

Brain Blaster: A Board Style Quiz Game

By Whiskers

Contents

Introduction

Board Layout

The Spinner

Playing Piece Movement

Question Generator

Odds and Ends

Introduction

While there is nothing overly complex in the design of this game, it's recommended that the reader have more than just a basic knowledge of MMF2. In writing this tutorial you will find that I fluctuate between over simplification of some details, while at other times assuming the reader knows what I'm talking about on others. The challenge was to find a happy medium for all readers. In addition, it's recommended that you go to the folder containing this tutorial and first play the game to see how it works.

How the Game is Played

In this tutorial the reader will learn to make a simple one-player game that challenges the user to correctly answer questions, as he or she moves around a classic board layout. In playing, the program's presentation will resemble this: A display screen of several topics or skill levels is displayed to the user. The player then clicks on a selection of his/her choice and the game play begins with a board layout appearing. The player now spins a spinner and moves the playing piece along the path of squares matching the number of the spin. After completing the move, a randomly produced question now appears on the screen. Points are added or subtracted from the participant's score depending upon the answer chosen by the player. As the player moves around the board, he or she must answer enough questions correctly to win the game. The winning game is brought to a conclusion in an animated and fanfare fashion.

The working of the game relies primarily on counters and alterable values to keep everything under control. In order to easily follow the events involved, I'll described them in an organized manner, building precept upon precept as we go. Your finished product should have around 100 questions in its "data base" to keep the game interesting and to avoid repetitive questions as much as possible.

I've purposely left out sound effects, as these will simply rely on whatever you have in your collection. I would recommend that you give the spinner a unique sound, use a soft click for each move of the playing piece, and a fanfare sound upon winning the game. In order to keep the tutorial as concise as possible, I've also left out some of the basic procedures used in MMF2, as this info can be gathered from other tutorials or the help files.

How the Game Works

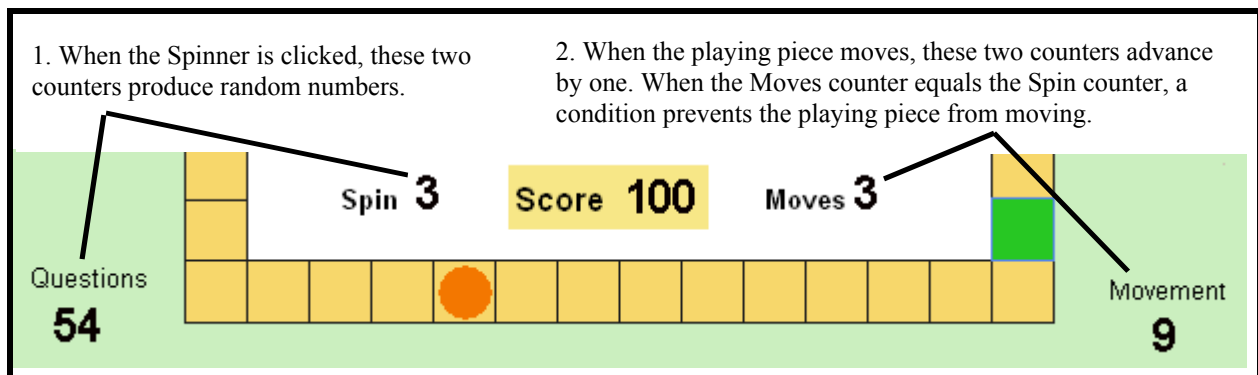
(1) The game begins when the player clicks on the "Spinner". This action activates a counter to "spin" rapidly. After one second, the counter stops on a random number.

(2) The playing piece now moves automatically from square to square. As the piece moves, a second counter advances by 1 for each move. When this second counter reaches the same number as the spinner, the playing piece stops.

(4) A third counter (out of view) constrains the playing piece movement to the proper direction when moving. This is done as follows: each time the playing piece moves, this third counter is advanced by one. When this counter reaches 13 the playing piece will now be as far as it can go on the bottom squares. An alterable value now changes allowing the piece to move upward on the board. This method of playing piece control (that is, by using alterable values) is continued until the piece has moved completely around the board.

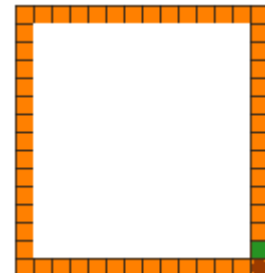
(3) Now a random question appears on the screen. How is this question rendered? A fourth counter, out of view, produces a random number between one and one-hundred whenever the player clicks the "Spinner". A question matching the number of this counter is produced on the playing screen. The spinner is non-operable until the player attempts to answer the question.

(5) In addition to the four counters, numerous alterable values control many of the events. Everything else in the game is left to the designer's choice.



Part 1: The Board Layout

1.1 The playing board consist of 51 squares arranged as illustrated. These squares are 32 x 32 (default size) background squares. To produce these, double click in the Frame Editor and select the Backdrop icon. Open the Backdrop in the graphics editor and change it to a light color with a black boarder. Next, duplicate (cut and paste) this backdrop square numerous times and arrange them in a rectangular layout like a classic board game. Use the keyboard arrow keys to nudge them into



place. The final square (the one to reach in order to win) is an active square of a different color. **Important: In order to follow this tutorial, lay out and position the exact amount of squares, as illustrated. There are 14 squares along the bottom and top. The sides (not counting the corner squares) consist of 13 each.**

1.2 A playing piece will now be constructed and placed on the starting square, as follows. Insert an Active Object onto the Frame Editor. In the Graphics Editor make this active object a solid circle of a bright color. This circle should be large enough to fill a Backdrop square. Move the playing piece onto the starting square.

Part 2: The Spinner

The spinner will produce a random number that determines the amount of spaces the player can move. It will work like this: The user will click on a counter object, which will cause the counter on the playing board to quickly "spin" for 1 second, finally stopping on a random number between one and three. We will do this with three events.

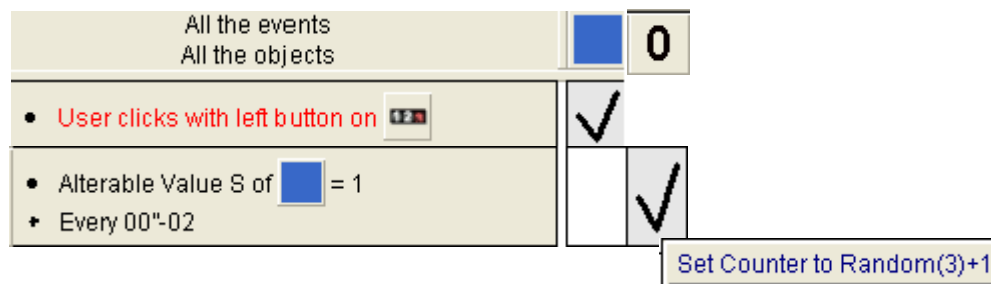
3.1 In the Frame Editor, insert an active object (any object) outside of the playing area. This object has no purpose other than to hold alterable values. We will use a blue square in this tutorial. Next, put a counter (#1) onto the playing area. This will be your spinner.

3.2 As stated, we will use three events to construct the spinner. Although most experienced MMF2 users will easily follow the logic used here, I'll share a brief explanation at the conclusion of what the three events do. In the Event Editor, right click on New Condition and then right click the "Mouse Pointer and Keyboard" icon. From the options, select Mouse > User Clicks on an Object > Counter.

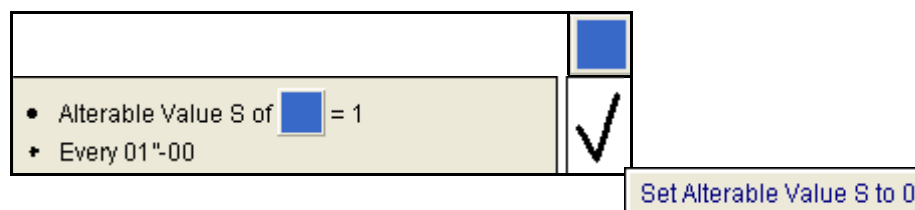
Next, in the grid box of this same condition, right click under the blue square and click on *Alterable values* > *Set*. Set alterable value S to 1. (Note: I've used "S" here, as this helps to identify that the value belongs to the spinner.) Also, enter this: Set Alterable Value X to 1. This value will be used to control the piece movement, as explained later.



3.3 This will be our second event for constructing the spinner. Click on New Condition, followed by right clicking on the blue square from the list of icons showing. From the pop up menu, click on *Alterable values > Compare to one of the alterable values*. In the expression box, enter 1. Click OK. We also want to insert another condition in this same event, so right click, click on *insert*, click on the *Timer* icon (the clock) and choose "Every." In the box that appears, in the bottom box (the 1/100 box) replace the 0 with the number 2. All other boxes should be 0. Click OK. Next, under the counter icon (the one used for the spinner), right click in the grid box of this event and choose *Set counter*. In the Expression Editor, enter this expression: $\text{Random}(3)+1$ and click OK.



3.4 Our third event, is this: Click on New Condition and repeat the entries of 3.3, with the following exception: After clicking on the *Timer* icon and choosing "Every", replace the number 0 in the second(s) box with a 1. All other boxes should be 0. Click OK. Now, under the blue square in the grid box, insert "Set alterable value S to 0".



Explanation The three events of the spinner work this way: The first events says, "whenever I click on the counter (spinner) change the value of the blue square from 0 to 1". The second event says, if the value of this blue square is 1, then every fraction of a second change the counter to any number between one and three. (Note: +1 assures that the counter never stops on 0). The third event stops the spinning by saying, if the value of the blue square is 1, then after 1 second change the value of blue square to 0.

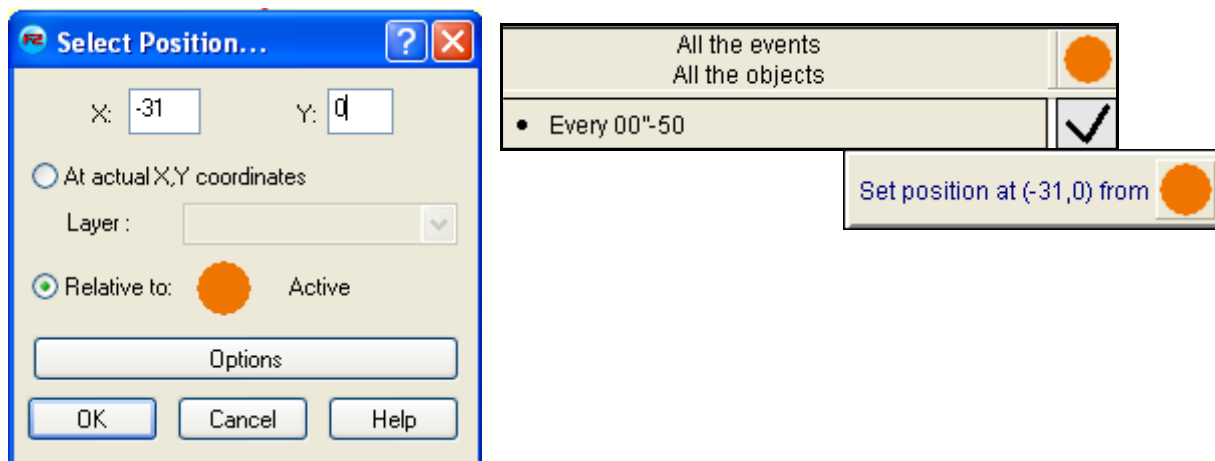
Part 3: Playing Piece Movement

Make a Start of Frame event and insert the following actions, as shown below. The value M stands for "movement".

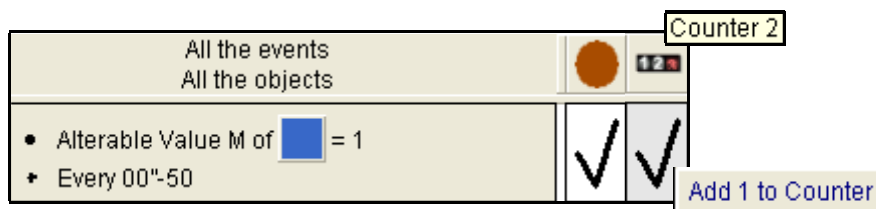


The playing piece is programmed to move automatically. It works like this. The user clicks on the spinner and the piece then moves the number spun from one square to the next. We will first set up the movement of the playing piece in the event editor. Then, we will add a few alterable values to control it. As you will see, the spinner, piece movement and the questions generated all work together.

4.1 In the event editor, click New Condition > The Timer (clock icon) > Every > 00"-50. Next, under the playing piece in this same condition, set the position of the piece, as shown here. If you were to run the frame now, the playing piece would move from right to left every half second. Now let's control it with a few alterable values.



4.2 The playing piece needs to move around the board on the squares. We will use a counter and several alterable values to control this. Add another counter (#2) in the frame editor. Place it in a prominent place on the playing area. This counter is used to record and display the number of spaces the piece has moved. Now go to the event editor. Here, we will insert an alterable value into the event shown above. This value will later change four times, one value for each direction. Also, add the action, as illustrated, under counter (#2). It will look like this:



4.3 The below illustration shows all the events needed to make the playing piece move and stop around as it travels around the board. All four events follow the same logic. Each event is set for the playing piece to move in one particular direction.

Alterable value X = 1: This value keeps the piece from moving prematurely when the game begins. When the player clicks on the spinner, this values changes from 0 to 1.

Alterable value S = 0: This keeps the spinner from being used for movement until the current move is completed.

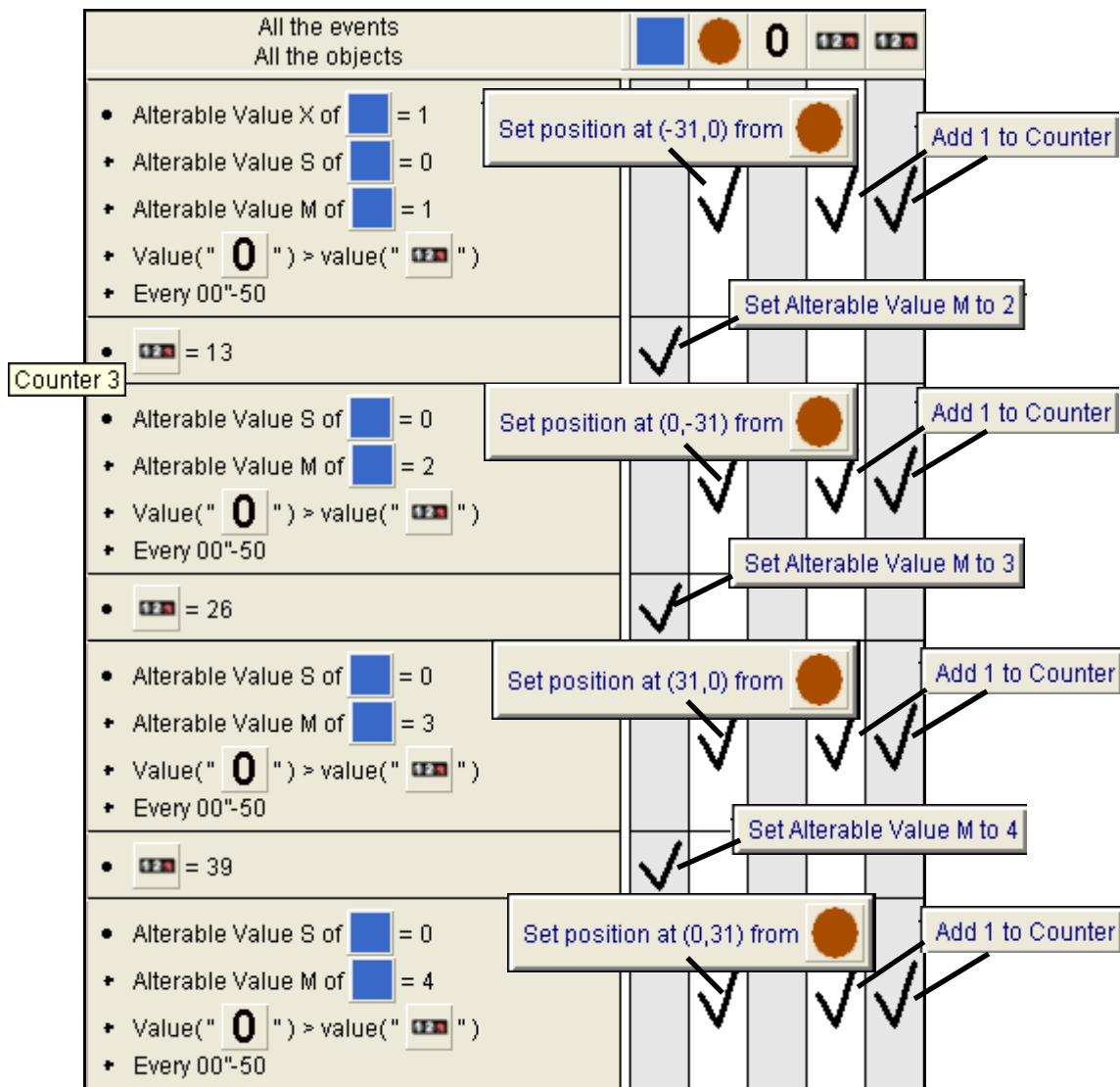
Alterable value M = 1: This is used to restrain the movement from right to left, as it travels along the bottom row or squares.

The value of counter 1 must be greater than the value of counter 2. It's for this reason, that one of the counters (#1) has been set to 1 at the start of the frame. Later, when both counters are of equal value, the question will appear.

Every 00"-50 determines the speed the the playing moves from square to square.

When counter #4 has reached 13, the value of the blue square changes to 2. This is the condition needed to move the piece upward.

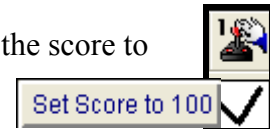
Counter 2 advances by one to display the number of moves. When the spinner is clicked, it's reset to 0. **Counter 3** records the number of moves and is instrumental in which direction the



Part 4: The Question Generator

4.1 Insert another counter (counter #4) into the Frame Editor, out of view of the playing area. This counter will be used to generate a random question. It will work this way: when the player clicks on the spinner, this fourth counter will produce a random number. When the playing piece has completed its moves, both counter 1 (the spinner) and counter 2 (moves) will equal one another, thus generating an event that produces a question on the board. The number arrived by the fourth counter will determine which question appears.

You will also need to insert a score counter onto the screen, as well. Set the score to start at 100 and place it on the playing board for easy viewing. The board set up is illustrated under the section, *How the Game Works*.



4.2 You will also need to insert an object called Random Pool. This tool will allow you to generate questions with no repetition. This ingenious (my opinion) add-on tool was developed by Mokhtar M. Khorshid and Michael Saad. It's available from this site: <http://www.clickteam.com/eng/bonus.php>

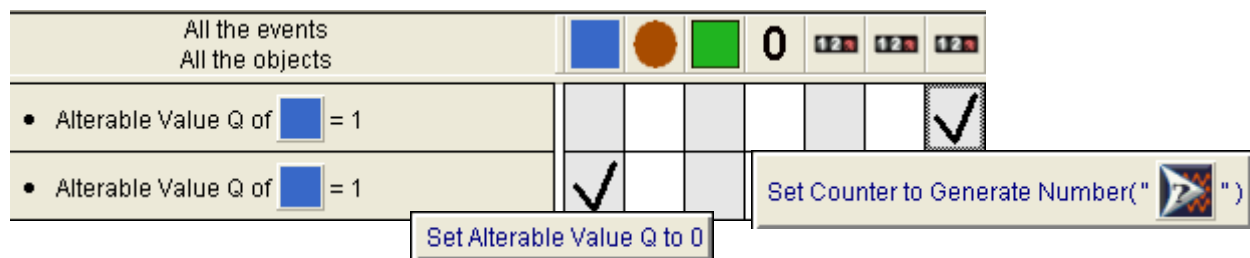
Insert the Random Pool object onto the frame. Set it up this way: Double click on the object and make the entries, as shown. The maximum should equal the number of questions you plan on using, 100 in this case. Go to the event editor and in the Start of Frame event right click under the Random Pool object and click on Pool Setup > Shuffle.

Minimum	<input type="text" value="1"/>
Maximum	<input type="text" value="100"/>
Seed	<input type="text" value="0"/>

4.3 Next, we want counter 4 to generate a random number when the player clicks on the spinner. We will set it up this way: Insert *set alterable value Q to 1*, as shown below. The "Q" stands for "questions".

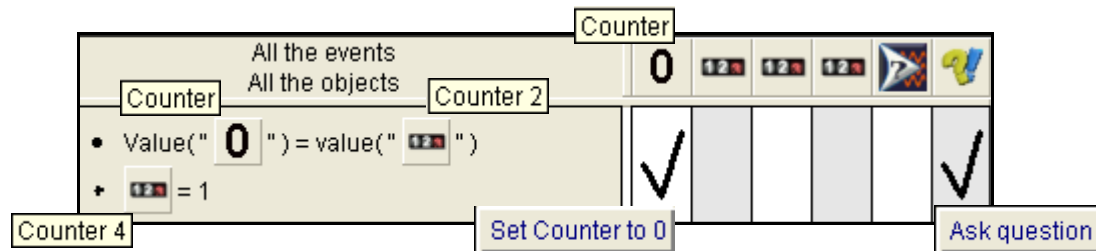


4.4 Make two new events, as shown below:



These two events act as a "toggle" switch. They work like this: When the player clicks on the spinner, value Q is set to 1. If value Q equals 1, then generate a random number (set counter > Retrieve Data From an Object > Random Pool > Get new generated number > OK) and set value Q to 0.

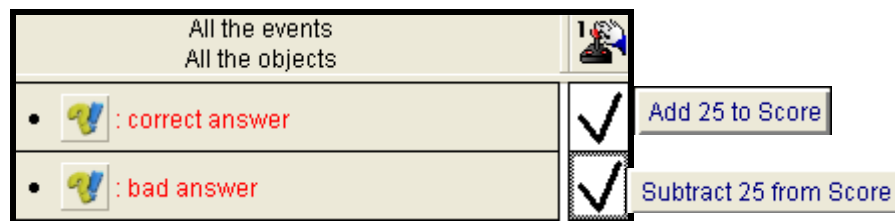
4.5 The question is made to appear from these events/conditions:



To make this event click on New condition > Special (the two computer monitors) > Compare two general values. Highlight the 0 in the upper box, then click on Retrieve data from an object > Counter 1 > Current value. Now highlight the 0 in the lower box and repeat the steps, except that you click on counter 2.

It all works like this: When counter 1 equals the same value as counter 2, then ask the question. Counter 4 determines which question will be asked. In this case, it's question #1. Counter 1 is also reset to 0 so that both counters will no longer be equal.

Now make a new event that will add or subtract points from the score: New Condition > Question and Answer > Is the answer correct? Make another event for is the answer false?

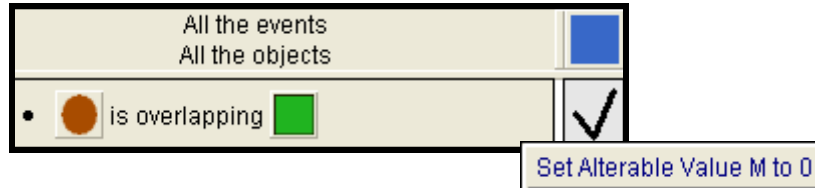


Repeat these events with another question, making counter 4 to equal 2. Continue this with additional questions, always advancing counter 4 to equal another number/question.

4.6 I would recommend around 100 questions to minimize the risk of running out of questions during game play. These questions can be study material to learn by or simply trivial questions taken from internet sites that host trivial games. These questions are placed off to one side of the playing area. Later, they will appear in the middle of the playing board when prompted by the user.

Part 5: Odds and Ends

5.1 To stop the playing piece once it has moved completely around the board, make the following event:



5.2 Add a couple of more counters to record the number of correct and incorrect answers.

5.3 Add sound effects for movement, questions appearing, correct answers, incorrect answers, etc. Also, add text boxes, title frames and a festive finale to complete the game.

Hope you find this tutorial of value. Best wishes, Whiskers

All Rights Reserved Copyrighted 2008 S. D. Threadgill