Celeste is PSPACE-hard

Lily Chung[1], Erik D. Demaine[1]

[1]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

lkdc@mit.edu (L. Chung), edemaine@mit.edu (E. Demaine)

Abstract

We investigate the complexity of the platform video game Celeste. We prove that navigating Celeste is PSPACE-hard in four different ways, corresponding to different subsets of the game mechanics.

Keywords: video games, hardness, PSPACE.

2010 MSC: 68Q17.

1 Introduction

Celeste[*] is a 2D platform video game released in 2018 by Extremely OK Games. It won the Best Independent Game and Games for Impact awards at The Game Awards 2018 and sold over a million copies [?]. In Celeste, the player controls a single character, Madeline, who must navigate various hazards along her journey. We consider the following decision problem about Celeste:

Definition 1.1 (CELESTE)**.** Given a Celeste level, is it possible for Madeline to traverse from a designated start location to a designated end location?

We amend a previous result of Ahmed et al. [?] which attempted to show that CELESTE is NP-complete. We give four proofs that CELESTE is instead PSPACE-hard, each using a different set of game mechanics. All of these proofs involve constructing a polynomial-time reduction to CELESTE from a motion-planning problem through a planar network of doors [2]. We make use of both *open-close-traverse* doors, as introduced in [1, 3, 4] and shown not to need crossovers in [2], and *self-closing doors*, as introduced in [2]. In all but one case we additionally show containment in PSPACE.

2 Main Results

Figure 1: Celeste entities mentioned in our results. From left to right: Madeline (the player, who can walk, jump, climb, and dash), Spinner (spikes), Seeker (enemies), Jellyfish (reducing Madeline's fall speed), Pufferfish (can be bounced on), Move block (can slide in one direction until collision), Barrier (obstacle for enemies), Kevin block (can be moved in all directions, and reverses their path after collision).

Theorem 2.1. CELESTE *with spinners, seekers, barriers, and move blocks is PSPACE-complete.*

---

[*]https://exok.com/games/celeste/. Celeste and its sprites are the properties of Extremely OK Games. Sprites are used here under Fair Use for the educational purpose of illustrating mathematical theorems.
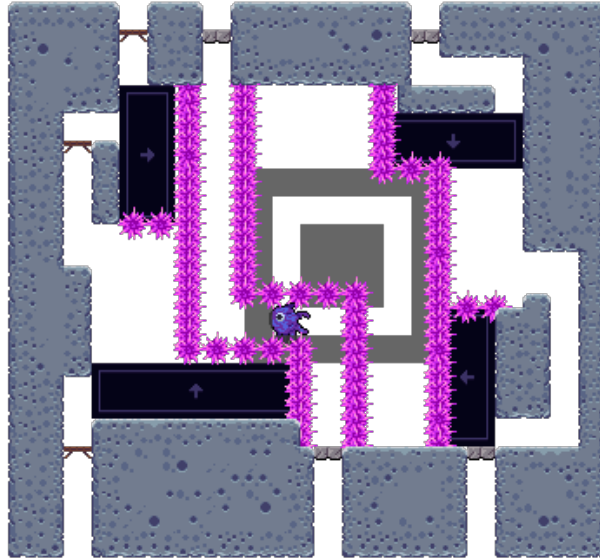
Figure 2: Part of the proof of Theorem 2.1: an open-close-traverse door constructed with spinners, a seeker, barriers, and move blocks. Currently in the "closed" state.

See Figure 2 for the main gadget in the proof of Theorem 2.1.

Theorem 2.2. CELESTE *with spinners, jellyfish, and barriers is PSPACE-complete.*

Theorem 2.3. CELESTE *with spinners and pufferfish is PSPACE-complete.*

Theorem 2.4. CELESTE *with spinners and Kevin blocks is PSPACE-hard.*

Other subsets of mechanics remain to be studied. In particular, it would be interesting to find subsets that are harder than PSPACE (e.g., undecidable), NP-complete, or (nontrivially) polynomial time.

References

[1] Greg Aloupis, Erik D. Demaine, Alan Guo, and Giovanni Viglietta, *Classic Nintendo games are (computationally) hard*, Theoretical Computer Science **586** (2015), 135–160.

[2] Joshua Ani, Jeffrey Bosboom, Erik D. Demaine, Yevhenii Diomidov, Dylan Hendrickson, and Jayson Lynch, *Walking through doors is hard, even without staircases: Proving PSPACE-hardness via planar assemblies of door gadgets*, Proceedings of the 10th International Conference on Fun with Algorithms (FUN 2020) (La Maddalena, Italy), September 2020, pp. 3:1–3:23.

[3] Giovanni Viglietta, *Gaming is a hard job, but someone has to do it!*, Theory of Computing Systems **54** (2014), no. 4, 595–621.

[4] _____ , *Lemmings is PSPACE-complete*, Theoretical Computer Science **586** (2015), 120–134.

[†]https://github.com/CelestialCartographers/Ahorn