Extended Abstract
The 23rd Thailand-Japan Conference on
Discrete and Computational Geometry, Graphs, and Games
Department of Mathematics, Chiang Mai University
TJCDCG³ 2020+1 Chiang Mai, Thailand

The Legend of Zelda: The Complexity of Mechanics

Jeffrey Bosboom[1], Josh Brunner[1], Michael Coulombe[1], Erik D. Demaine[1],
Dylan H. Hendrickson[1], Jayson Lynch[2], Elle Najt[3]

[1]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[2]Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada
[3]Department of Mathematics, University of Wisconsin, Madison, WI, USA
jbosboom@mit.edu (J. Bosboom), brunnerj@mit.edu (J. Brunner), mcoulomb@mit.edu (M. Coulomb), edemaine@mit.edu (E. Demaine), dylanhen@mit.edu (D. Hendrickson), jayson.lynch@waterloo.ca (J. Lynch), lnajt@math.wisc.edu (L. Najt)

Abstract

We analyze some of the many game mechanics available to Link in the classic Legend of Zelda series of video games. In each case, we prove that the generalized game with that mechanic is polynomial, NP-complete, NP-hard and in PSPACE, or PSPACE-complete. In the process we give an overview of many of the hardness proof techniques developed for video games over the past decade: the motion-planning-through-gadgets framework, the planar doors framework, the doors-and-buttons framework, the "Nintendo" platform game / SAT framework, and the collectible tokens and toll roads / Hamiltonicity framework.

Keywords: video games, hardness, NP, PSPACE.

2010 MSC: Primary 68Q17; Secondary 68Q25.

1 Introduction

The Legend of Zelda* action–adventure video game series consists of 19 main games developed by Nintendo (sometimes jointly with Capcom), starting with the famous 1986 original which sold over 6.5 million copies [6], and most recently with Breath of the Wild which was a launch title for Nintendo Switch (and is arguably what made the Switch an early success). In each game, the elf protagonist Link explores a world with enemies and obstacles that can be overcome only by specific collectible items and abilities. Starting with nothing, Link must successively search for items that unlock new areas with further items, until he reaches and defeats a final boss enemy, Ganon.

Across the 35-year history of the series, many different mechanics have been introduced, leading to a varied landscape of computational complexity problems to study: what is the difficulty of completing a generalized Zelda game with specific sets of items, abilities, and obstacles? Reviewing the two Zelda wikis† and playing the games ourselves, we have identified over 80 unique items with unique mechanics, throughout the 19 games in the Zelda franchise.

In tribute to the fun and challenge of the Zelda series, we propose a long-term undertaking where the video-game-complexity community throughly catalogs these mechanics and analyzes which combinations lead to polynomial vs. NP-hard computational problems. Toward this goal, we analyze in this paper the complexity of several new combinations of various items. Table 1 summarizes our results, along with previously known results about Legend of Zelda [1, 3].

Our new results also serve to highlight different techniques for proving polynomial/NP algorithms, NP-hardness, and PSPACE-hardness of video games involving the control of a single agent/avatar. For algorithms, we see the powerful approach of dynamic programming combined with the technique of shortcutting. For hardness, one major category is Hamiltonian Path inspired reductions, often simplified

†https://zelda.fandom.com/wiki/Category:Items and https://www.zeldadungeon.net/wiki/Category:Items

Game Mechanics	Games with Mechanics	Result	Ref
Hookshot, Pots, Pits	ALTTP, LA, PH, ALBW	$\in$ P	**new**
Hookshot, Pots, Pits, Keys	ALTTP, LA, PH, ALBW	NP-c.	**new**
Switch Hook, Diamond Blocks, Pits	OoA	$\in$ P	**new**
Crystal Switches, Raised Barriers	ALTTP, LA, OoA, OoS, PH, ALBW	$\in$ P	**new**
Roc's Feather, Pegasus Seeds	OoA, OoS, MM	NP-c.	**new**
Bombs, Renewing Cracked Walls	OoT, MM, OoA, OoS, TMC, ST	NP-c.	**new**
Ice Arrows, Water	MM	NP-c.	**new**
Fairies, Bottles, Unavoidable Damage Region	ALTTP, LA, OoT, MM, OoA, OoS, FS, TWW, FSA, TMC, TP, PH, ST, SS, ALBW, BotW	NP-c.	**new**
Magic Armor, Unavoidable Damage Region	ALTTP, OoT, TWW, TP	NP-c.	**new**
Bow or Bombs, and Crystal Switches for Raised Barriers	ALTTP, LA, OoA, OoS, PH	NP-c.	**new**
Colored-tile floor puzzles	LA, OoA, TMC	NP-c	**new**
Kodongos, low walls, sword	ALTTP	NP-hard	**new**
Buzz Blobs, Master Sword	ALTTP, LA, OoA, OoS, TMC, ALBW, TFH	NP-hard	**new**
Decayed Guardians, Bombs	BotW	NP-hard	**new**
Magnetic gloves, metal orbs, ledges, jump platforms	OoS	PSPACE-c.	**new**
Cane of Pacci, ground holes, ledges, tunnels	TMC	FPT in duration PSPACE-c.	**new** **new**
Magnesis Rune, metal platforms	BotW	PSPACE-c.	**new**
Statues, Pressure Plates, Doors	ALTTP, OoT, MM, OoA, OoS, FS, TWW, FSA, TMC, TP, PH, ST, SS, ALBW	PSPACE-c.	**new**
Ancient Orbs, Pedestals, Doors	BotW	PSPACE-c.	**new**
Minecarts	OoA, OoS, TMC	PSPACE-c.	**new**
Once-Pushable Blocks	Zelda I, LA, OoA, OoS, TMC	NP-c.	[1]
Once-Pushable/Pullable Blocks, hookshot, chests, pits, tunnels	ALTTP, LA, OoT, MM, TWW, ALBW	NP-c.	[1]
Keys, Doors, Ledges	AoL, ALTTP, LA, OoT, MM, OoA, OoS, FS, TWW, TMC, TP, PH, ST, SS, ALBW	NP-c.	[1]
Once-Pushable Blocks, Ice	OoT, MM, OoS, TMC, TP, ST	PSPACE-c.	[1]
Buttons, Doors, Teleporters, Pits, Crystal Switches	ALTTP, ALBW	PSPACE-c.	[1]
Spinners	OoA, OoS	PSPACE-c.	[3]

Table 1: Summary of new and past results about complexity of various Zelda mechanics.

with Viglietta's Metatheorem 2 [7] concerning collectible items and toll roads. Next is the Nintendo-style SAT reduction from [1] which later acted as inspiration for the door-opening gadgets in [4]. For PSPACE-hardness, we use the door-and-button framework of Forišek [5] and Viglietta [7]. Finally, we use the door gadget from [1] which, along with the other previous work, inspired the gadgets framework for the complexity of motion-planning problems [2–4] which we also use. As a secondary goal, we hope that this paper offers a nice sampling of proof techniques showing the hardness infrastructure that have been built up in recent years.

References

[1] G. Aloupis, E. D. Demaine, A. Guo, and G. Viglietta. Classic Nintendo games are (computationally) hard. *Theoretical Computer Science*, 586:135–160, 2015. Originally at FUN 2014.

[2] J. Ani, J. Bosboom, E. D. Demaine, Y. Diomidov, D. Hendrickson, and J. Lynch. Walking through doors is hard, even without staircases: Proving PSPACE-hardness via planar assemblies of door gadgets. In *Proceedings of the 10th International Conference on Fun with Algorithms (FUN 2020)*, pages 3:1–3:23, 2020.

[3] E. D. Demaine, I. Grosof, J. Lynch, and M. Rudoy. Computational complexity of motion planning of a robot through simple gadgets. In *Proceedings of the 9th International Conference on Fun with Algorithms (FUN 2018)*, pages 18:1–18:21, 2018.

[4] E. D. Demaine, D. Hendrickson, and J. Lynch. Toward a general theory of motion planning complexity: Characterizing which gadgets make games hard. In *Proceedings of the 11th Conference on Innovations in Theoreti cal Computer Science (ITCS 2020)*, pages 62:1–62:42, 2020.

[5] M. Forišek. Computational complexity of two-dimensional platform games. In *Proceedings of the 5th International Conference on Fun with Algorithms (FUN 2010)*, pages 214–227, 2010.

[6] Video Game Sales Wiki. The Legend of Zelda. https://vgsales.fandom.com/wiki/The_Legend_of_Zelda.

[7] G. Viglietta. Gaming is a hard job, but someone has to do it! *Theory Comput. Syst.*, 54(4):595–621, 2014.