Contents

Object of the Game

Span-It! is a two-player connect-the-dot game. Each player is assigned an orientation at the start of the game. One player is horizontal and the other is vertical. Players alternate turns placing connectors on the board. The object of the game is for each player to span the board in the direction of his/her favored orientation by completing a path from one end to another.

The challenge is to block your opponent while still making progress. A game will never end in a draw. Someone always wins (or quits!)

By default, *Span-It!* assumes the horizontal player is human, and the vertical player is the computer using a default personality. Any combination of human and computer players is allowed.

Contents

What are Personalities?

Span-It! has an exciting option that is not available in many games. Not only is it possible to play against the computer, but you may define **virtually every aspect** of how the computer plays.

Like many game playing programs, *Span-It!* assigns a number to the perceived importance of every empty position on the game board. This number is called the weight. The board position (node) with the best weight is the one *Span-It!* will select for the next move.

The computer assigns the weights based on a well-defined set of factors. For example, *Span-It!* will add the value of the *PathIsWinner* factor to the weight of an empty position that can win the game. There are several factors that influence the weight of a node.

A *personality* is a set of values assigned to the factors used by the computer during auto-play mode. When you press one of the "Load Personality" buttons in the new game dialog box, you may select different personalities. Some personalities are more "intelligent" than others. You also may edit or create your own.

Remember that personalities only influence the way the computer plays. Human players probably use different methods to "calculate" moves. The selected personality for a human player is the one that the HINT menu option uses. The way to tell the computer to play instead of a human is by checking the "computer plays" box in each player definition section of the new game dialog box.

It is possible to have the computer play itself using two different personalities. The "referee" is a completely different part of the program. The opposing personalities are not allowed to cheat!

Contents

Registration

Span-It!

Span-It! is shareware. You may try the program free of charge and may make copies for others. If you continue to play Span-It!, a $15 registration fee is required. This program may be distributed freely or for a nominal media charge (less than $5) provided that no modifications are made to any files. Authenticity may be verified for free by the program's author via. SASE for a limited time. For registration information, see the file REGISTER.TXT or REGISTRATION under the HELP menu.

THE PROGRAM IS PROVIDED "AS-IS". NO WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, ARE MADE AS TO IT OR ANY MEDIUM IT MAY BE ON. NO REMEDY IS PROVIDED BY THE AUTHOR FOR INDIRECT, CONSEQUENTIAL, PUNITIVE OR INCIDENTAL DAMAGES ARISING FROM IT, INCLUDING SUCH FROM NEGLIGENCE, STRICT LIABILITY, OR BREACH OF WARRANTY OR CONTRACT, EVEN AFTER NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.
USE AT YOUR OWN RISK.

For $15 registration (plus shipping) you will receive:

1. Your own **personalized copy** of this and the next release of *Span-It!*
2. **Prototypes** of *Span-It!* on alternate operating systems.
3. The ***Span-It! Toolkit***. Several programs that will help you create better personalities.
4. One entry per person in the free ***Span-It!*** **Ultimate Personality Competition.** (Registration is not required to enter the competition. Just mail a disk or printout of your best *Span-It!* personality to the registration address below. Winning personalities will be included in the next release of *Span-It!*)

Registration Forms can be printed via. the help screen (FILE | PRINT TOPIC) or else you may copy the file REGISTER.TXT to the printer from DOS by:

```
C:\SPANIT\> COPY REGISTER.TXT PRN:
```

Please send registration of $15 U.S. (plus $2 shipping in Continental United States, $5 elsewhere) to:

Mark Chapman
Span-It! Registration
1126 Wauwatosa Rd.
Cedarburg, WI 53012

Your name and shipping address:

Media type (3.5" or 5.25" floppy disk):

Your Span-It! handle (the name to appear on the personalized "About" screen):

Don't forget to enter your best personality in the ***Span-It!*** **Ultimate Personality Competition.**

For academic discussion, my Internet address is chapman@miller.cs.uwm.edu.

Your suggestions are always welcome.

Watch for the I.B.M. OS/2 2.1 version soon!

Thank You for trying my game.

Contents

Personality Factor Descriptions

Before reading this section, please make sure you are familiar with the material in Basic Personality Concepts

This is a step-by-step description of how the value of a trial node is computed. Part A describes how the weight of a single trial node is computed. Part B describes how all trial nodes on the board are compared to select the best move.

Part A: Compute the weight of a single trial node

1. Determine the initial weight of the trial node.

NodeMajor: If the trial node is a major node, assign NodeMajor to the initial weight of the trial node.

NodeMinor: If the trial node is minor node, assign NodeMinor to the initial weight of the trial node.

NodeRandom: Always add a random number between 0 and NodeRandom to the weight of the trial node.

2. Add the change in path weight to the weight of the trial node.

The path weight may be computed for any Path. It does not matter if the path is a Real Path or a Resultant Path. All we care about are the characteristics of the connected set of nodes. The order the nodes were connected will not change the path weight.

A path weight PW is computed as follows:

PathMajorSpan: Assign variable PS to the number of rows or columns the path spans in the major direction. Assign the initial path weight to:

PW = (Sign of PathMajorSpan) * (AbsoluteValue of PathMajorSpan)^PS.

PathMinorSpan: Assign variable ps to the number of rows or columns the path spans in the minor direction. Add the PathMinorSpan factor into the path weight by the following formula:

PW = PW + (Sign of Path Major Span) * (Absolute Value of PathMinorSpan)^ps.

PathIsAnchored: If the path is connected to a major edge, add PathIsAnchored to the value of PW. (Note: If a path is anchored at two opposite major edges, the path is a winner).

if the path is connected to a major edge,

PW = PW + PathIsAnchored

PathIsWinner: If a path spans the board in the major direction, add PathIsWinner to W. (The value of PathIsWinner usually is very large compared to every other factor.)

If the path wins the game

PW = PW + PathIsWinner

PathMemberCount: Assign M to the total number of nodes in a path.

PW = PW + PathMemberCount ^M .

Now that we know how to compute the weight of a path, let's see how path weight influences the value of the trial node.

A. Identify the real paths that the current player owns that are adjacent to the trial node.

B. Compute the Existing Path Weight (EPW) as the sum of the PW of the paths identified in (A).

C. Generate the resultant path.

D. Compute the Resultant Path Weight (RPW).

E. Add the change in path weight (RPW minus EPW) to the weight of the trial node.

3. Add the change in board weight to the weight of the trial node.

BoardPathCount: Assign np = the number of new paths created by selecting the trial node.

NP = 1 (When a new path is started)
NP = 0 (When an existing path is expanded)
NP = -1 (When two paths are merged)

Now add NP times BoardPathCount to the trial node weight.

Part B: How all trial nodes are compared to select the next best move

GamePredictability%: Let MyW = value of selecting the trial node. Let YourW = weight if the auto-player uses the current personality to assess the value if **the other player** were to select the trial node. (The referee does not allow "cheating". To calculate YourW, the auto-personality has to use its own weights.) The final value W of the trial node is computed by:
W = MyW + (GameTreePredictability% * YourW) / 100

GameTreeMaxBreadth: The computer attempts to "look ahead" a certain number of moves to help decide the weight of each free node. It could take a very long time to evaluate every possible outcome of a single game. GameTreeMaxBreadth is the factor that limits the breadth of the beam search used for the game tree.

GameTreeMaxDepth: GameTreeMaxDepth is the maximum number of moves ahead

the computer looks. KEEP THIS VALUE SMALL. The total number of moves that the computer will look at has an upper bound of:
GameTreeMaxBreadth ^ GameTreeMaxDepth
abbreviated to: b ^ d.

A 486-33 can compute about 1,000 moves in a reasonable amount of time. 2,000 moves takes about twice as long. The important thing to realize is that the number of moves is exponential with respect to d. If b = 10 and d = 3, the computer already will examine up to 1,000 moves per turn. If you let b = 10 and d = 6, the computer will examine up to 1,000,000 moves per turn!

Let b and d remain small! Otherwise the computer will just take to much time to respond. It will be interrupted at the Maximum Auto Move time; thus, the personality will not get a chance to finish the calculations anyway!

GameTreeBreadthTrim%: In addition to the maximum auto move time function, each personality has an option to trim or increase the breadth of the search as it descends levels in the game tree. It is most likely that you would want this factor to be a positive number. Otherwise the breadth at each level will increase and the time to select a move will grow very quickly.

GameTreeValueFade%: If you read this far, you deserve to find out that the personalities don't really use an actual "game tree" to look ahead moves. I made my own algorithm to allow experimentation into how important the value of the perceived weights change as we look deeper into the game tree. This factor is used to fade or increase the value of the possible moves when looking ahead. It is applied as the values are passed back up each level. A positive value means that the value of each level decreases based on the level of the game tree.

Basic Personality Concepts
Contents

Basic Personality Concepts

The information in this section is crucial to the understanding of the Factor Definitions, It is assumed that you have played a few games of *Span-It!* It may be helpful to examine the Board Diagram.

Vocabulary:

Node
Major Node
Minor Node
Orientation
Owner
Path
Trial Node
Resultant Path
Real Path
Path Major Span
Path Minor Span

Factor Definitions

Contents

About the Author

Education:

Bachelor of Science University of Wisconsin, Milwaukee 1992 (Computer Science.)
Programmed in C and C++ in a UNIX environment.
Worked on a wide variety of projects, including parallel processing and infinite precision arithmetic.
Currently a graduate student at UWM studying Cryptography and Data Security.

Work Experience Overview:

Worked at least five years at a local consulting firm.
Managed projects and coordinated personnel.
Supplied primary customer contact for several major accounts.
Prepared and presented custom sales and training seminars.
Designed and programmed a wide range of applications.
Developed software for UNIX, DOS, WANG VS and other operating systems.
Programmed in C, COBOL, Paradox(TM), Informix(TM), and more.
Helped install and administer Novell(TM), LanTastic(TM), and UNIX Networks.
Distributed data collection applications with hand-held computers.
"Downsized" several operations from old minicomputers to UNIX or PC's.

Self-study highlights:

Artificial intelligence and game playing.
Windows 3.x (tm) API (obvious!)
OS/2 (tm) PM Programming (*Span-It!* is on the way...)

Contents

A quick thanks to my good friend, Jay Dittmann.
His persistence and detailed criticism pushed the limits of his free time, my compiler, our modems,
and most wonderfully -- the quality of the final product.

Contents

Node: a position on the board.

(Sample)

Major Node: a position on the board that always connects in the direction of the owner's *orientation*. (See "maj" in the diagram)

Minor Node: a position on the board that always connects in the **opposite** direction of the owner's *orientation*. (See "min" in the diagram)

Orientation: the direction in which a player is trying to connect nodes to win the game. The player who is trying to connect top to bottom has the *vertical orientation*. The player who is trying to connect left to right has the *horizontal orientation*.

The orientation does not refer to any specific node. It only refers to the players and the object of the game.

Owner: the player who has placed a connector in a node is the owner of that node.

Path: a connected set of nodes owned by the same player. The horizontal player owns two paths in the diagram.

Path Major Span: the number of major nodes spanned by a path. In the diagram, the horizontal player's path in the upper-left hand corner has a major span of three. The horizontal player's other path has a major span of zero. If the vertical player moves in the major node highlighted in purple, the Resultant Path will also have a major span of three.

Path Minor Span: the number of minor nodes spanned by a path. In the diagram, the horizontal player's path in the upper-left hand corner has a minor span of 1. The horizontal player's other path also has a minor span of 1.

Trial Node: a single empty node that is the target of the current weight evaluation during auto play.

Resultant Path: the path that would be created if the Trial Node were selected as the next move. In the diagram, consider the "if vertical goes here" node as the trial node. The PathMajorSpan and PathMinorSpan graphed in green is that of the resultant path.

Real Path: a path that is already visible on the board.

Diagram of the *Span-It!* Board: