Theory of Recursed

Jimin Park

September 8, 2017

In this article, various aspects of Recursed are described, including mechanics, gadgets, and theorical analysis of it.

This article is aimed towards people trying to make a custom Recursed puzzle, or people who are interested in theorical side of Recursed. If the reader of this article haven't see the ending of Recursed, then one is advised against continue reading this article, as this article contains many strategies of Recursed where discovering them is a big fun part of the game.

Contents

1 Introduction

Recursed is a puzzle game released in 2016, developed by Portponky. The main objective of a Recursed puzzle is to reach to a destination using various game elements.

1.1 Game Elements

1.1.1 Room

A *room* is a rectangular area with tiles and various items in it. The size of a room is 20×15, where there are 20 tiles horizontally and 15 tiles vertically.

There can be items (which will be described later) in a room, and there is no (logical) limit how many items can be in a room.

A player can't go outside of a room. The left, right, or top edge of a room is blocked by an invisible barrier, which prevents the player from escaping. However, any item can go past the invisible barrier (by throwing or using a fan). Items gone beyond the left or the right edge will be removed[1], but items went beyond the top edge will be there, which can be retrieved after removing the fan.

When the player reaches the bottom edge of the room, the player will bounce. However, this behavior is *not* used to solve any of the puzzles in the game, as it is impossible[2] to do.

1.1.2 Crystal and Diamond

A *crystal* is a fixed item in a room, marking a destination of a puzzle. If a player collects a crystal, the player will exit the puzzle and it will be marked as crystal-solved.

A *diamond* appears in a paradox room, which can be entered after a player invokes a paradox. When a diamond is collected, the player will exit the puzzle and it will be marked as diamond-solved.

After invoking a paradox, a crystal will be unable to be collected. When a player attemps to collect a crystal after invoking a paradox, the crystal will simply disappear when touched.[3]

[1] There was a bug that items gone beyond the left or the right edges were not removed, but it will be fixed in a future update.

[2] Actually, it is possible to touch the bottom edge in the puzzle "Dump".

[3] This behavior was introduced by the developer to prevent players from getting crystal easily with fans obtained from a paradox room.

1.1.3 Throw

1.1.4 Block

1.1.5 Keys and Door

1.1.6 Chest

1.1.7 Green Objects

1.1.8 Water and Acid

1.1.9 Paradox

A *paradox* can be happen with green chests.

1.1.10 Jar

1.1.11 Fan

1.1.12 Oobleck

1.1.13 Glitch

1.1.14 Other Items

1.2 Tiles

1.2.1 Empty

1.2.2 Solid

1.2.3 Ledge

1.2.4 Water and Acid

2 Game Mechanisms

2.1 Ways of Invoking a Paradox

3 Complexity of Recursed

There haven't much research on algorithms to solve a Recursed map, or how complex such an algorithm would be.

First, let's introduce a proper decision problem based on the puzzle of Recursed.

Definition 3.1. The *Standard Recursed Decision Problem*, or the *Standard RDP* for short, is the problem of deteriming whether a puzzle P is solvable, where P is given as list of rooms $\{r_1, r_2, \cdots, r_n\}$. Here, the size of a room r_i is fixed to be 20×15, and the number of initial items in a room r_i is bounded by a fixed constant. (To prevent the constant from increasing arbitrarily, let's set the constant to 300, the number of tiles in a room.)

Therefore, the length of the input for the Standard RDP can be represented in $\tilde{\Theta}(n)$ in terms of the amount of rooms n, for any proper encoding (where additional $\log n$ terms might appear due to specify room #).

There are several interesting sub-problems of the Standard RDP.

Definition 3.2. The *Pure Recursed Decision Problem*, or the *Pure RDP*, is a subset of the Standard RDP where none of objects in a room is green.

In any instance of the Pure RDP, any chest is a pure function. So, for any item and any chest, going into the chest with the item will result in a set of finite, fixed actions.

Definition 3.3. The *Simple Recursed Decision Problem*, or the *Simple RDP*, is a subset of the Pure RDP where no fissures present in any of the rooms.

It will be shown that the maximum number of items in a room in any step of a good solution can be bounded by a polynomial function of n.

Lemma 3.1. *There is a fixed polynomial function $p(n)$, such that in any instance P of the Simple RDP, if there's a solution to P, then there will be a solution to P such that the maximum number of items in a room in any step of the solution is bounded by $p(n)$, where n is the amount of rooms in P.*

Lemma 3.2. *There is a fixed polynomial function $q(n)$, such that in any instance P of the Simple RDP, if there's a solution to P, then there will be a solution to P such that the maximum depth achieved in any step of the solution is bounded by $q(n)$, where n is the amount of rooms in P.*

Theorem 3.1. *The Simple RDP is in PSPACE.*

3.1 Gadgets

3.2 Reduction from 3-SAT

4 Recursed Solver

Currently, there is no available automatic Recursed solver. And this would be impossible to do if the Standard RDP turned out to be in Turing-complete. However, various heuristics might be used to check the solvability of a puzzle. For example, if there's no reference to a chest with a key, then it is certain that the puzzle can't be solved, and this was used to spawn crows in the game.