7-2008

The Game of Bridg-It

Sandy Dean

The Game of Bridg-It

Sandy Dean

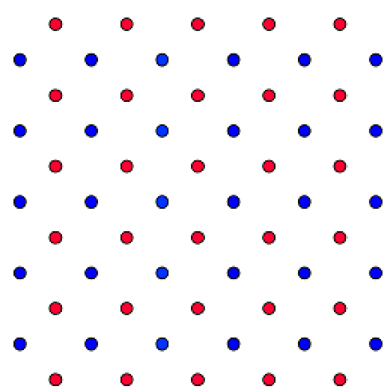
In partial fulfillment of the requirements for the Master of Arts in Teaching with a Specialization in the Teaching of Middle Level Mathematics in the Department of Mathematics.
University of Nebraska-Lincoln

July 2008

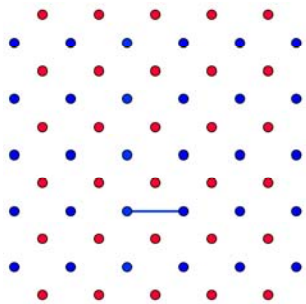
On the surface, Bridg-It appears to be a simple game of connecting dots to form lines across the board. Playing Bridg-It is simple. Understanding and playing Bridg-It well is more complicated. To understand the theory and strategy behind Bridg-It, one must first understand certain elements of graph theory, such as disjoint spanning trees. Only then can one master the game of Bridg-It.

What is Bridg-It?

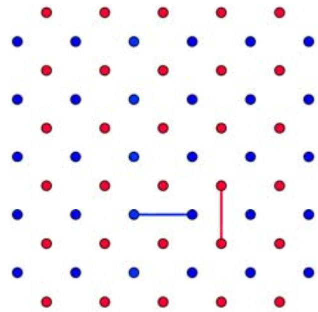
Bridg-It is a simple connection board game that was created around 1960 by a man named David Gale. Bridg-It was first published in *Scientific American* under the name Gale. It is a two-player game and the board consists of two rectangular arrays of dots, one array for each player. The players move by alternately connecting two dots within their array to form a line or bridge. The objective for player one is to build a bridge from left to right and the objective for player two is to build a bridge from top to bottom. The player who is first to create a bridge that connects their opposite edges of the board, wins.



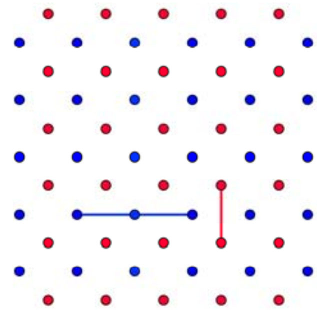
The rules to play Bridg-It are as follows: players take turns connecting two adjacent dots of their own color with a bridge. Adjacent dots are considered to be dots directly above, below, to the right, or to the left of another dot with the same color. A newly formed bridge cannot cross a bridge already played and whoever connects their opposite edges of the board first wins. To give you a better idea of how the game works, I'll demonstrate a sample game. Assume player one is blue and player two is red. Player one goes first.



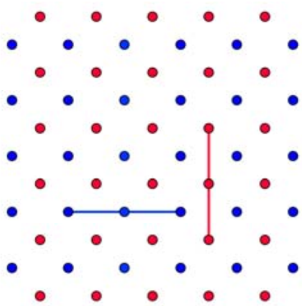
Player one creates the first bridge.



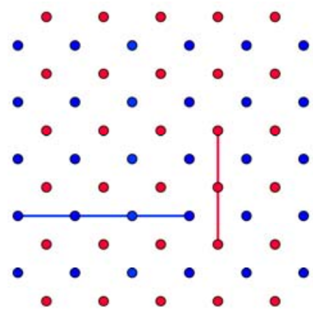
In response, player two attempts to block.



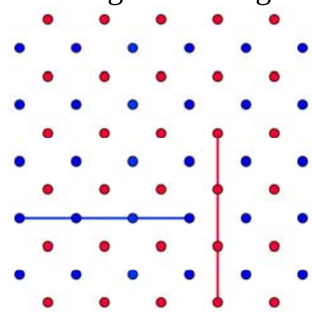
Player one continues to build bridge to the edge.



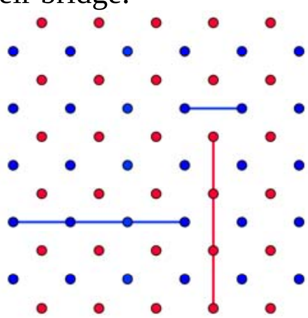
Player two begins to build their bridge.



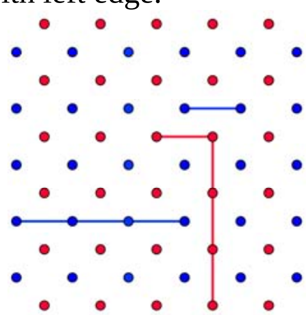
Player one connects bridge with left edge.



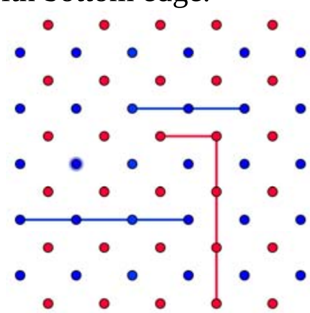
Player two connects bridge with bottom edge.



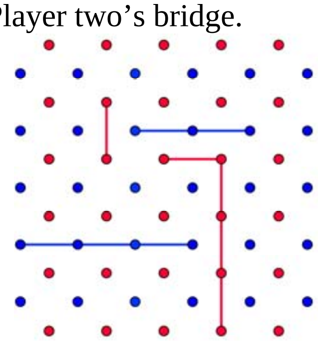
Player one attempts to block Player two's bridge.



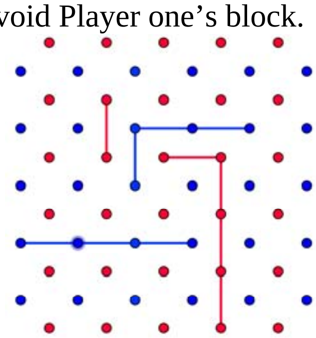
Player two attempts to avoid Player one's block.



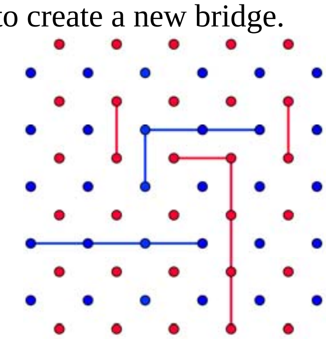
Player 1 blocks and begins to create a new bridge.



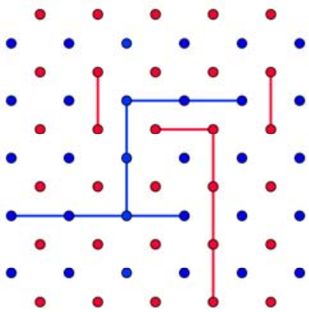
Player two blocks while still leaving the possibility to connect with old bridge.



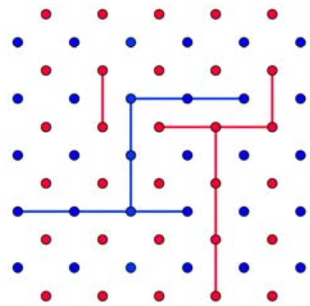
Player one blocks the connection of Player two's bridges.



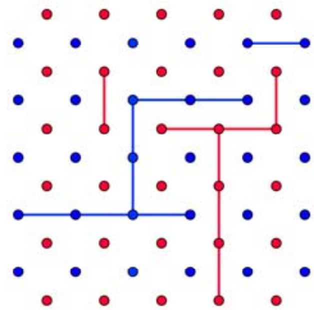
Player two attempts to create a bridge, while again blocking.



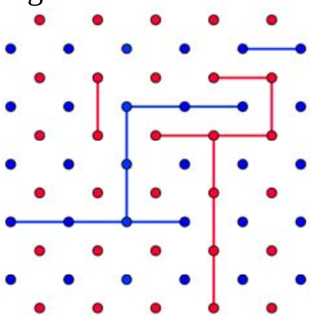
Player one connects bridges.



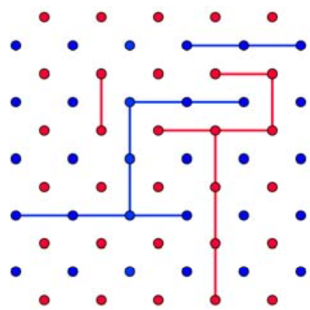
Player two connects bridges.



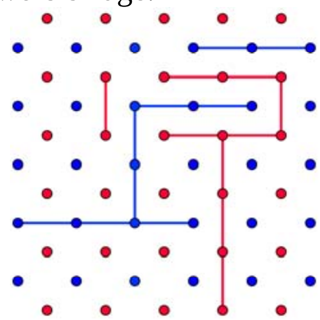
Player one blocks Player two's bridge.



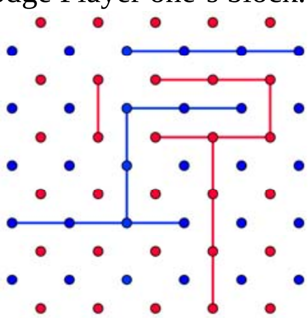
Player two attempts to dodge Player one's block.



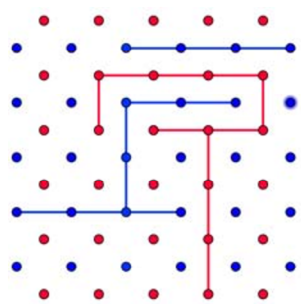
Player one blocks Player two's bridge again.



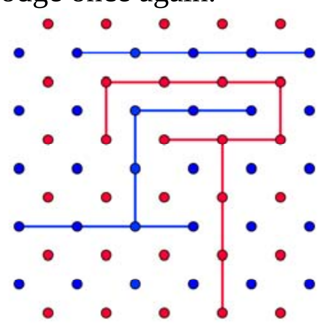
Player two attempts to dodge once again.



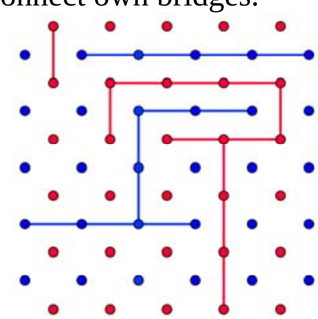
Player one blocks, but also creates an opportunity to connect own bridges.



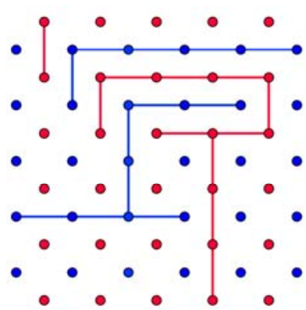
Player two blocks Player one's attempt to connect bridges.



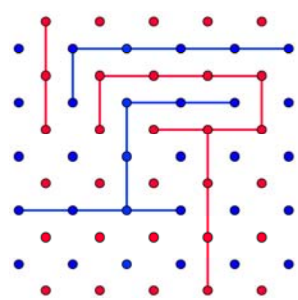
Player one blocks Player two's chance to win.



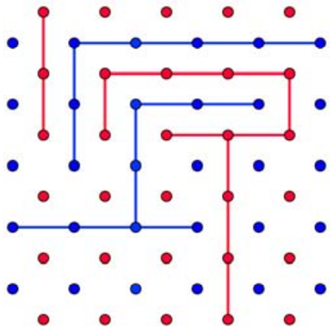
Player two blocks Player one's chance to win.



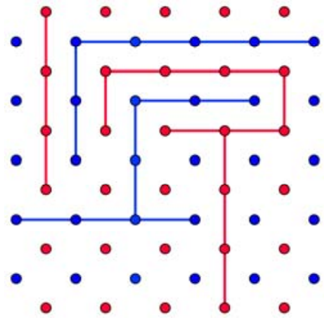
Player one moves towards connecting to the left edge.



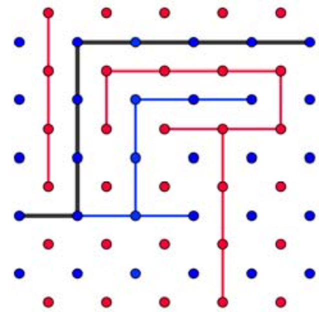
Player two moves towards connecting bridges and blocks Player one's win.



Player one creates two opportunities to win. One with immediate connection to left edge. The other by connecting to another bridge. The move also blocks Player two's chance to connect bridges.



Player two blocks one of Player one's opportunities to win.



Player one uses the other opportunity to win! They have connected the left and right sides of the board.

By creating a bridge that connected opposite sides, player one has won the game. In connecting the blue opposite sides, player one has also ended player two's opportunities to construct a bridge that connects the red opposite sides. Thus Bridg-It will never end in a tie since a bridge cannot be completely blocked except by a complete bridge of the opposite color.

As you can see, Bridg-It is a game anyone can play. Basically, it is a game of connecting dots to create lines. Anyone can play, but it is not an easy game to win. The key to mastering the game of Bridg-It lies within graph theory.

Graph Theory

When broken down to the basics, Bridg-It is a game based off of graph theory. The element of graph theory that the game is concerned with is disjoint spanning trees. To begin, let us look at some definitions that will aide in our understanding of graph theory. The following definitions were referenced from *MathWorld*--A Wolfram Web Resource, as well as, the textbook <u>Discrete Mathematics for Teachers</u>.

- A graph is a collection of vertices and edges connecting some subset, or portion, of them.

- If there exists a graph G' whose vertices and edges form subsets of the vertices and edges of a given graph *G*, then it is said that G' is a subgraph of *G*.
- A circuit is a path that begins and ends at the same vertex.
- A graph is connected if, for each pair of vertices *v* and *w*, there is a path from *v* to *w*.
- The degree of a vertex is the number of edges which touch the vertex.

Next let us look at the definition of a tree. As defined by the textbook Discrete Mathematics for Teachers, a tree is a connected graph with no cycles. As defined by the same textbook, a cycle in a graph is a simple circuit (a circuit in which no edges repeat) in which only one vertex (the initial vertex) is repeated and it is repeated exactly once. In other words, a tree is a graph in which the vertices are connected and there is no more than one way from a given point.

Now we will explore the nature of trees and their properties. To begin, we will look at three theorems from the previously mentioned textbook.

Theorem 1: In a tree there is a unique simple path between each pair of vertices.

Theorem 2: In a tree with *n* vertices, there are precisely *n* – 1 edges.

Theorem 3: In a tree that has been disconnected by the removal of one edge, two distinct trees are formed.

As we prepare to prove these theorems, we will rely on the characteristics of a tree – namely, that trees are connected and have no cycles. We will also use the fact that if a graph has a simple circuit (a circuit in which no edges repeat), then it will also have a cycle (a simple circuit in which only the initial vertex is repeated exactly once).

For Theorem 1, we intend to prove that in a tree there is a unique simple path between every pair of vertices. To do so, suppose that graph *G* is a tree.

Proof of Theorem 1:
Since *G* is a tree, it is connected by definition and so there is a path between each pair of vertices. We now will show that there is a unique path between each pair of vertices. Suppose that graph *G* has two simple paths, P_1 and P_2, between the vertices v_a and v_b. Let v_c be the vertex where P_1 and P_2 diverge. Then there must exist a cycle between P_1 and P_2 containing v_c. This contradicts our assumption that *G* is a tree. Therefore *G* must have a unique path between each pair of vertices

If graph *G* is a tree, then there is only one simple path between every pair of vertices.

For Theorem 2, we intend to prove proposition *P(n)*: If a tree has *n* vertices, then it has *n* – 1 edges.

Proof of Theorem 2:
Basis Step: A tree with one vertex will have no edges since an edge would create a cycle. Thus, since when $n = 1$ there is 1 -1 = 0 edges, *P*(1) is true.

A tree with two vertices will have one edge. Thus, since when $n = 2$ there is 2 – 1 = 1 edge, *P*(2) is true.

Inductive step: Suppose each tree with *k* vertices has $k - 1$ edges. Let *T* be a tree with $k + 1$ vertices. Let $x_1, x_2, x_3, \ldots, x_{r-1}, x_r$ be the vertices in a simple path in *T* of maximal length. Observe that x_{r-1} is the only vertex to which x_r is adjacent for the following reasons.

- If x_r is adjacent to any one of $x_1, x_2, x_3, \ldots, x_{r-2}$, this creates a simple circuit and the existence of a cycle. However, *T* has no cycles since it is a tree.
- The vertex x_r cannot be adjacent to vertices of *T* not on the path since the path has maximal length.

Thus x_r has a degree 1. Let $\hat{T}$ be the subgraph of *T* formed when vertex x_r and the edge $\{x_{r-1}, x_r\}$ are deleted from *T*.

Suppose *u* and *v* are vertices in $\hat{T}$. By Theorem 1, there is a unique simple path in *T* from *u* to *v*. Further, since x_r has degree 1, x_r is not on this path. Thus the path is a path from *u* to *v* in $\hat{T}$, so $\hat{T}$ is connected. Since *T* has no cycles, $\hat{T}$ has no cycles. Thus $\hat{T}$ is a tree with *k* vertices. By inductive hypothesis $\hat{T}$ has $k - 1$ edges. Thus *T* has *k* edges. By mathematical induction we have shown that for a tree with *n* vertices, then it has $n - 1$ edges.

For Theorem 3, we intend to prove that if a tree is disconnected by the removal of a single edge, then two distinct trees are formed. To do so, suppose that graph *G* is a tree.

Proof of Theorem 3:

Since *G* is a tree, by definition it is connected with no cycles. Since there are no cycles, there exists a unique path between each pair of vertices. Suppose the graph *G* has vertices v_a and v_b which are connected by an edge *e*. Suppose edge *e* is removed from the graph. In graph $G - e$ vertices v_a and v_b are no longer connected. Hence $G - e$ has two distinct connected components, one containing v_a and one containing v_b. Observe that removing an edge will not create a cycle, both components are acyclic. Since both components are connected and acyclic, by definition, both are trees.

If a tree is disconnected by the removal of one edge, then two distinct trees are formed.

After exploring the characteristics of trees, we are now ready to move on to discussing the type of trees that are used in the strategy of Bridg-It, spanning trees. By definition from the textbook Discrete Mathematics for Teachers, a spanning tree of a graph *G* is a subgraph of *G* that includes all the vertices of *G* and is also a tree. Basically a spanning tree is a tree than connects every vertex in graph *G.* Since a spanning tree is a type of tree, then it shares the characteristics of the definition of a tree, as well as, the characteristics of Theorems 1, 2, and 3:

- A spanning tree is a connected graph.
- A spanning tree contains no cycles.
- In a spanning tree there is a unique simple path between each pair of vertices.
- In a spanning tree with n vertices, there are precisely $n - 1$ edges.
- In a spanning tree that has been disconnected by the removal of one edge, two distinct trees are formed.

Every connected graph contains a spanning tree. In fact every connected graph with three or more vertices contains more than one unique spanning tree. Through research, I found that every complete graph, a graph in which each pair of vertices is connected by an edge, has n^{n-2} unique spanning trees. This is known as *Cayley's formula*.

Now let us look at some real life spanning trees, such as a computer network or an electric power network. We know that in a spanning tree every vertex is connected to every other vertex. Since there are no cycles, we also know that the removal of any edge disconnects the graph. If an edge was removed in one of these networks, it could potentially cause problems. If an edge goes down in an electric power network, anyone connected after the removed edge went down, would no longer have electricity. The network needs a more reliable design, a design that if one edge was removed, another edge could take its place. Many times networks create a second spanning tree just for this purpose, to re-connect the network if an edge is lost.

When creating a second spanning tree, it is important that it does not share an edge with the original spanning tree. That would defeat the purpose of having the second spanning tree, the ability to re-connect if an edge was lost. Two spanning trees that connect all the same vertices, but do not share an edge are called two edge-disjoint spanning trees. The purpose in creating a second spanning tree is to ensure if an edge is removed or failed, the network will continue to work.

Just how many edges are needed from the second spanning tree to reconnect the original spanning tree?

Theorem 4: In two edge-disjoint spanning trees, if one spanning trees is disconnected, then a single edge from a second disjoint spanning tree can reconnect the original spanning tree.

Proof of Theorem 4:
Suppose that graph G has two edge-disjoint spanning trees, a primary tree H_1 and a backup tree H_2. Suppose H_1 had an edge, e_1, removed. By Theorem 3, $H_1 – e_1$ is now two distinct trees, T_1 and T_2. H_2 contains a path, P, from one endpoint x of e_1 to the other endpoint y. Since P starts in T_1 and ends in T_2, it must contain an edge, $e_{2,}$ with one endpoint in T_1 and one endpoint in T_2, thus e_2 reconnects T_1 and T_2, resulting in the spanning subgraph $H_1 – e_1 + e_2$.

By Theorem 2, let H_1 be a tree with n vertices and $n – 1$ edges. When e_1 is removed the disconnected H_1, $H_1 – e_1$, has $n – 2$ edges. When e_2 is added to $H_1 – e_1$ to reconnect the tree, $H_1 – e_1 + e_2$, the resulting tree has $n – 2 + 1$ or $n – 1$ edges.

Since $H_1 – e_1 + e_2$ is connected and has $n – 1$ edges, by definition $H_1 – e_1 + e_2$ is a spanning tree.

Therefore, only one edge of the second spanning tree is needed to reconnect all vertices of the original spanning tree.

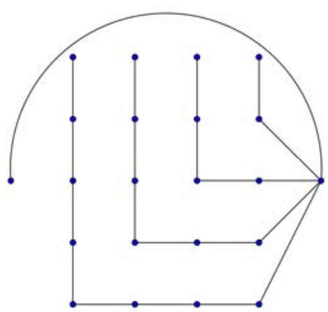
A Winning Strategy for Bridg-It

This is the same strategy used in Bridg-It. There are two edge-disjoint spanning trees to connect all the dots on the board. If a player removes an edge from the original spanning tree by putting their own bridge through it, you can use a single edge from the second spanning tree to reconnect your original spanning tree.

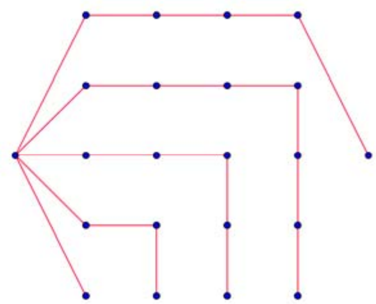
We can consider player one’s part of the board as a graph. In doing so, we can create two edge-disjoint spanning trees on the graph. Each time player two makes a move, they are erasing an edge from player one’s original spanning tree. Each time player one makes a move, they are making an edge that is off limits for player two to erase. Player one will always be able to win because if player two erases an edge from the original spanning tree, player one can use an alternate edge from the second spanning tree to reconnect their graph.

In creating two edge-disjoint spanning trees in player one's graph, we can think of the sides as a single combined vertices or post. We can do so because the bridge only needs to connect with a single vertex of each side to win. It doesn't matter which vertex it connects to. Then we can connect all vertices in the graph, making sure not to share edges with the other spanning tree.

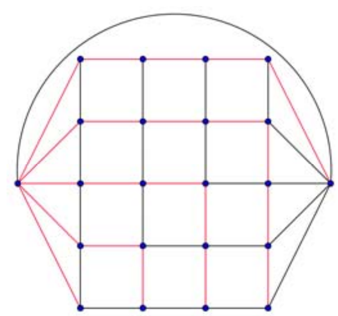
Original Spanning Tree



Second Spanning Tree



Both Spanning Trees



Assume that player two takes the "zeroth" turn. On this turn, they take away the edge on the original spanning tree that connects the two posts, since that edge was not initially part of the game. Next, player one can make an arbitrary move to choose some edge. In choosing this edge, player one has made that particular edge impossible to erase or cut by player two. Also by choosing this edge, player one is using the edge in both trees, making it in essence a double edge. Player two cannot cut a double edge. They cannot block both trees the double edge is a part of. They can only block the edge from one tree, leaving the other tree connected for player one to continue their bridge.

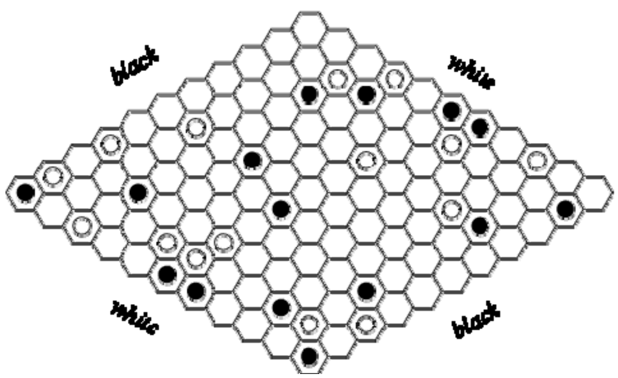
The strategy for player one is to defend and block the strategy of player two. In doing so, player one continues to create more double edges, of which player two can only block on one tree. The double edges created by player one gradually form a tree that connects the opposite sides of the board thus creating the winning bridge.

Player one has the upper-hand in Bridg-It because they can use the two edge-disjoint spanning trees to eventually build a bridge that connects their opposite sides of the board. The only way player one can lose is if they waste a move by either not defending properly against player two or by creating a cycle. If player one does make a mistake by wasting a move, player two could capitalize and seize advantage to win. Otherwise player one is guaranteed to win.

Player one can use the two edge-disjoint spanning trees to ensure victory because each time player one creates an edge to defend against player two's strategy, they are creating an edge in both trees or a double edge. The game ends when only double edges remain. These double edges connect to form a tree held by player one. The tree is actually a bridge player one has constructed between opposite sides of the board to win!

Similar Games

There are many connection games similar to Bridg-It, including Twixt, Star, Shannon Switching Game, and HEX. I'm going to focus on the latter, HEX. HEX was invented by Pat Hein in 1942 and also independently invented by John Nash, the mathematician in the book A Beautiful Mind, in 1948. Originally HEX was called John after its inventor and for where it was frequently played, on the tiles of bathroom floors. The game received the name HEX when a commercial version was issued by the company Parker Brothers in 1952.



HEX is a two player game on a diamond-shaped board made up of hexagons. The game is played usually on an 11x11 board for a total of 121 hexagons, although Nash believed a 14x14 board to be of optimum size. Player one is typically the white pieces while player two is typically the black. The two players take turns between placing their color on the board. Placement is only allowed in unoccupied hexagons. The goal to HEX is to complete a chain of pieces between two sides of the same color.

Much like Bridg-It, HEX cannot end in a tie. The only way to stop your opponent in HEX is the same as in Bridg-It, for you yourself to create a chain that crosses the board to opposite sides of the same color. Unlike Bridg-It, there is no explicit strategy known for HEX. Although, there is not an explicit strategy, player one has the upper-hand in HEX by strategy stealing.

Strategy stealing is when player one uses the exact same winning strategy as player two, thus guaranteeing the win for player one. Strategy stealing for player one works in the following manner. Player one begins by making an arbitrary move. Player two makes a move towards a winning strategy. Player one then "steals" the strategy by making moves that follow player two's strategy. If the strategy ever calls for moving in a hexagon already chosen, player one can make another arbitrary move. Extra moves can only improve a player's position in HEX. Suppose player two has a winning strategy, then by strategy stealing, player one also has a winning strategy. Since both players cannot win and the game cannot result in a tie, there is a contradiction, thus a winning strategy for player one. Strategy stealing in HEX ensures the win of player one the same way it ensured the win in Bridg-It.

Classroom Activity

The class that I designed my activity around was a seventh grade classroom. Graph Theory is not in the curriculum, but I definitely believe it is something students could grasp if it was introduced. I would begin the activity by explaining the rules of Bridg-It to my students. I would also post the rules in my room so students could refer to them. Then I would have my students play a few games of Bridg-It against one another to get an idea of what the game was like. After students had time to experiment with the game, I would get the class back together and ask "What can you do to win Bridg-It?" Students would not have enough to exposure yet to give me the strategy of how to win Bridg-It, but we could begin brainstorming on what the possible strategies could be.

After the discussion, I would give students an opportunity to play the game a few more times against one another. After the allotted time, I would have students individually write down what they think the strategy is to win at Bridg-It. Next I would ask students to try certain strategies of Bridg-It against each other. When they were done, we would bring back their results to the class. Each strategy needed to be attempted four times by each pair, each student getting the opportunity to be player one twice. Students are to keep track of the wins, not in name, but by whether player one or player two won. Students are to also write the results of each strategy attempted.

Player 1	Player 2	Please Explain Results
Strategy: Form a bridge to the opposite side as quickly as possible; don't worry about what Player 2 is doing. **Number of Wins:**	**Strategy:** Form a bridge to the opposite side as quickly as possible; don't worry about what Player 1 is doing. **Number of Wins:**	
Strategy: Form a bridge to the opposite side as quickly as possible; don't worry about what Player 2 is doing. **Number of Wins:**	**Strategy:** Don't worry about forming a bridge, strictly worry about blocking any bridge Player 1 might form. **Number of Wins:**	
Strategy: Don't worry about forming a bridge, strictly worry about blocking any bridge Player 2 might form. **Number of Wins:**	**Strategy:** Form a bridge to the opposite side as quickly as possible; don't worry about what Player 1 is doing. **Number of Wins:**	
Strategy: Don't worry about forming a bridge, strictly worry about blocking any bridge Player 2 might form. **Number of Wins:**	**Strategy:** Don't worry about forming a bridge, strictly worry about blocking any bridge Player 1 might form. **Number of Wins:**	
Any other strategies suggested by students should also be included.		

After students completed trying the different strategies, the class would get back together and report results. Using each strategy, we would discuss how many times player one won as compared to player two. Is there an obvious effective strategy if you are player one? Is there an obvious effective strategy if you are player two? Is it more effective to be player one, player two, or is there any difference in your chance of winning between the two? We would discuss as a class what our results meant.

Next we would play again, this time against the computer. Students could pick what they thought to be the most effective strategy and whether there was an advantage between player one and player two. I would have students write down the strategy they attempted and whether they were player one or two. Then students would play the computer ten times, keeping track of their wins and losses. Again, we would report our results back to the class and discuss what we thought was going on in answer to the question "What can you do to win Bridg-It?" If students had figured out the strategy, we would move on to the next part of the activity. If not, we would continue playing Bridg-It, exploring different strategies until students discovered the correct one.

After students discovered the strategy to winning Bridg-It, we would begin to discuss the graph theory behind it. I would give students the definition of a tree, followed by several pictures of graphs that were trees and graphs that were not. Students would decide if the graph was a tree or not. If not, students would explain why it was not a tree. Once I thought students had a conceptual understanding of what a tree was, we would move on to spanning trees. We would gain understanding of what a spanning tree was by drawing examples. I would ask students to draw a certain number of vertices, and then attempt to draw a spanning tree to connect their vertices. Again, we would repeat this process until I thought students had a conceptual understanding of a spanning tree. For homework that night, I would challenge students to draw a spanning tree on the Bridg-It board.

The next day students would bring back their attempts. I understand that not every student may have finished their spanning tree, but we would discuss what students tried, what ideas failed, and what ideas succeeded. Hopefully at least one student would have completed a spanning tree that we could show to the class. I would then show students the two edge-disjoint spanning trees used in the strategy of Bridg-It and explain why it worked. I would then have students play Bridg-It on a piece of paper that had the spanning trees sketched lightly on them. In doing so, hopefully students could see that as player two erased an edge of the tree, they could use an edge from the other tree and still be victorious.

I believe from start to finish the activity would take approximately three days. As previously mentioned, graph theory is not necessarily part of the seventh grade curriculum. I believe students' reasoning skills could benefit greatly from attempting different strategies to see which was most effective. While the activity would teach graph theory to the students, the main learning focus will be attaining the reasoning skills that will allow them to pick the most effective strategy. This is a skill they will use many times in their future.

REFERENCES

Crisler, N., Fisher, P., & Froelich, G. (1994). *Discrete Mathematics Through Applications*. New York, NY: W.H. Freeman and Company.

Lau, L.C. (2007). Introduction to number theory. Retrieved June 28, 2008, from Discrete Mathematics Web site: http://www.cse.cuhk.edu.hk/~chi/csc2110/notes.html

Thompson, M. Bridg-It. Retrieved June 28, 2008, Web site: http://home.flash.net/~markthom/html/bridg-it.html

Weisstein, E.W. Game of Hex. Retrieved June 28, 2008, from From MathWorld--A Wolfram Web Resource Web site: http://mathworld.wolfram.com/GameofHex.html

West, D.B. (1996). *Introduction to Graph Theory*. Upper Saddle River, NJ: Prentice-Hall.

Wheeler, E., & Brawner, J. (2005). *Discrete Mathematics for Teachers*. Boston, MA: Houghton Mifflin Company.

Wikipedia, HEX. Retrieved June 28, 2008, from Thompson, M. Bridg-It. Retrieved June 28, 2008, Web s Web site: http://en.wikipedia.org/wiki/Hex_(board_game)