More NP Complete Games

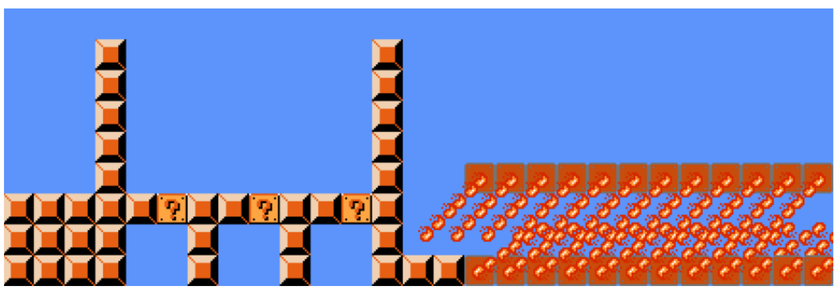
Richard Carini and Connor Lemp

February 17, 2015

Attempts to find an NP Hard Game[1]

As mentioned in the previous writeup, the search for an NP Complete game requires a lot more thought and trial-and-error than anticipated. We delved into Mario, searched through Zelda, and looked at Donkey Kong. It seemed that a lot of the games that were NP Complete involved moving a character from one portion of a game grid to the other, avoiding obstacles along the way. The decision problem was always centered on the question "Is this arrangement possible to complete?" We noticed that all the games we read used the fundamental rules of the game in order to construct possible sections of the game in order to reveal that the game was NP Hard. For instance, in the Super Mario proof, the following was a gadget construction:

Mario Gate[2]



This arrangement of item boxes, spinning Firebars, and blocks forces Mario to have jumped underneath an item box to have a Star item (granting Mario invincibility) in order to be able to traverse past the Firebars. Each of these constructions, although following the rules and mechanics of the game completely, were combined to make Super Mario represent a circuit that is completable if and only if the corresponding circuit is satisfiable. Using this information, we thought of another game that has these principle elements and decided we could try to see if this game led to any results. The game is called, perhaps aptly, the World's Hardest Game.

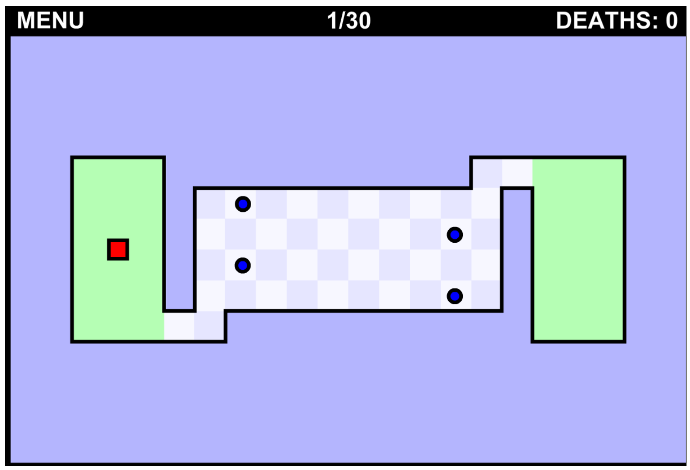
The World's Hardest Game

The World's Hardest Game begins with a brief instruction: "You are the red square. Avoid the blue circles and collect the yellow circles. Once you have collected all the yellow circles, move to the green beacon to complete the level. Some levels consist of more than one beacon; the intermediary beacons act as check points. You

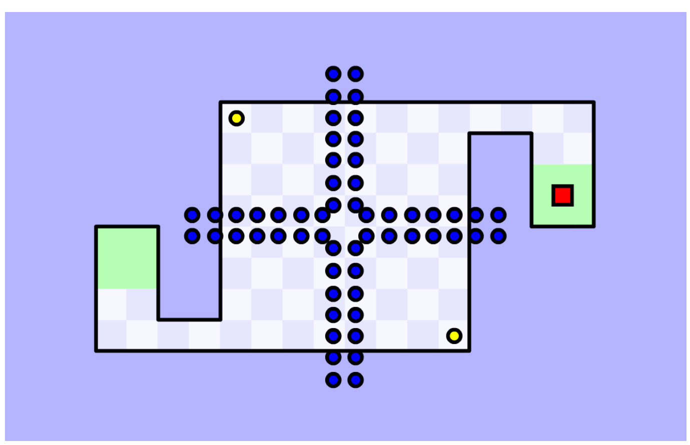
[1] Written by Richard Carini

[2] Aloupis, Greg et al.

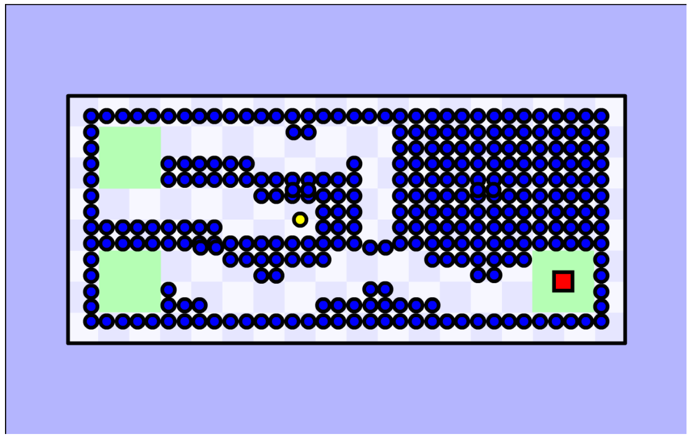
must complete all 30 levels to submit your score. Your score is a reflection of how many times you died; the less, the better." For our purposes, we will be working under the assumption that we don't have more than one death, and only need to complete one level. Also, we will not need to use the check points, as these are only implemented to make the game less frustrating for the player.
While this does give us some rules to work with, it does not allow us to look inside how the game mechanics may work. We need these mechanics so that we can manipulate the stages within the boundaries of the game in order to relate it to an NP Complete problem. In order to do this, we must begin playing the game. The first level is as follows:



This level if fairly straightforward. Four blue circles fly horizontally through the stage in an oscillating fashion. All four circles move at the same rate, and they move within a straight line. Delving further into the game, we receive information about a few more rules. For instance, in level 11, we see a spinning pinwheel of blue circles with two yellow circles to collect



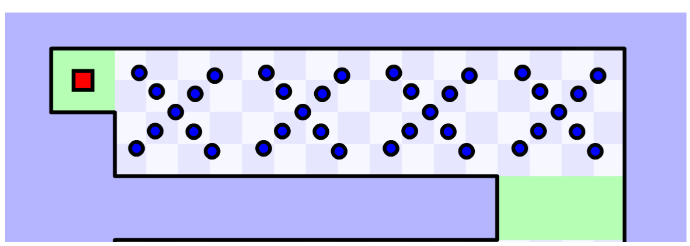
In this level, we realize that circles can have varying speeds, can move off of the game grid, and can even start and stop their motion. Finally, in level 12, we receive a large amount of information about the mechanics of our game:



This level lets us know that blue circles can be arranged into a “wall” formation and can move over one another. Overall, we have also noticed that the movement of the blue circles, although varied, remains periodic: the blue circles in each level always return to their original positions after a certain amount of time. Additionally, the yellow circles appear to always remain stationary, and there are no blue circles within the green beacon areas at any point in time.

Attempts to Create a Circuit from The World’s Hardest Game

Since all of the proofs that we have been introduced to so far dealing with NP Complete problems involved circuit satisfiability, it seems like a natural place to attempt to construct a circuit from The World’s Hardest Game in order to prove that the game is NP Complete. We start by attempting to create a wire. We constructed an arrangement that looks a lot like level 6 in the game:



Since all of the blue circle pinwheels are moving in a circular, clockwise motion, we are forced to move to the right along the top of the grid. However, we can go back if we reapproach the grid from the bottom; this is resolved if we place a few blue circles between the last two pinwheels, so the approach would result in a death. This creates a very sturdy wire, however we must be able to create AND’s and OR’s and declare truth values. Since all of the yellow circles need to be collected before ending the game, it seems like a natural AND operator. The OR’s could possibly be simply different rooms; the “rooms” (or passageways) with the value of TRUE must be maneuverable for our red circle, and the ones with the value of FALSE must not be. However, this raises more questions than answers: firstly, we have no way of initializing our variables with true or false statements. Additionally, if we wanted to change the value of our room from TRUE to FALSE or vice versa, we would need to reconstruct our entire grid to compensate. This method, in short, will not work for us. What we need now is another way of looking at the game.

A Different Approach

We recall that we can call a problem NP Complete if we are able to develop a correlation between that problem and a known NP complete problem such that one is true if and only if the other is as well. So far, we have introduced the problems of SAT, 3SAT, circuits, and vertex coverings as our "go-to" NP Complete problems. However, we have yet to introduce another problem that may yield more promising results for our game: the Hamiltonian Path. Although we will not formally introduce the proof for why this problem is NP Complete, we will show how this problem relates to our own game.

Hamiltonian Paths

We first introduce the idea of a Hamiltonian Path by delving into its definition: We say that an arbitrary graph G is traceable if there exists some path along the edges of the graph that visits each vertex of the graph and visits them all exactly once. Such a path along the vertices is called a Hamiltonian Path. This problem has been proven to be NP Complete and is used commonly in order to prove that other problems are NP Complete by relating them to Hamiltonian Paths. We must first show that the World's Hardest Game is in NP; then, if we are able to construct a World's Hardest Game grid that represents an arbitrary graph, and show that the World's Hardest Game is completable if and only if the graph is traceable, we can conclude our proof that the World's Hardest Game is NP Complete.

The World's Hardest Game is in NP

We can easily verify that the World's Hardest Game is in NP by definition. If we are given an arbitrary World's Hardest Game and given a "solution-path" for how to navigate through the game, it is easily verifiable by checking that solution.

The World's Hardest Game is NP Hard[3]

We begin our construction of our World's Hardest Game grid by having a green start area and one green exit area, each on opposite sides of the grid. Given an arbitrary graph, we start by creating a "room" for each of our vertices, all lined horizontally to each other between the exit and the entrance. We create a long

[3]Written by Richard Carini and Connor Lemp

path that extends from the Entrance to the edge of the farthest room that we have created, and another path that extends from the Exit to the edge of the first room we've created. Each room has one main entrance and one main exit to the room at the top, that connects to the main Entrance and Exit paths to the game. In order to keep the player from continually going back and forth between the rooms and the respective paths, we construct one-way gates (much like the one-way wires that we have introduced previously) that only allow the player to go from the Entrance path to the rooms area once, and from the rooms area to the Exit path once.

Each room contains a maze of blue circles that takes a time of T_{room} to navigate (the value of this time will be determined by other factors, which we will discuss in greater detail soon). Each room also contains a single yellow dot to collect. In addition to the two openings for the entrance and exit, each room will have more entrance ways at the top: one for each vertex that the room's corresponding vertex is connected to. For instance, in our given graph G, if the vertex V_1 connects to both V_2 and V_3, the room corresponding to V_1 will have one opening for the entrance, one opening for the exit, one opening for the room corresponding to V_2, and the last for the room corresponding to V_3.

Now we construct a series of pathways between each room that corresponds to our edges in our given graph. In our previous example, this would mean constructing a pathway that connects V_1 to V_2 and another pathway that connects V_1 to V_3. So far, we have an Entrance, an Exit, a number of rooms corresponding to a number of vertices, and an entanglement of pathways that represent edges and connections between different portions of our game. In order to keep the player from going from one edge to another without passing through a necessary room, we must present a crossover gadget. This gadget will appear at every four-way intersection in our game. There will be blue dots that over the widths of the two opposing entrances to our intersection, and after a certain amount of time, will slide back. After these dots slide back, other blue dots will slide over and cover the other two entrances of the intersection. This allows us to have as many crossovers as we would like, but keep our player from navigating between different paths.

We now place a row of blue dots across the top of the game grid that slowly lowers down onto the game. From our construction, we have that the player must enter from the Entrance path to the rooms area, traverse each room, collect every yellow dot from each room, exit to the Exit path area, and navigate to the green Exit. This must happen all before the blue dots reach the Exit pathway, which, by our construction, will be located at the top of the game.[4] We now must find the speed that the blue dots must be traveling downward such that the player can complete the game if and

[4]Note that the player may be able to reach the green Exit area well before the dots reach the pathway without collecting the dots. It can be argued that the player may be able to wait out the blue dots and let them pass without collecting all of the yellow dots; however, if the player does this, they are unable to go back into the rooms area because we have constructed one-way entrances to our Exit pathway. Therefore, the player cannot win; they have no choice but to lose and start over.

only if there exists a Hamiltonian path in the corresponding graph G.
We will now show that the collapsing ceiling dots will take $T = nT_{room} + T_{extra}$ for a graph on n vertices, where T_{extra} is the maximum time it could take to traverse all edges, leave the entrance and make it to the exit, and $T_{extra} < T_{room}$. This is to ensure that the player has enough time to complete each room and edge in between, but the extra time allotted to travel along each edge is not enough to go through a room multiple times. Thus we arrive at the formula

$$T_{extra} \leq (n+1)\left(T_{em} + T_{cross}\binom{n+2}{2}\right) + c$$

where T_{em} is the time it takes to travel along the longest edge, T_{cross} is worst case time it takes to navigate across a crossover gadget, and c is a small amount of time allotted for traveling between edges inside rooms. The $\binom{n+2}{2}$ is the maximum number of edges on a graph with $n+2$ vertices (n vertices plus the entrance and exit points). This is the maximum number of crossover gadgets any one edge could need.
The bound $T_{extra} < T_{room}$ is satisfiable once one realizes that T_{room} can be made arbitrarily large since we can make the length of the height of the rooms as long as we would like.[5] Thus the room can be a maze, and T_{room} can be as large as we like without increasing T_{extra}.
To summarize, we can construct an instance of World's Hardest Game from our graph G such that the level is completable in T time iff G has a Hamiltonian path. For one direction, if G has a Hamiltonian path, then we can follow that path in the World's Hardest Game instance to beat it. This will solve the World's Hardest Game instance, as it collects a yellow dot from each room and can make it to the Exit from the room it ends in. If the World's Hardest Game instance can be completed in time, we can follow the same series of rooms to find the Hamiltonian path on G. This tells us World's Hardest Game is NP-Hard. Combined with our proof that World's Hardest Game is in NP, we know the game is NP-Complete.

[5] T_{extra} is not correlated with the height of the vertex rooms, so each room can be arbitrarily tall since we stated that all connections between rooms would be made from the top of the room. This means we have as much room to work with as we want below the edges.

Sources:

The proof for the mentioned video games that were proven to be NP Complete, including Mario, Zelda, Donkey Kong, Pokemon, and others:
Aloupis, Greg, et al. *Classic Nintendo Games are (Computationally) Hard.*
http://arxiv.org/pdf/1203.1895v2.pdf

Play The World's Hardest Game for yourself: Critoph, Stevie. Snubby Land. *The World's Hardest Game.*
http://www.addictinggames.com/action-games/theworldshardestgame.jsp

The book that sparked the idea of using Hamiltonian paths for completing our proof:
Garey, Michael R., David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.*

Special thanks to Padraic Bartlett for helping provide the idea of the collapsing blue dots.