss a bad guy by a pixel or two when attacking him. Yet at the same time, you'll notice that Link moves fluidly through the gameworld; even though the environment is tile-based, you can tap the directional pad and link will move a single pixel in that direction. If you're moving to the left and decide to go up, down or right, Link instantly moves in that other direction when you change directions on the gamepad.

What's happening here is a very neat trick. While Link can move a single pixel at a time, in any direction, the longer he continuously moves in any direction the more he gravitates toward aligning himself with the underlying grid of the screen. The tile grid for *LoZ* is 16 tiles wide by 14 tiles high (including 3 tiles for the status display at the top of the screen). Each tile is 16×16 pixels. Link operates on a half-tile grid, though (32×28 tiles, 8×8 pixels each). As Link moves, if he's not currently aligned with the half-tile grid, he is adjusted, one pixel at a time, toward the closest

In case that description doesn't make sense, here's an exercise to help demonstrate the technique:

Assume Link is basically in the center of the screen, lined-up with a tile. In other words, Link occupies a whole tile, just as he would in a strictly tile-based game (like a traditional boardgame or wargame). If you tap on the directional pad to the left, Link will move one pixel to the left. If you tap again, he'll move another pixel to the left. This is pixel-fine control, which makes Link's movement feel fluid.

Now, if you hold the directional pad to the left, Link will move continuously. As soon as you let up, he stops. If you hold left then immediately reverse to hold right, he instantly changes direction (as he does for up and down as well). This is basically lag-free, which makes Link's movement feel responsive.

Okay, return back to Link sitting squarely on a tile. If you tap left twice he'll be two pixels over the left edge of his starting tile. Now press up four times. You'll notice that after each press Link will move up one pixel. But, he'll also move to the right one pixel, either the first two times you pressed up or the first and third times you pressed up (I don't have the game in front of me and I can't remember the rate at which it corrects). So, even though you never press right on the gamepad, Link has now returned to the same horizontal position as he was when he started (before you moved him left two pixels). This is the built-in correction.

What does this correction do for you?

The correction prevents the subtle but annoying problem wherein the player would "snag" on the corners of objects that he anticipated passing by. The more the player moves continuously the more aligned Link becomes, which has the same effect as speed-sensitive steering in a car.

The correction also has benefits for attacking as well. When Link attacks (with his sword, for example), the "kill zone" lies in the tile(s) immediately along his facing direction. Since enemies align along a similar half-grid as Link, the correction serves to line up Link with his enemies (as opposed to missing the enemy by a few frustrating pixels).

<http://troygilbert.com/deconstructing-zelda/movement-mechanics/>

Go

MAR

JUN

OCT

◀ 24 ▶

2014

2015

2016

[47 captures](#)

24 Feb 2015 - 17 Nov 2019

▼ About this capture

gameworld is still tile-based, player (and enemy/NPC) movement is not strictly aligned on any grid. We get the same corrections though, but using different mechanisms.

Movement is corrected by “rounding” the corners of most things Link can collide with. Thus, when he encounters them the “physics” of the game deflects him at a 45-degree angle along his direction of movement (e.g., if he was moving north/up and encountered a corner he would move northwest/east until he passed the corner).

Attacking is corrected in a similarly interesting way. Instead of Link’s sword having a killzone along a straight line down his facing direction, his sword “swipes” in a large arc, an arc that happens to be pretty close to full two “tiles” in the gameworld (one directly in front and half above and half below). This allows the player to attack while slightly mis-aligned with an enemy and still land a blow.

The result for both actions in both games is the same: the player’s desire is successfully expressed in the gameworld, regardless of the potentially pedantic ways of the computer.

[Share on Twitter](#)[Share on Facebook](#)[Share on Google+](#)

[I made Dumbhack.](#) **[I made Puzzles by Thumb.](#)** **[My tumblr of inspiration.](#)**

[Follow me on Twitter.](#)