

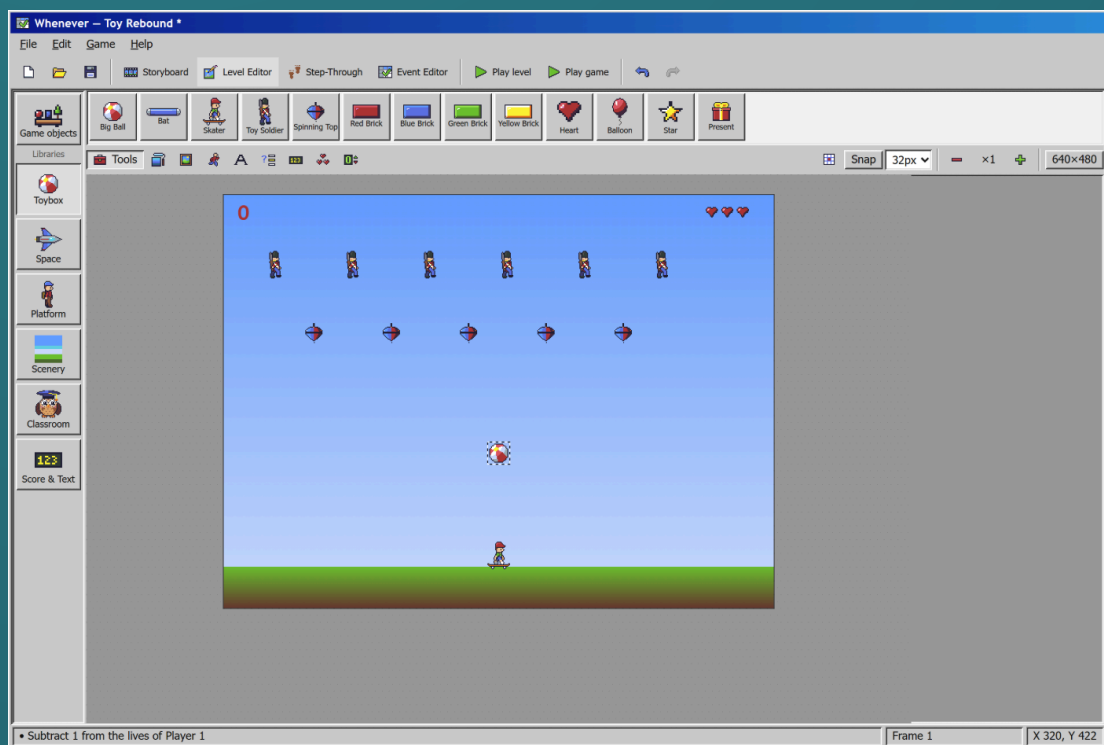


USER'S GUIDE

Whenever

User's Guide

Whenever lets you make games by placing objects and ticking boxes in a grid of rules. There is no code to type. This guide builds your first game with you, then covers each part of the program in turn.



Whenever the ball collides with a brick, destroy the brick, bounce the ball and add 50 to the score.

01 Welcome

What you need · The first time you open it · The screen at a glance · The four editors · Handy hints

02 Your first game: Toy Rebound

Step 1: Start a new game · Step 2: Lay out the scenery · Step 3: Let the ball fall through the floor · Step 4: Add the toys · Step 5: Set the movements · Step 6: Meet the events Whenever made for you · Step 7: Decide what happens, by playing · Step 8: Lose a life at the bottom · Step 9: Winning and losing · Step 10: Play it · Ideas to try next

03 The Storyboard

Frame cards · Right-click a frame · Moving from frame to frame · Frame set-up · Game set-up

04 The Level Editor

Libraries and the shelf · Placing objects · Selecting, moving and resizing · Right-click an object · The toolbar · The holding area · Keys

05 Objects and movements

The eight kinds · Properties · Alterable values · Movements · Directions · Platform games · Drawing a path

06 Events

Reading the grid · Moments and states · Making an event · Settings boxes · Which copy does it act on? · Changing events · Sentence view · How a game runs

07 The Step-Through Editor

What makes it stop · Choosing what happens · The other buttons

08 Pictures and animations

The Animation editor · Frames and directions · The Picture editor · Hot spot and action point

09 Sounds

Choosing a sound · Sound actions

10 Expressions

The expression calculator · Writing expressions · Functions about objects · Other functions

11 Beyond 1994

Scrolling levels · Groups of events · Families · Global values · Touch joystick · Opening a web page

12 Finding bugs

The controls · Events that light up · Watch · A way to hunt bugs · Common bugs

13 Playing and sharing

The play window · Saving your work · Stand-alone games · Putting games online



Welcome

Whenever is a game maker that runs in a web browser. You make a game in three moves: put objects on a play area, choose how they move, and tick boxes in a grid of rules called events. It follows Klik & Play, the game maker Maxis and Clickteam published in 1994.

Every rule reads like a sentence: **Whenever** the ball collides with a brick, destroy the brick, bounce the ball and add 50 to the score. That is where the name comes from.

What you need

- A recent web browser (Chrome, Edge, Firefox or Safari) on a computer. A mouse with a right button helps, because many menus open with a right-click.
- Nothing to install and no account. Whenever saves your game in the browser as you work.
- To keep a copy, or move a game to another computer, use **File** ► **Save game**. It downloads a `.whenever` file you can load again.

The first time you open it

A welcome box offers three ways in. **Quick tour** walks round the program in five steps. **Open the example game** loads Toy Rebound, the game you build in the next chapter. **Start a new game** gives you an empty game.

Teachers will also find nine starter games under **File** ► **Open a lesson starter**. They go with the lesson plans in Whenever in the Classroom. A link ending in `?lesson=3` opens lesson 3's starter directly.

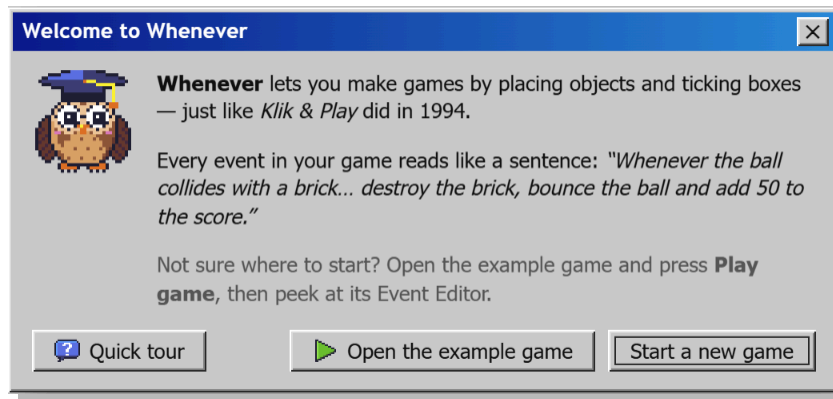


FIG 1.1 The welcome box, with Quick tour, Open the example game and Start a new game

■ The screen at a glance

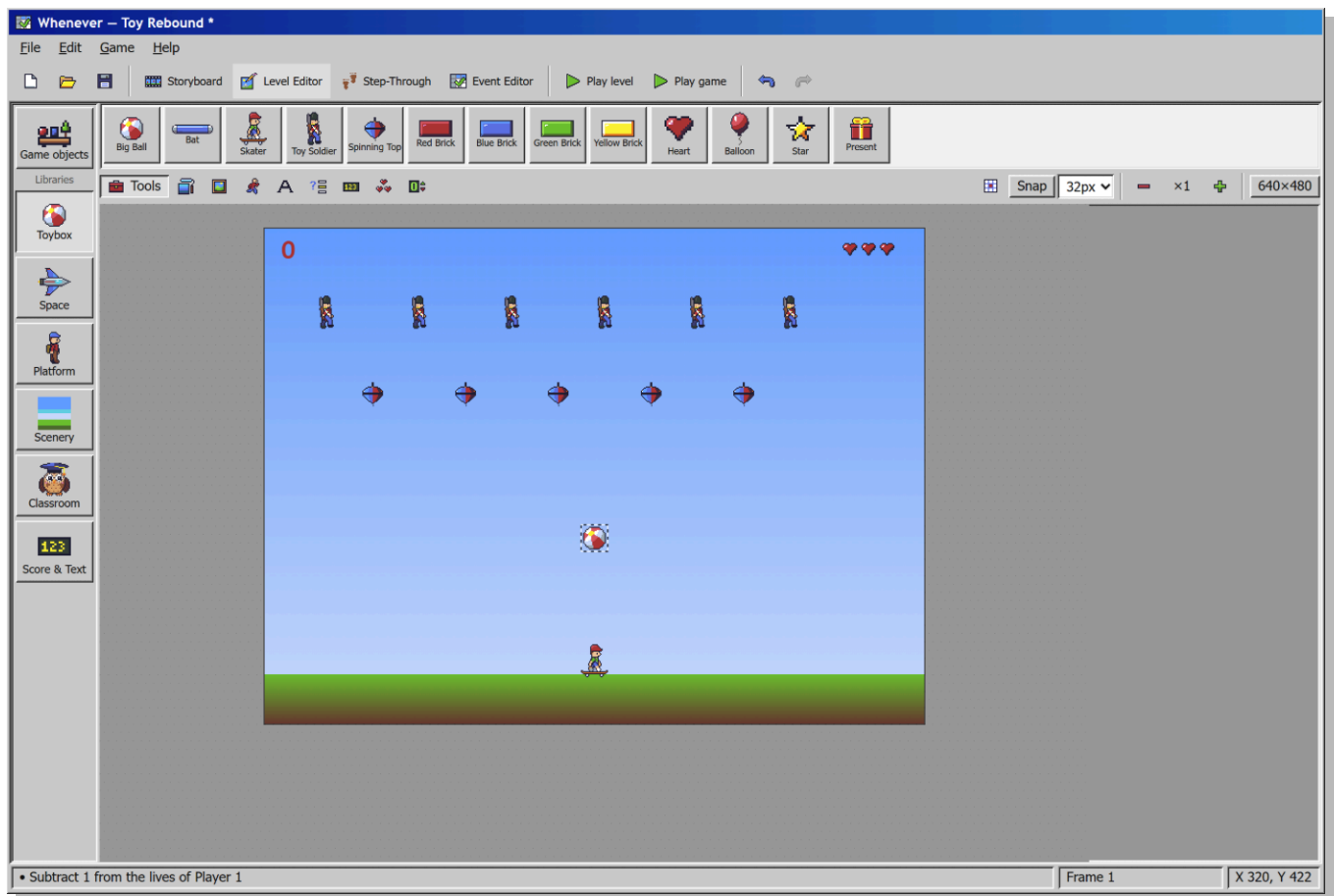


FIG 1.2 The Whenever window showing the Level Editor with Toy Rebound open

Part	What it does
Title bar	The game's title and file name. A  means there are changes you haven't saved to a file.
Menu bar	File, Edit, Game and Help .
Toolbar	New, Load and Save; buttons for the four editors; Play level () and Play game (); Undo and Redo.
Editor area	Shows one of the four editors.
Status bar	A hint for whatever is under the pointer, the name of the current frame, and the pointer's position on the play area.

The four editors

Editor	Use it to	Shortcut
Storyboard Editor	See the frames of your game (title screen, levels, game over) and put them in order.	
Level Editor	Place, move and resize the objects in a frame.	
Step-Through Editor	Play the frame and decide what happens as things happen.	
Event Editor	See and change every rule in a grid.	

Handy hints

Rest the pointer on any button or box for a moment and a yellow hint explains it. The status bar shows the same hint. To stop the pop-ups, untick **Edit** ► **Preferences** ► **Show handy hints**. The same Preferences menu switches between light and dark themes.

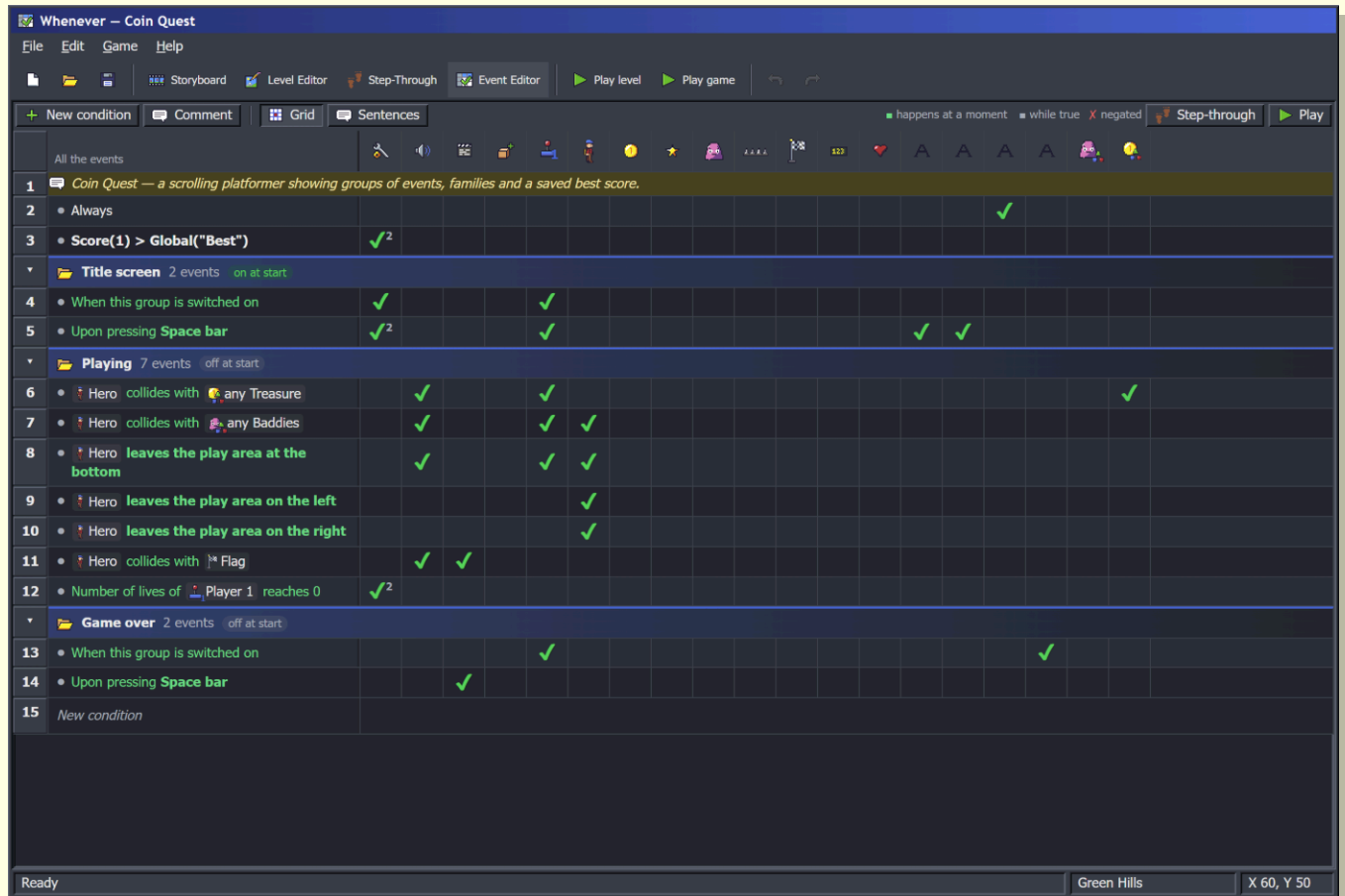


FIG 1.3 The Event Editor in the dark theme



Your first game: Toy Rebound

In about 30 minutes you will build Toy Rebound: bounce a beach ball off a skater's head to knock over toy soldiers and spinning tops. It is the demo game from chapter 1 of the 1994 Klik & Play manual. A finished copy is in **File** ▶ **Open an example** ▶ **Toy Rebound** if you want to peek.

■ Step 1: Start a new game

- 1 Choose **File** ▶ **New game**. The Storyboard shows one empty frame, *Frame 1*.
- 2 Choose **Game** ▶ **Game set-up...**, type *Toy Rebound* as the title, and click **OK**.
- 3 Double-click Frame 1 to open it in the Level Editor.

■ Step 2: Lay out the scenery

The column on the left lists the libraries. Click one and its objects appear on the shelf along the top.

- 1 Click **Scenery**.
- 2 Click **Blue Sky** on the shelf, then click the top-left corner of the play area (the white rectangle). The sky fills the play area.
- 3 Click **Ground**, then click near the bottom-left corner so it lies along the bottom.
- 4 If something is a little out of place, drag it, or nudge it with the arrow keys.

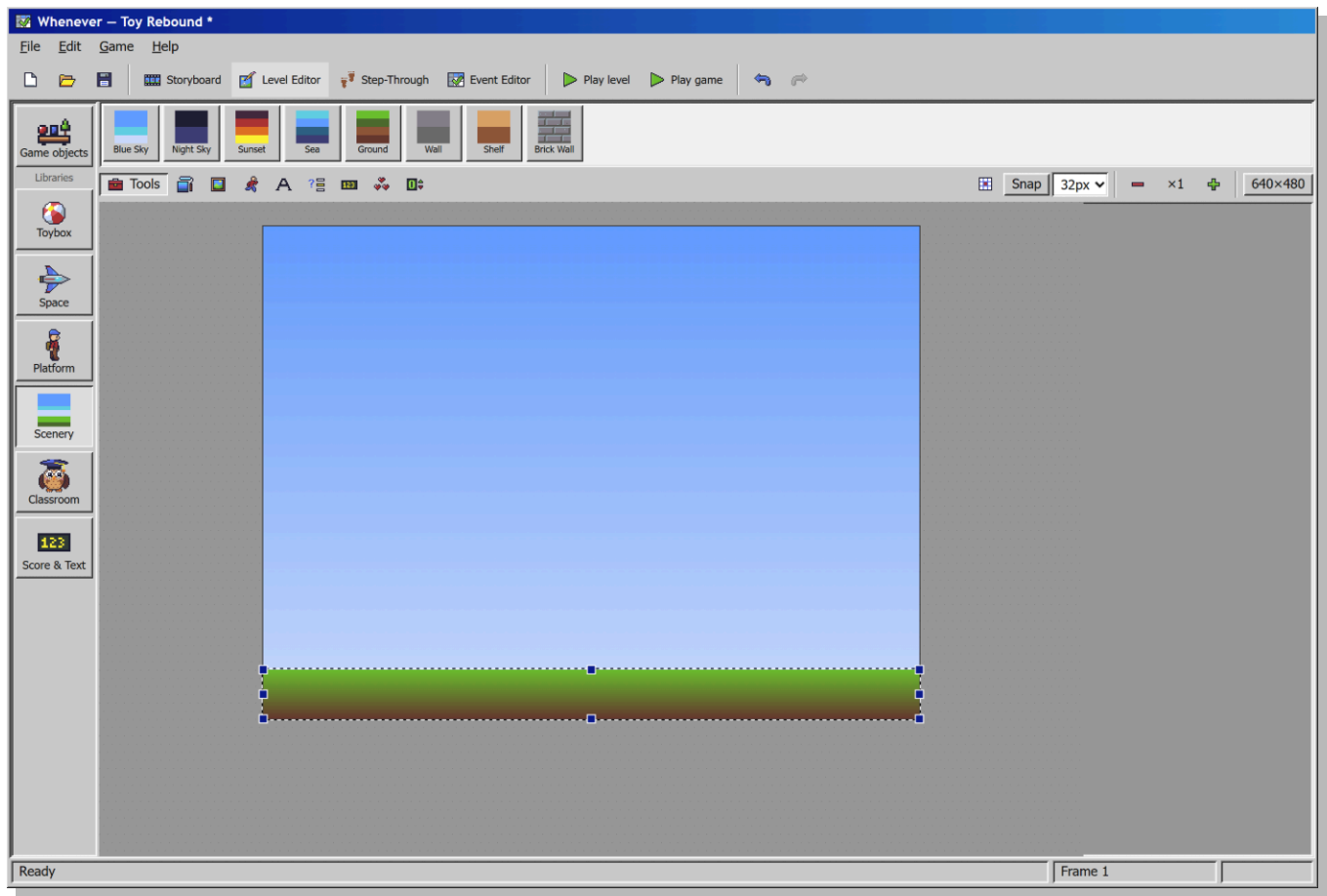


FIG 2.1 The Level Editor with the Blue Sky and Ground placed and the Ground selected

■ Step 3: Let the ball fall through the floor

Ground is solid, so the ball would bounce off it and never fall out of the game. Right-click the Ground and choose **Obstacle** ► **No**. Now it is just scenery.

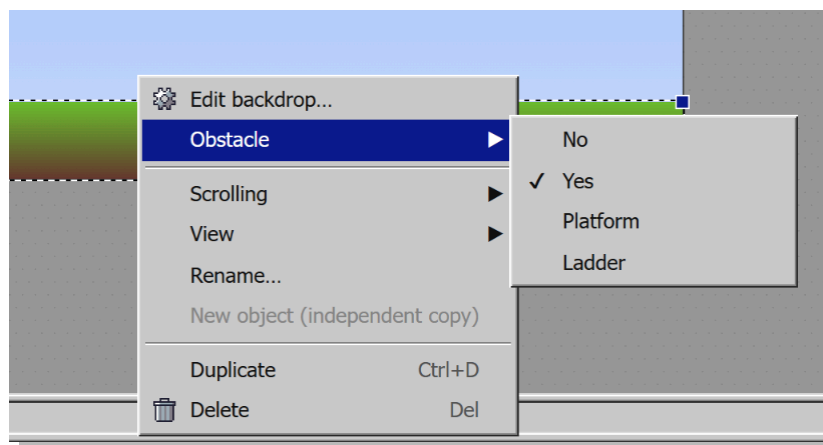


FIG 2.2 The Ground's right-click menu, with Obstacle open and No, Yes, Platform and Ladder listed

■ Step 4: Add the toys

- 1 Click the **Toybox** library.
- 2 Click **Toy Soldier** on the shelf. Right-click six times in a row across the top of the play area: each right-click drops a soldier and keeps going. Press **Esc** to stop.
- 3 Do the same with **Spinning Top**: a row of five, a little lower.
- 4 Click **Skater** and click once in the middle of the ground. A left-click drops one object and stops.
- 5 Click **Big Ball** and drop it in the middle of the play area.
- 6 Click the **Score & Text** library. Put **Score** in the top-left corner and **Lives** in the top-right corner.

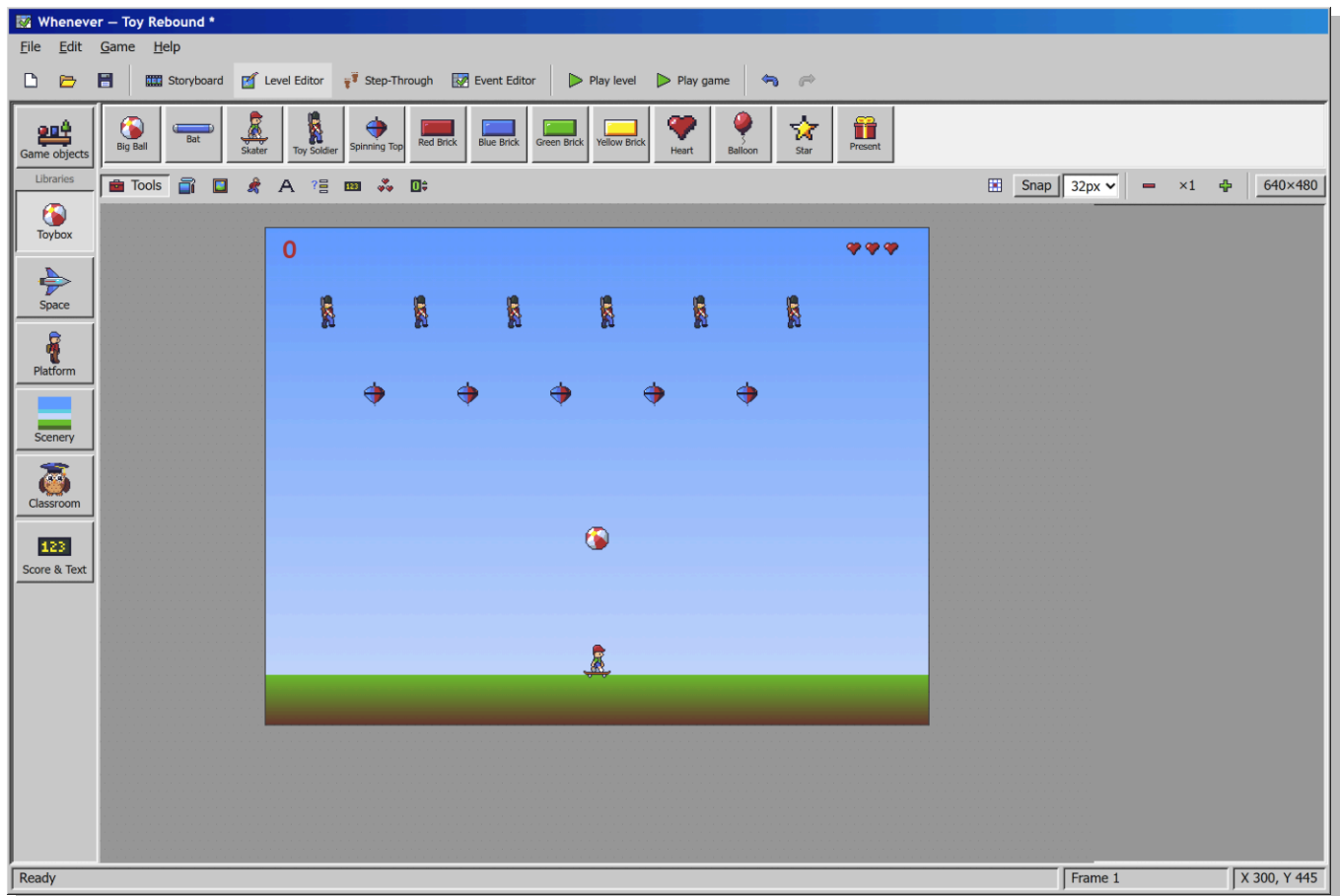


FIG 2.3 The finished layout: soldiers, spinning tops, the skater, the ball, the score and the lives

■ Step 5: Set the movements

Library objects already know how to move. Big Ball bounces around by itself. Skater moves with the arrow keys in eight directions. Toy Soldiers march back and forth. Our skater is the bat, so he should only go left and right.

- 1 Right-click the Skater and choose **Edit movement....**
- 2 Under **Possible directions**, click **None**, then click the squares for left and right.

Click **Try movement** and press the arrow keys to test it. Press **Esc** to stop, then click **OK**.

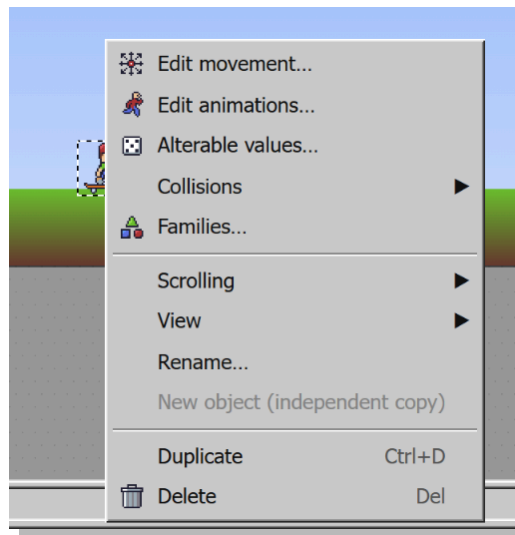


FIG 2.4 The Skater's right-click menu, with Edit movement at the top

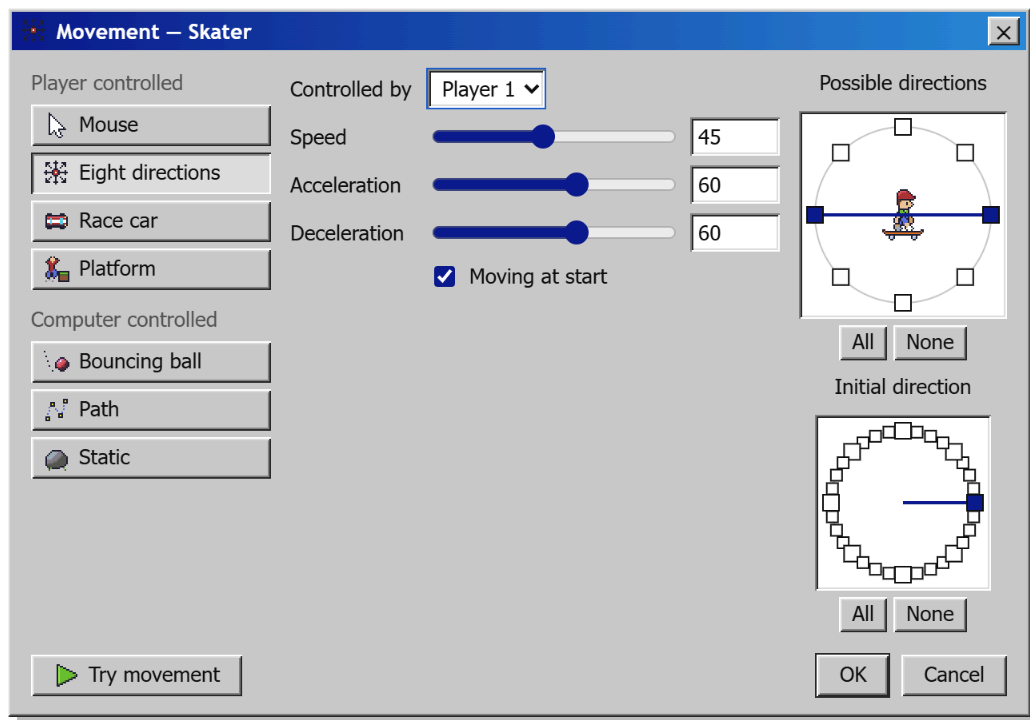


FIG 2.5 The movement dialog for the Skater: Eight directions, with only left and right ticked

Right-click Big Ball and open its movement too. It is a **Bouncing ball** that starts going up-left or up-right. The **Bounce randomiser** adds a little wobble so the ball doesn't get stuck in a loop. Leave it as it is.

■ Step 6: Meet the events Whenever made for you

Click **Event Editor** on the toolbar. There are already four events. The first time you place a moving object, Whenever adds a few sensible rules for it:

- Whenever the Skater leaves the play area, or hits the backdrop, it stops.
- Whenever the Big Ball leaves the play area, or hits the backdrop, it bounces.

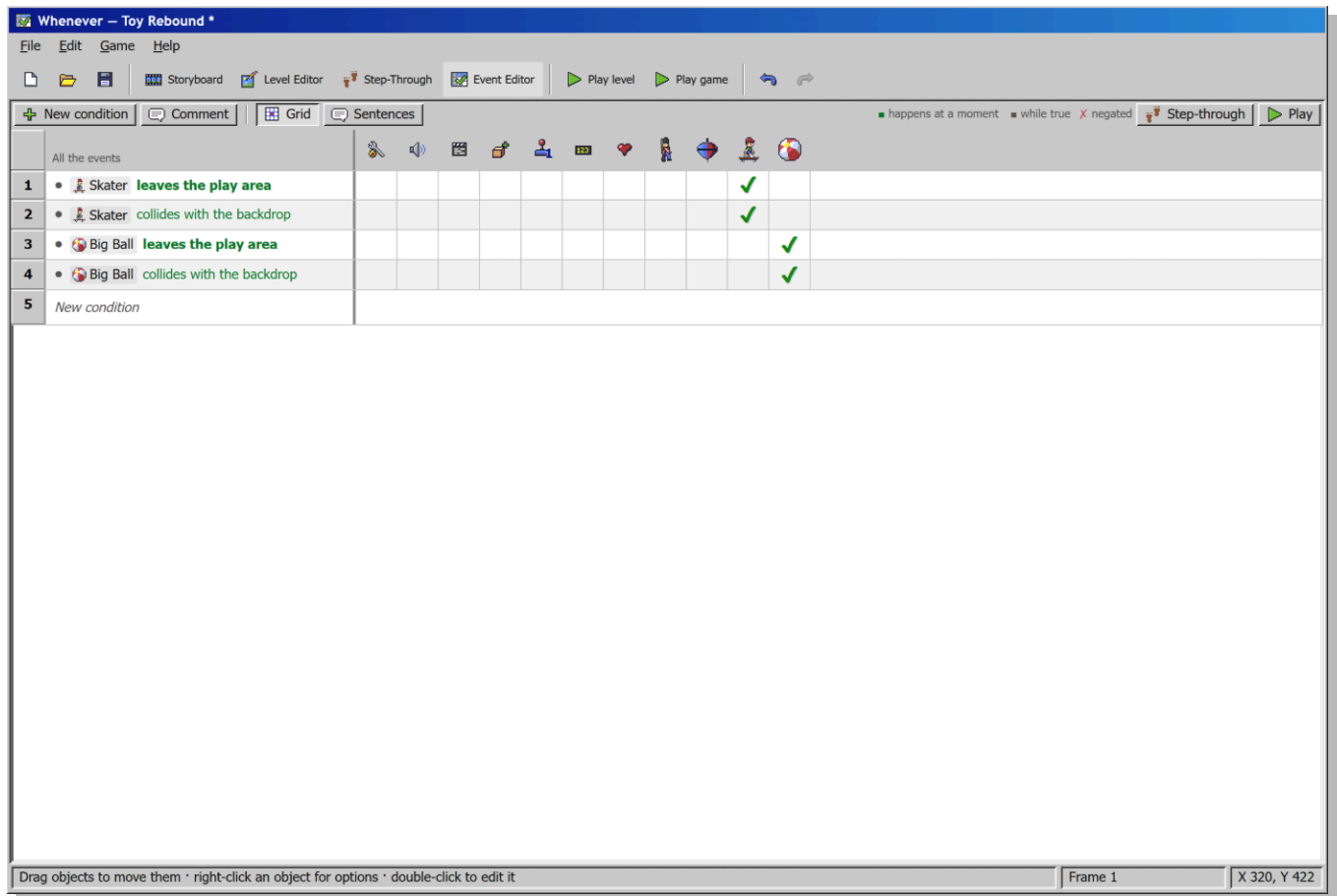


FIG 2.6 The Event Editor with the four automatic events

Each row is one event. The conditions are on the left. A tick in a column means "do something to this object". Green conditions happen at a moment, like a collision.

Press **F7** to play the level. The ball bounces, but it passes straight through the toys. Close the game window with **Esc**.

■ Step 7: Decide what happens, by playing

The Step-Through Editor plays your game and stops whenever something happens that your events don't handle yet.

- 1 Click **Step-Through** on the toolbar. The game starts on the left.

2

Soon the ball hits a toy and the game stops. The panel says *Spinning Top collides with Big Ball*. (Your game may stop on a Toy Soldier first. The steps are the same.)

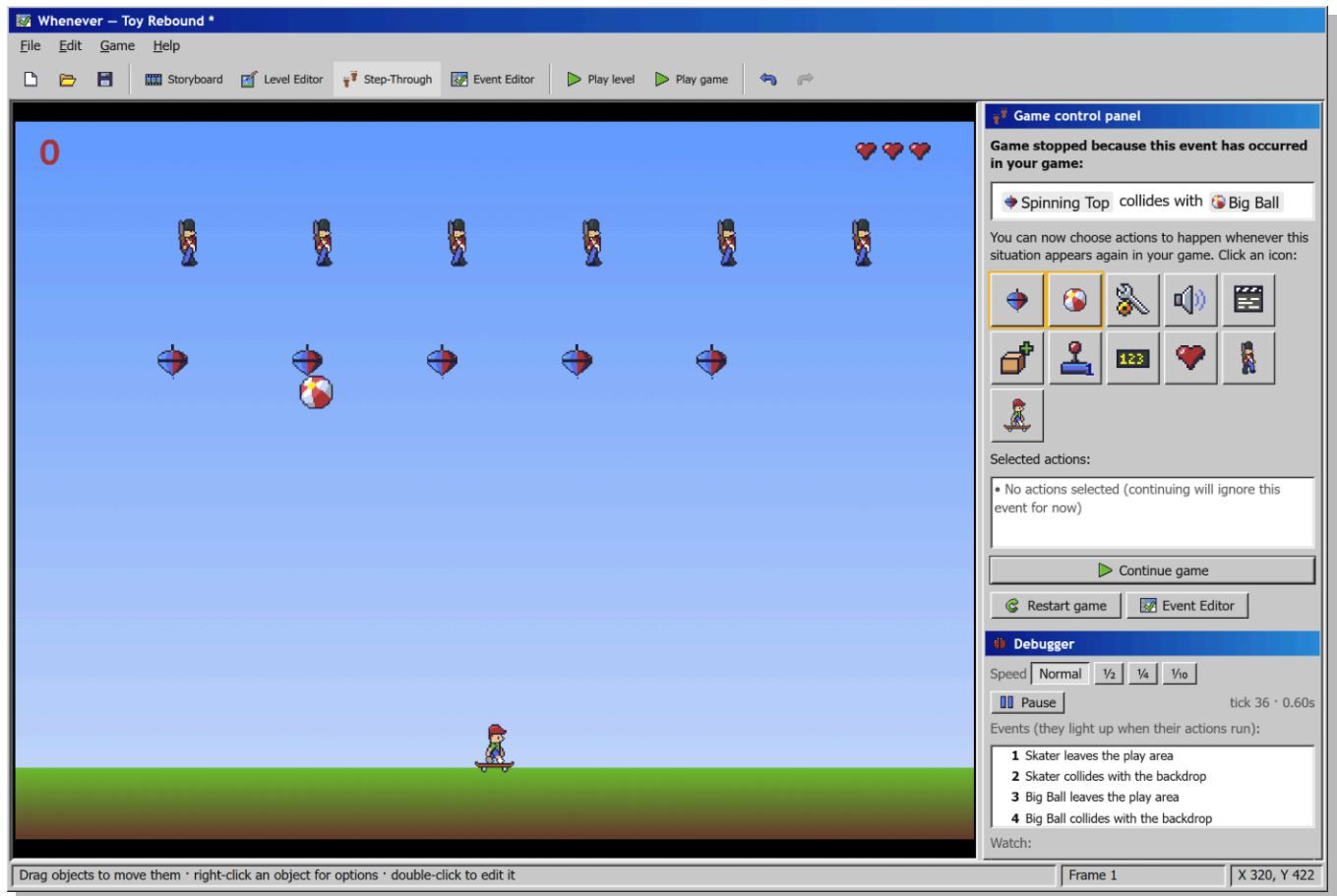


FIG 2.7 The Step-Through Editor stopped because the Spinning Top collided with the Big Ball

3

The objects involved are outlined. Click the **Spinning Top** icon and choose **Destroy**.

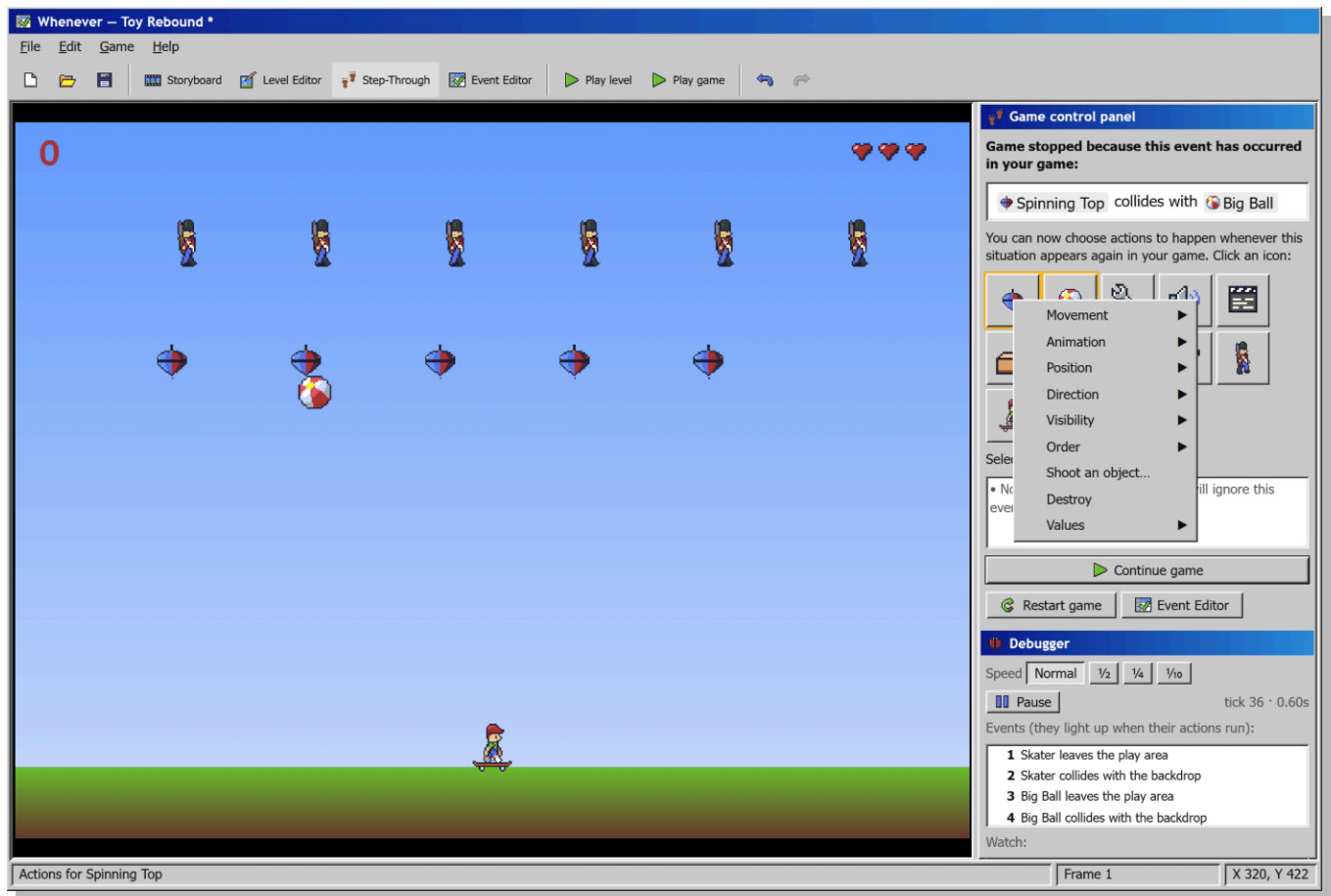


FIG 2.8 The Spinning Top's action menu in the Step-Through Editor

- 4 Click the **Big Ball** icon and choose **Movement** ▶ **Bounce**.
- 5 Click the **Player 1** icon (the joystick) and choose **Score** ▶ **Add to score**. Type 50 and click **OK**.

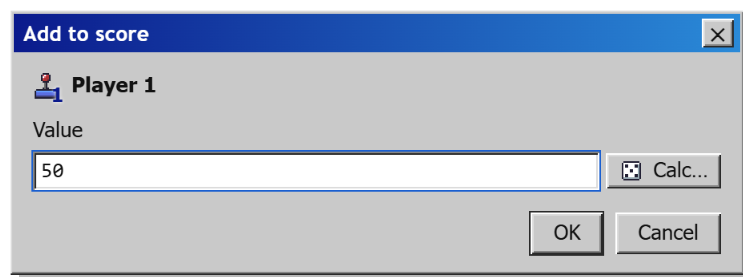


FIG 2.9 The Add to score box with 50 typed in

- 6 For a sound, click the **Sound** icon (the loudspeaker) and choose **Play sample**. Click **Choose...**, click **Tin clank** in the sound maker to hear it, click its **+** to add it, then **OK**.
- 7 Check the **Selected actions** list, then click **Continue game**.

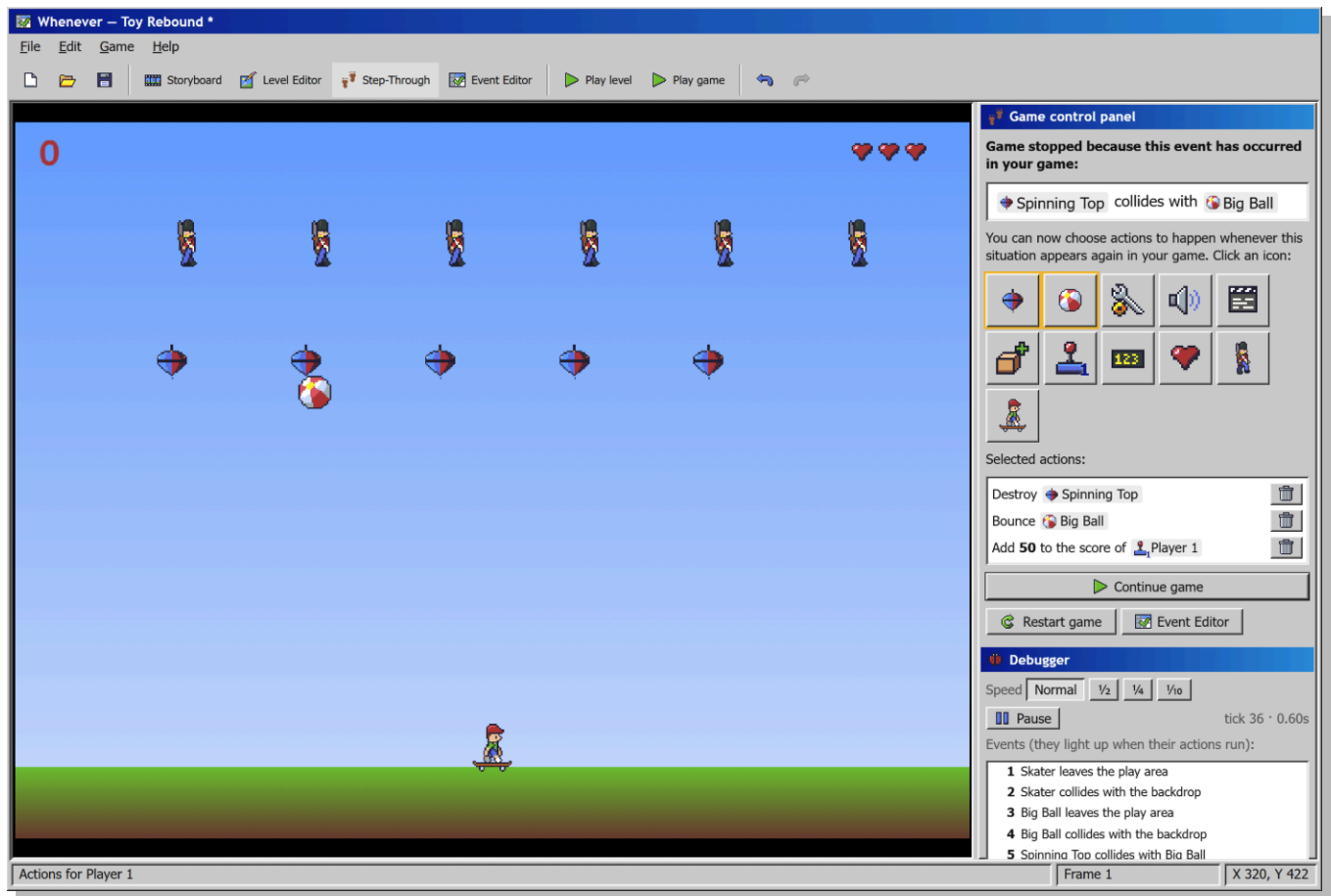


FIG 2.10 Three actions selected for the collision: destroy the top, bounce the ball, add 50 to the score

The game carries on until the next new situation. Give the Toy Soldier the same four actions. When the ball first lands on the Skater, choose **Movement** ► **Bounce** for the Big Ball, and the **Bounce / boing** sound if you like.

■ Step 8: Lose a life at the bottom

The ball can't fall out yet, because the automatic event bounces it off every edge. Add a second rule for the bottom edge. Both rules happen at the same moment: the first bounces the ball upward, and the new one puts it back in the middle.

- 1 Open the Event Editor and click **New condition**.
- 2 Click the **Big Ball** icon and choose **Position** ► **Test position of the object**.

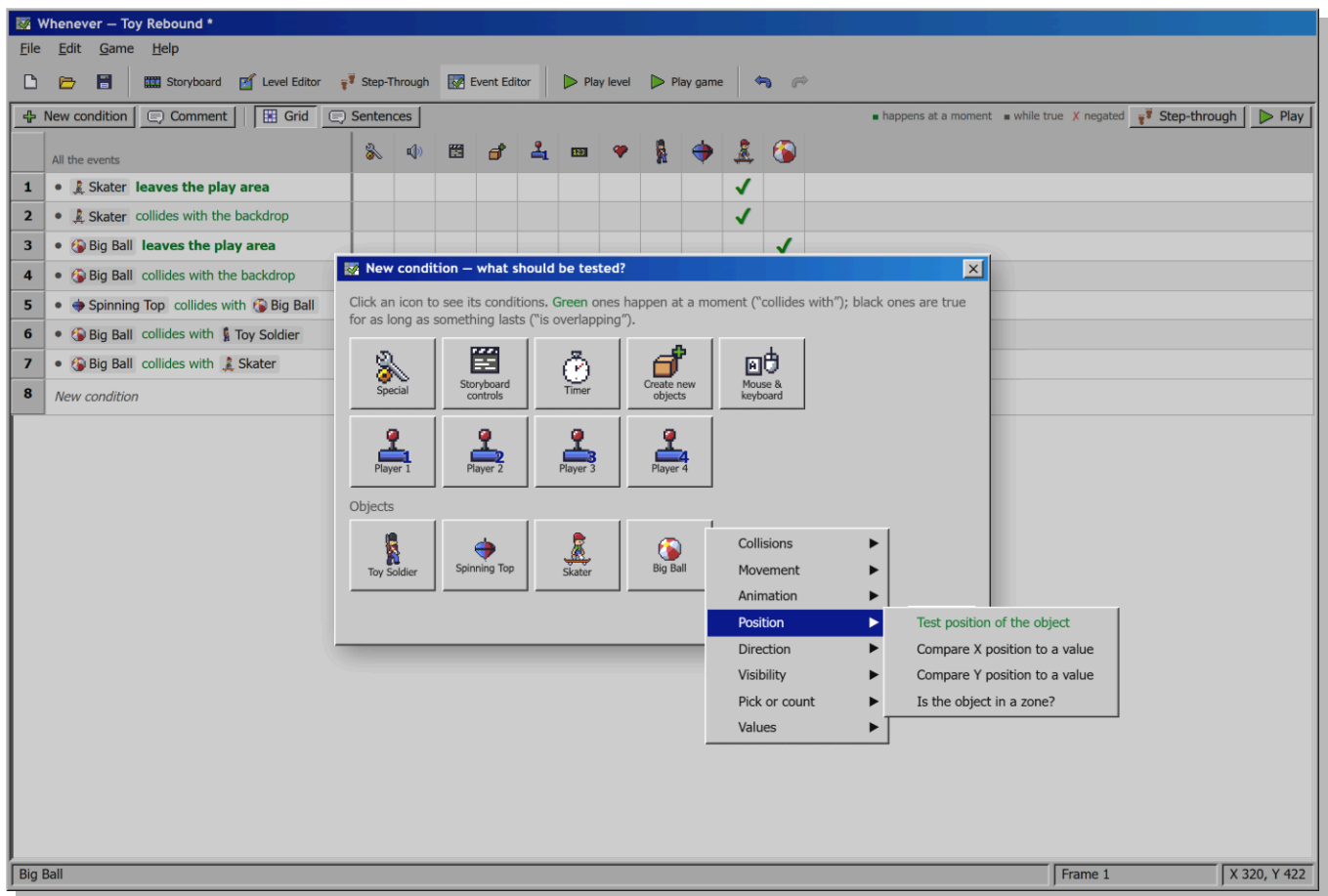


FIG 2.11 Choosing Position, then Test position of the object, for the Big Ball

Click the arrow pointing out of the bottom of the box. The box reads *Selected: the object leaves the play area at the bottom*. Click **OK**.

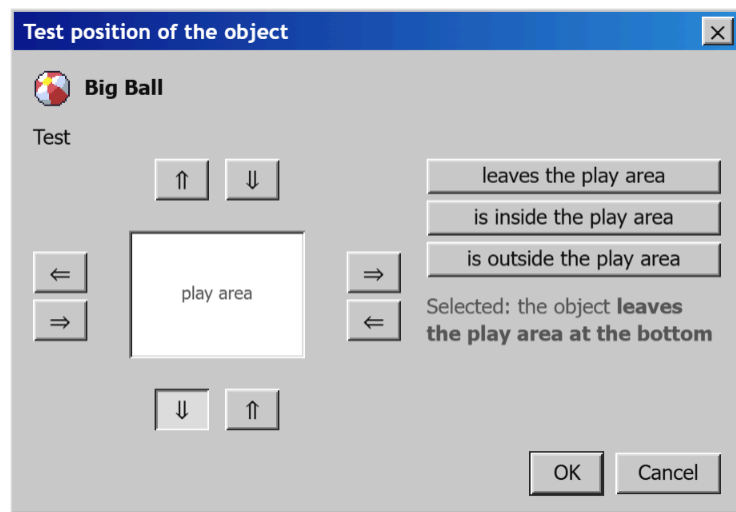


FIG 2.12 The Test position box with the bottom arrow chosen

- 4 A new row appears. Right-click its box under **Player 1** and choose **Number of lives** ► **Subtract from number of lives**. Type **1** and click **OK**.

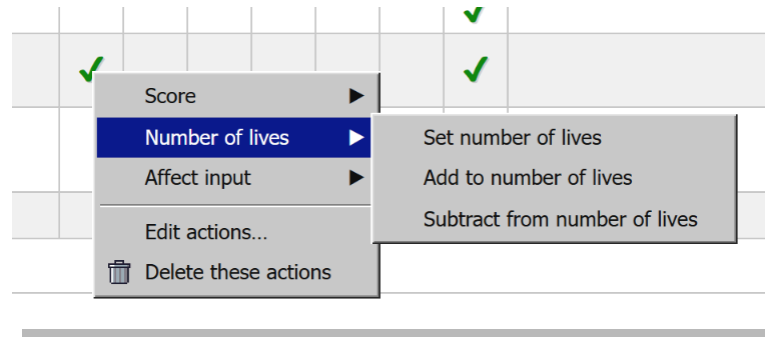


FIG 2.13 Right-clicking the Player 1 box to subtract a life

- 5 Right-click the box under **Big Ball** in the same row and choose **Position** ► **Select position....** Type X 320 and Y 300, or click the middle of the little picture. Click **OK**.
- 6 For a sound, right-click the box under **Sound** and choose **Play sample** again. Add **Ouch!** from the sound maker.

■ Step 9: Winning and losing

- 1 Click **New condition**, click **Toy Soldier**, and choose **Pick or count** ► **Have all been destroyed?**
- 2 Right-click the new event's number and choose **Add a new condition**. Click **Spinning Top** and choose the same thing. Now both must be true.
- 3 Right-click the box under **Storyboard controls** (the clapperboard) and choose **End the game**.
- 4 Click **New condition**, click **Player 1**, and choose **When number of lives reaches zero**. Give it **End the game** too.

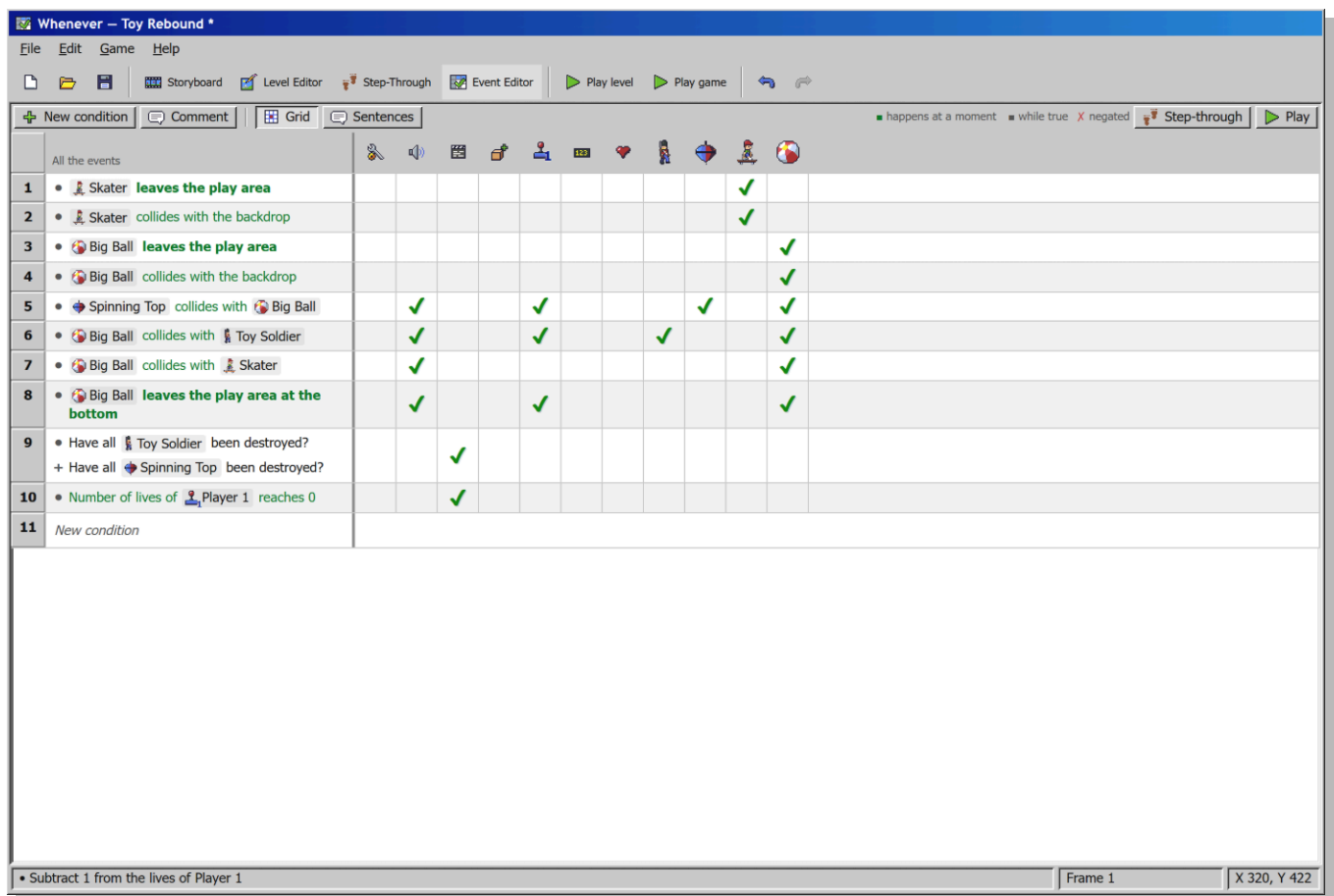


FIG 2.14 The finished events: ten rows, from the automatic ones to winning and losing

Click **Sentences** on the Event Editor's toolbar to read the whole game back as plain English.

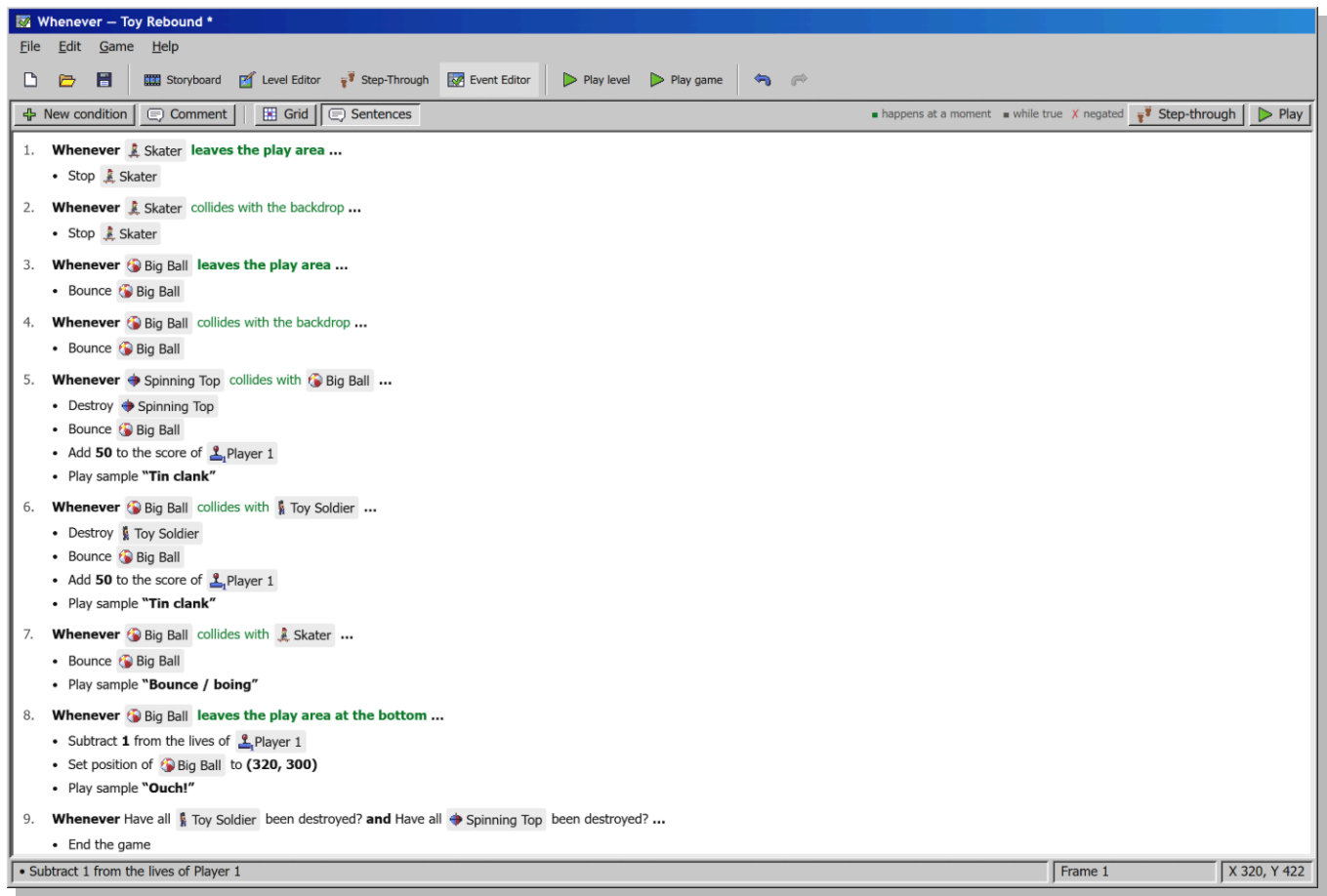


FIG 2.15 The same events in the sentence view

■ Step 10: Play it

Click **Play game** or press **F8**. Move the Skater with the left and right arrow keys. When you win or lose, a box offers **Play again**.

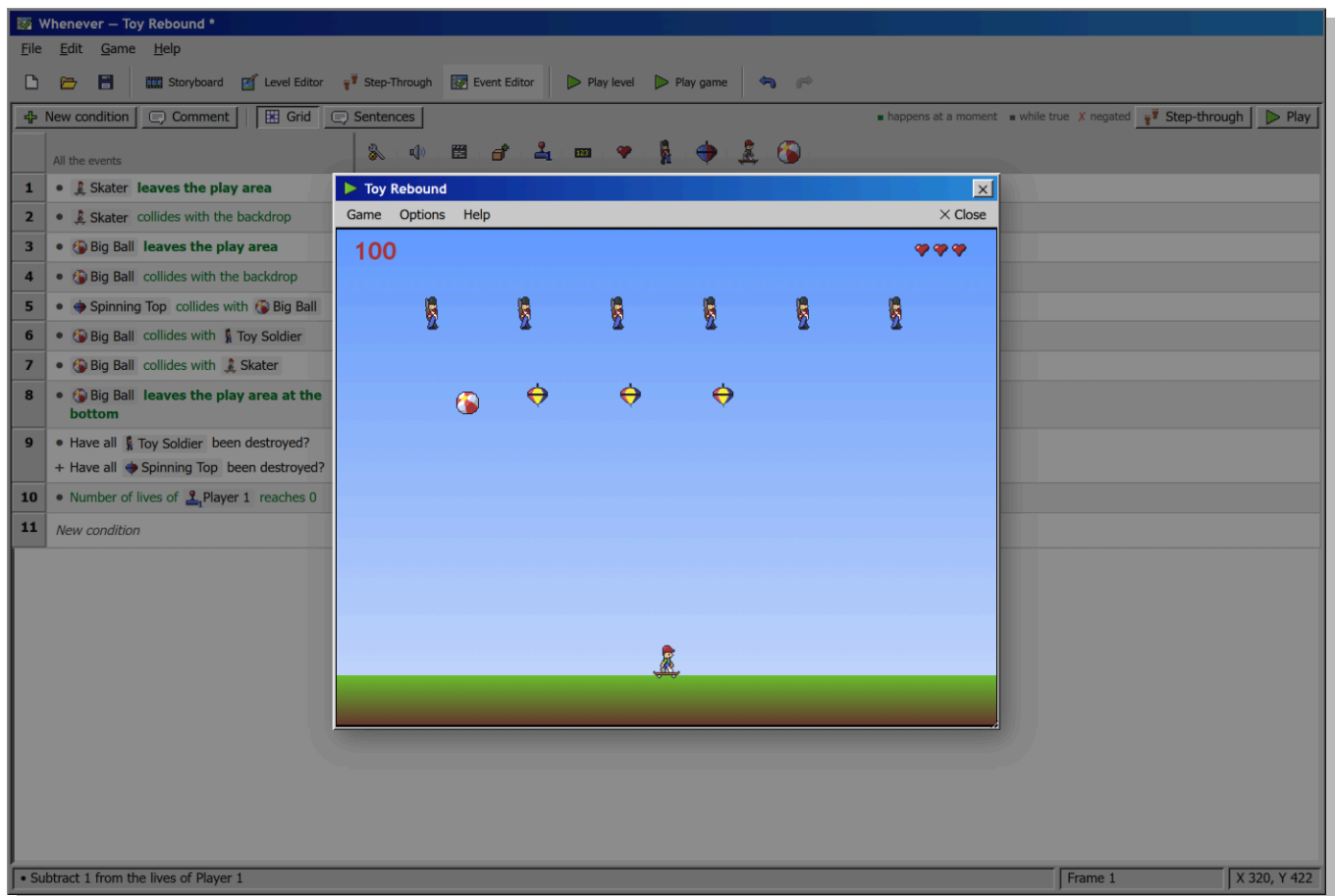


FIG 2.16 Toy Rebound running in the play window

Save your work with **File** ▶ **Save game** (`Ctrl+S`). To share it, see *Playing and sharing*.

■ Ideas to try next

- **A "Well done" screen.** On the Storyboard, click **New frame** and add a Text object that says *Well done!* Then change the winning event's action from **End the game** to **Next frame**.
- **Speed up.** Each time the ball hits the Skater, make it faster: Big Ball ▶ **Movement** ▶ **Set speed...** with `Speed("Big Ball") + 5`.
- **Compare.** Open the finished example and look at how its events differ from yours.



The Storyboard

A game is a row of frames: a title screen, the levels, a game-over screen. The Storyboard Editor shows them as cards in order. Play always starts at frame 1, and moves on only when an event says so.

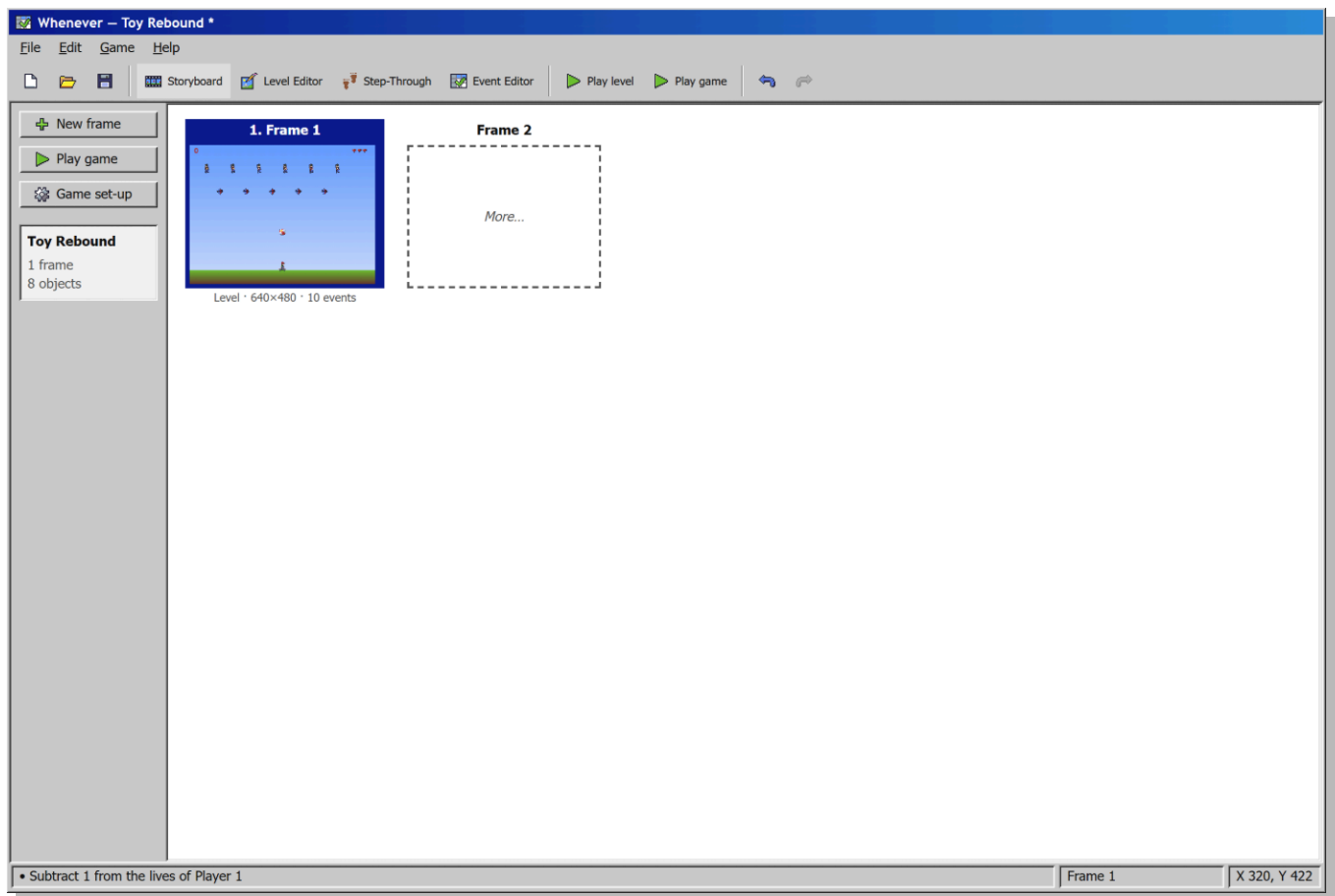


FIG 3.1 The Storyboard with two frames, Frame 1 and Well done

The column on the left has three buttons: **New frame**, **Play game** (**F8**) and **Game set-up**. Under them, a box shows the game's title, author, and how many frames and objects it has.

Frame cards

Each card shows the frame's number and name, a picture of it, and a line such as *Level · 640×480 · 10 events*.

- **Click** a card to select it. **Double-click** to open it in the Level Editor.
- **Drag** a card to change the order. A blue line shows where it will land.
- The last card, **More...**, adds a new frame at the end.
- **Keys:** ← and → select the previous or next frame, **Enter** opens it, **Delete** removes it.

■ Right-click a frame

Command	What it does
Frame set-up...	Name, size and background colour (see below).
Goto Level Editor / Step-Through Editor / Event Editor	Opens the frame in that editor.
Play level as finished (F7)	Plays just this frame.
Rename...	Gives the frame a new name.
Duplicate	Adds a copy called ... <i>copy</i> straight after it. The copy uses the same objects, so changing an object's pictures changes both.
Move left / Move right	Changes the order.
Delete	Removes the frame. If it has objects or events, Whenever asks first. A game always keeps at least one frame.

■ Moving from frame to frame

Frames change only when an action tells them to. These actions live under **Storyboard controls** (the clapperboard) in the Event Editor: **Next frame**, **Previous frame**, **Jump to frame...**, **Restart the current frame**, **Restart the game** and **End the game**. If the last frame says *Next frame*, the game ends.

■ Frame set-up

Open it from the frame's right-click menu, from **Game ▶ Frame set-up...**, or with the size button at the right of the Level Editor's toolbar.

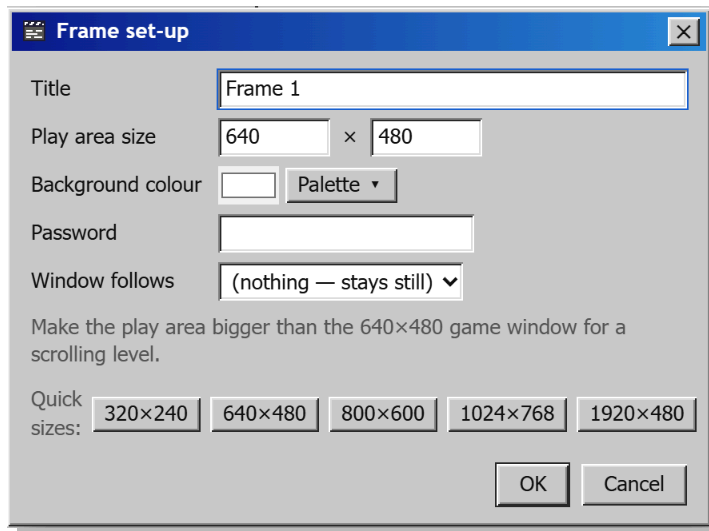


FIG 3.2 The Frame set-up box

Setting	What it does
Title	The frame's name, shown on its card and in the status bar.
Play area size	Width and height in pixels (64 to 8192). Quick-size buttons offer 320×240, 640×480, 800×600, 1024×768 and three windows wide.
Background colour	The colour behind everything. White to start.
Password	Optional. Players type it under Game ► Password... in the play window to start a new game on this level. Capitals don't matter.
Window follows	For play areas bigger than the game window: the object the window keeps in the middle. See <i>Beyond 1994</i> .

■ Game set-up

Open it from **Game ► Game set-up...** or the Storyboard's **Game set-up** button. These settings belong to the whole game.


Game set-up
✕

Title

Toy Rebound

Author

Window size

640 × 480

☐ resize frames too

Border colour

Palette ▾

Players start with

0 points and 3 lives

Options

☒ Game has a menu bar (Game / Options / Help)
 ☒ Crisp pixel scaling
 ☒ Panic key (F5)

Touch joystick:

on touch screens ▾

Players' controls

	Up	Down	Left	Right	Fire 1	Fire 2
Player 1	Up arrow	Down arrow	Left arrow	Right arrow	Shift	Ctrl
Player 2	W	S	A	D	Q	E
Player 3	I	K	J	L	U	O
Player 4	Numpad 8	Numpad 5	Numpad 4	Numpad 6	Numpad 7	Numpad 9

Global values (numbers the whole game remembers)

+ Add a global value

Use them for things like Level, Gems or High score. They carry on from frame to frame.

How to play (shown from the game's Help menu)

OK

Cancel

FIG 3.3 The Game set-up box

Setting	Starts as	What it does
Title	My game	Shown on the game window and in its About box.
Author	(blank)	Your name, shown in About.
Window size	640 × 480	The size of the game window. Tick <i>resize frames too</i> to change frames that were the old size.
Border colour	Black	Fills the space round the play area when the window is bigger than the game.
Players start with	0 points and 3 lives	Every player's score and lives at the start.
Game has a menu bar	On	The Game / Options / Help menus above the game.
Crisp pixel scaling	On	Enlarges by whole numbers so pixel art stays sharp.
Panic key (F5)	On	F5 hides the game behind a blank screen, and brings it back.
Touch joystick	On touch screens	An on-screen joystick and fire buttons. Choose <i>always</i> or <i>never</i> to override.
Players' controls	See below	The keys each player uses.
Global values	None	Numbers the whole game remembers. See <i>Beyond 1994</i> .
How to play	(blank)	Instructions shown from the game's Help menu.

To change a key, click its box in the **Players' controls** table and press the new key. Esc cancels. These are the keys each player starts with:

Player	Up	Down	Left	Right	Fire 1	Fire 2
1	↑	↓	←	→	Shift	Ctrl
2	W	S	A	D	Q	E
3	I	K	J	L	U	O
4	Numpad 8	Numpad 5	Numpad 4	Numpad 6	Numpad 7	Numpad 9



The Level Editor

The Level Editor is where you build a frame. Pick a library on the left, pick an object from the shelf along the top, and click the play area to drop it.

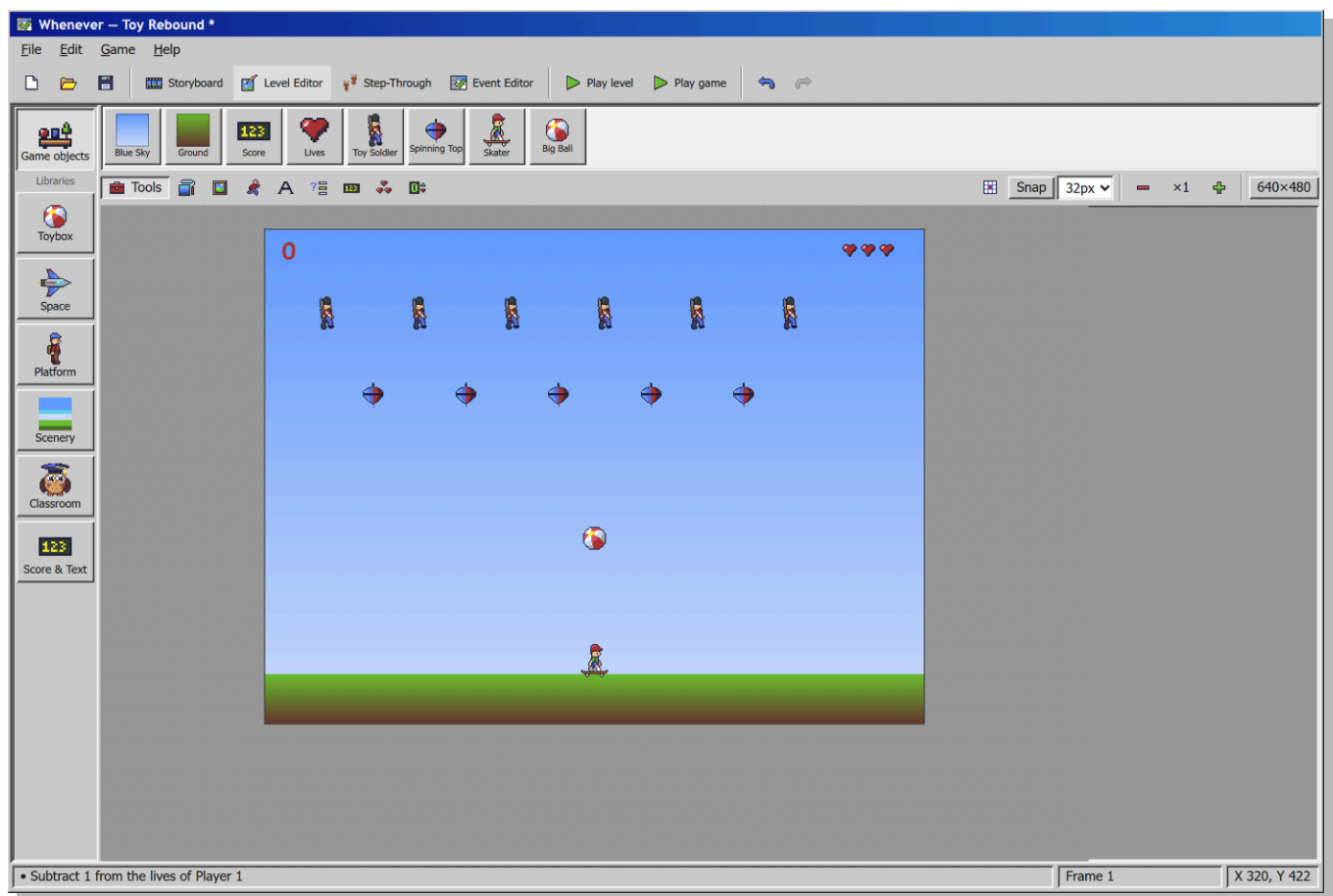


FIG 4.1 The Level Editor showing Game objects: every object already in the game

Libraries and the shelf

Game objects at the top of the left column lists every object already in your game. Right-click one of its tiles for **Edit...** or **Delete from game**. Deleting an object that is in use asks first, and also removes its conditions and actions from your events.

The six libraries hold ready-made objects with original pixel art. Most already have a movement set up. Rest the pointer on one to see what it does.

Library	What's in it
Toybox	Big Ball (bounces), Bat (follows the mouse left and right), Skater (arrow keys), Toy Soldier (marches), Spinning Top, four Bricks, Heart, Balloon (bobs), Star, Present
Space	Spaceship, Fighter (left and right only), Alien (marches in waves), UFO, Laser, Side Laser, Enemy Bolt, Explosion (plays once), Asteroid and Small Rock (drift and bounce), Planet
Platform	Hero (walks, jumps, climbs), Grass Block, Ledge, Dirt Block, Castle Brick, Ladder, Coin, Spikes, Blob (patrols), Door, Flag, Cloud, Bush, Tree
Scenery	Blue Sky, Night Sky, Sunset, Sea, Ground, Wall, Shelf, Brick Wall. All can be resized.
Classroom	Owl, Pointer (follows the mouse), Tick, Cross, Trophy, Apple, Pencil, Question
Score & Text	Score, Lives, Counter, Health Bar, Title, Message

■ Placing objects

1 Click an object on the shelf. A see-through copy follows the pointer.

2 **Left-click** the play area to drop one copy and stop. **Right-click** to drop a copy and keep going. **Esc** stops.

The first time you place a library object, it joins your game's objects. Every copy you place after that is the same object: copies share pictures, movement and events. To make one copy separate, right-click it and choose **New object (independent copy)**.

The first time a moving object goes into a frame, Whenever also adds a few events for it. They are ordinary events, so you can change or delete them.

Movement	Events added
Bouncing ball	Leaves the play area → Bounce. Collides with the backdrop → Bounce.
Eight directions, Race car	Leaves the play area → Stop. Collides with the backdrop → Stop.
Mouse	Collides with the backdrop → Stop.

■ Selecting, moving and resizing

■ **Click** an object to select it. **Ctrl-click** or **Shift-click** adds or removes one. **Drag** across empty space to select everything the box touches.

■ **Drag** a selected object to move it. The arrow keys nudge it 1 pixel; **Shift** + arrow nudges 10 pixels, or one grid square when Snap is on.

- A single Quick Backdrop, Text or bar Counter shows eight blue handles. Drag them to resize it.
- **Double-click** an Active object to open the Animation editor, a Backdrop to open the Picture editor, and anything else for its properties.

■ Right-click an object

Command	Shown for	What it does
Edit movement...	Active objects	How it moves. See <i>Objects and movements</i> .
Edit animations...	Active objects	The pictures it shows.
Alterable values...	Active objects	Numbers each copy remembers, such as Health.
Collisions ▶ Fine / Box	Active objects	Pixel-perfect collisions, or a quicker rectangle.
Edit picture..., Edit text... and so on	Other objects	Opens its properties.
Obstacle ▶ No / Yes / Platform / Ladder	Backdrops	Scenery, a solid wall, a floor you can jump up through, or something to climb.
Alignment, Make invisible at start	Text	Left, centred or right; hidden until an event shows it.
Families...	Active objects	Puts it in a family, such as Baddies.
Scrolling ▶	Everything	How it moves when the window scrolls. See <i>Beyond 1994</i> .
View ▶ To front / Forward one / Backward one / To back	Everything	What is drawn in front of what.
Rename...	Everything	Names must be different so events can tell objects apart.
New object (independent copy)	Objects with two or more copies	Makes this copy a separate object.
Duplicate, Delete	Everything	<code>Ctrl+D</code> and Del do the same.

■ The toolbar

- **Tools** shows eight buttons that make brand-new objects: Quick Backdrop, Backdrop, Active object, Text, Question, Score, Lives and Counter. Afterwards you are placing the new object.
- The **grid** button shows a dotted grid. **Snap** lines objects up to it. The list beside it sets the grid size, from 8 to 64 pixels (32 to start).
- – and + zoom from $\times\frac{1}{4}$ to $\times 8$. Ctrl + mouse wheel does the same.
- The size button (for example 640×480) opens Frame set-up.

■ The holding area

A grey, dotted border 160 pixels wide surrounds the play area. Objects parked there exist when the frame starts, but sit out of sight until an event moves them in. The status bar shows the pointer's X and Y; they go below zero in the holding area.

■ Keys

Key	What it does
<code>Esc</code>	Stop placing; select nothing.
<code>Delete</code> or <code>Backspace</code>	Delete the selected objects.
<code>Ctrl+C</code> , <code>Ctrl+X</code> , <code>Ctrl+V</code>	Copy, cut, paste. Each paste lands 16 pixels further on.
<code>Ctrl+D</code>	Duplicate.
<code>Ctrl+A</code>	Select everything.
<code>Arrow keys</code> , <code>Shift + arrow keys</code>	Nudge.
<code>Ctrl+Z</code> , <code>Ctrl+Y</code>	Undo and redo, up to 300 steps.



Objects and movements

Every object is one of eight kinds. Only Active objects move, animate and collide. The other kinds are scenery, words or numbers.

■ The eight kinds

Kind	What it's for	How you change it
Active object	Anything that moves, animates or reacts: players, balls, enemies.	Animation editor, movement, alterable values, families.
Backdrop	A picture that stays put: scenery, walls, floors.	Picture editor; Obstacle.
Quick Backdrop	A resizable block of colour, gradient or tiled pattern.	Properties: fill, border, obstacle.
Text	Titles and messages, with several paragraphs to switch between.	Properties.
Counter	A number to count with, shown as digits or a bar, or kept hidden.	Properties.
Score	Shows a player's score.	Properties.
Lives	Shows a player's lives, as a row of pictures or a number.	Properties.
Question	A multiple-choice question for quizzes.	Properties.

■ Properties

Double-click any object except an Active object or Backdrop to open its properties. Changes apply to every copy.

- **Text:** a list of paragraphs (the first shows at the start; actions switch to others), font, colour, alignment, and **Invisible at start**.
- **Quick Backdrop:** a fill (**Solid**, **Gradient** or **Mosaic**, a tiled picture), an optional border, the **Obstacle** type, and the size for new copies.

- **Counter:** **Initial value**, **Minimum** and **Maximum**, and **Display as** *Hidden, Numbers, Horizontal bar* or *Vertical bar*.
- **Score** and **Lives:** which player, and how to show it.
- **Question:** the question and up to five answers. Tick the correct one. In your events, use **Ask the question**, then test **Is the answer correct?** or **Is the answer wrong?**

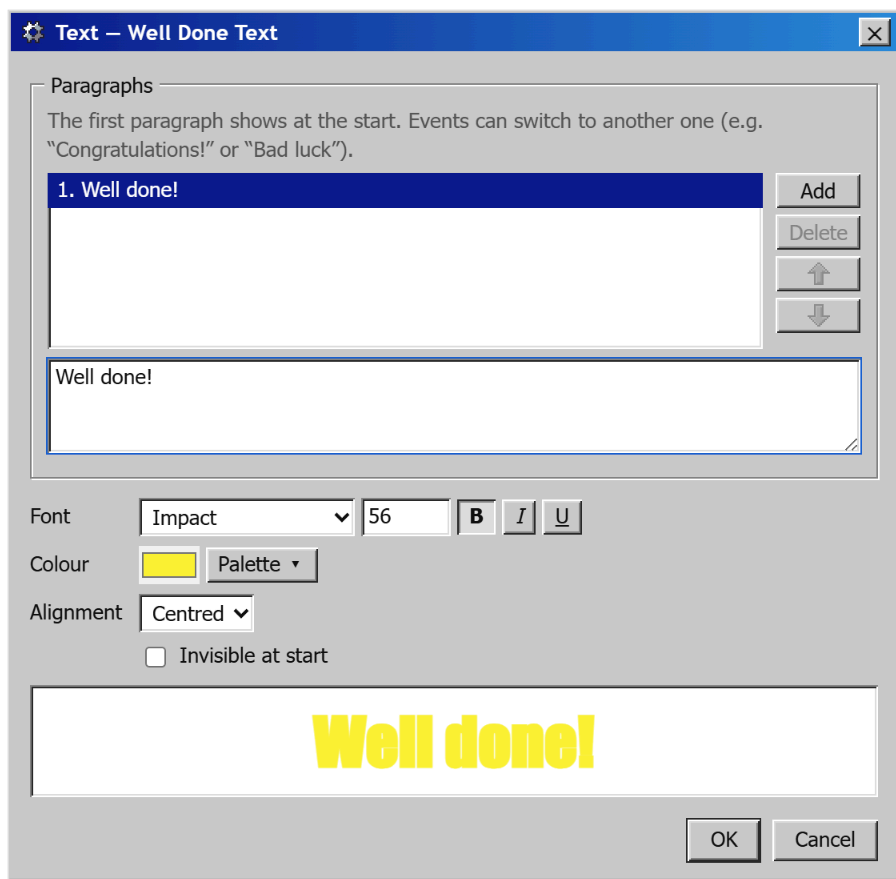


FIG 5.1 The properties of a Text object: paragraphs, font, colour and alignment

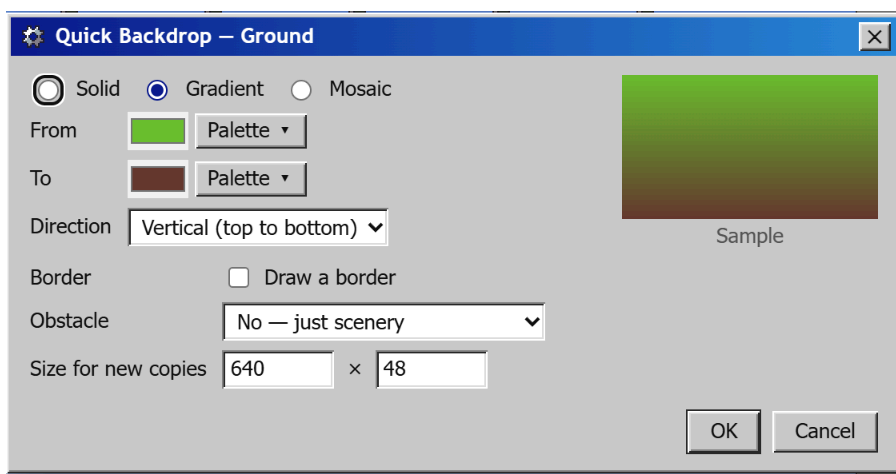


FIG 5.2 The properties of a Quick Backdrop: fill, border and obstacle

■ Alterable values

Each copy of an Active object remembers its own numbers, lettered A to Z. Give them names that say what they're for, like *Health* or *Ammo*: right-click the object and choose **Alterable values....**

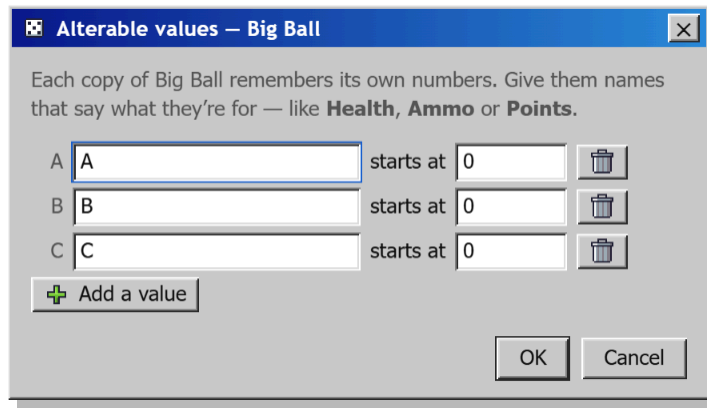


FIG 5.3 The Alterable values box for the Big Ball

Events use them through the object's **Values** menu (set, add, subtract, compare), and expressions read them with `Value("Alien", "Health")`.

■ Movements

Right-click an Active object and choose **Edit movement....** The list on the left has the player-controlled movements, then the computer-controlled ones. **Try movement** plays the frame with only this object and the backdrops, and no events; `Esc` stops it.

Movement	What moves it	Main settings
Static	Nothing. It can still animate and react.	None.
Mouse	The mouse pointer.	Where it may go: anywhere, left and right only (a bat), up and down only, or a box round its start.
Eight directions	Arrow keys or joystick.	Player, Speed, Acceleration, Deceleration, Possible directions, Initial direction.
Race car	Up accelerates; left and right steer.	Speed, Acceleration, Deceleration (the brakes), Rotating speed, Number of angles, reverse gear.
Platform	Walk, jump and climb, with gravity.	Speed, Acceleration, Deceleration, Gravity, Jump strength, Jump controls.
Bouncing ball	Itself.	Speed, Ball deceleration, Bounce randomiser, Bounce security, Number of angles, Initial direction.
Path	Itself, along a path you draw.	The path, Loop the movement, Reverse at end, Reposition object at end; a speed and a pause at each point.

The sliders run from 0 to 100. At speed 50 an object crosses a 640-pixel window in about two seconds. Untick **Moving at start** to keep an object still until a **Start** action.

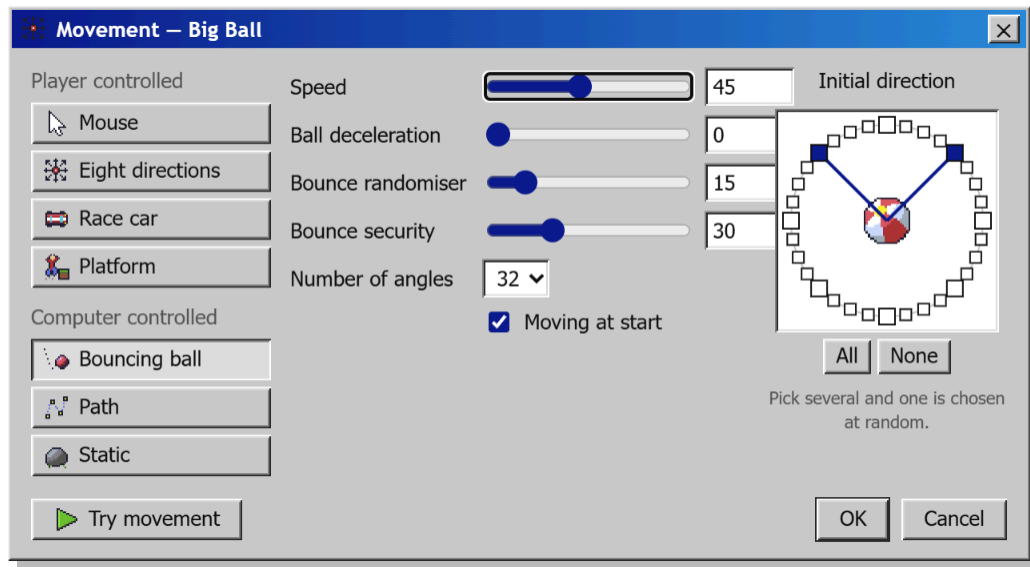


FIG 5.4 The Bouncing ball settings for the Big Ball

■ Directions

Directions use a wheel of 32 squares. Direction 0 points right and the numbers go round anticlockwise: 8 is up, 16 is left and 24 is down. The bigger squares are the eight joystick directions. Tick several initial directions and one is picked at random each time.

■ Platform games

The Platform movement needs floors. Set them with the Obstacle type of your backdrops: **Yes** for solid walls and ground, **Platform** for ledges you can jump up through and stand on, and **Ladder** for things to climb. The jump button starts as Button 1 (**Shift**); **Jump controls** can change it to the up arrow or Button 2.

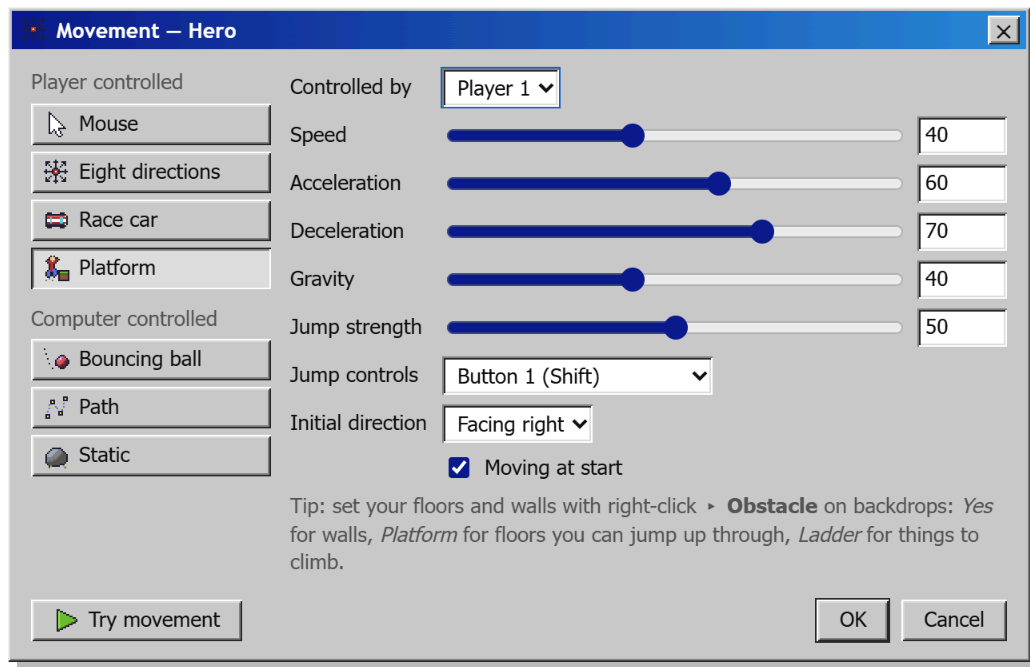


FIG 5.5 The Platform movement settings for the Hero

■ Drawing a path

Choose **Path** and the dialog shows a small copy of the frame. The blue dot is the starting point.

- **New line:** click to add the next point.
- **Tape mouse:** hold the button down and draw freehand.
- **Edit points:** drag points about. **Delete point** and **Clear path** do what they say.
- Click a point to set the **Speed from here** and a **Pause** in seconds.



Events

Events are the rules of your game. Each event is one row: whenever all of its conditions are true, its actions happen. The Event Editor shows every event in the current frame as a grid.

■ Reading the grid

Each row is an event, numbered on the left. Its conditions come next. Then there is one column for each thing that can do something. A ✓ in a box means the event does something to that column; a small number shows how many actions there are.

Column	Its conditions	Its actions
Special	Compare two values, Always and Never, limit conditions, global values, groups.	Global values, groups, open a web page.
Sound	None.	Play and stop samples and music.
Storyboard controls	Start of frame, end of frame.	Next, previous or any frame; restart; end the game; scroll the window.
Create new objects	Count objects, or test for none in a zone.	Create an object.
Player 1 to 4	Joystick, score, lives.	Score, lives, ignore or restore the controls.
Each object	Collisions, position, movement, animation, direction, visibility, values, counting.	Movement, position, direction, animation, visibility, destroy, shoot, values, and more.
Families	The same as their objects, for any member.	The same as their objects, for any member.

Two more icons have conditions but no column: **Timer** (every so many seconds, or at a set time) and **Mouse & keyboard** (key presses, clicks, the pointer).

■ Moments and states

Conditions come in two colours, and the difference matters.

- **Green conditions happen at a moment:** *collides with, upon pressing a key, every 2 seconds.* They are true for one instant, so their actions happen once.

- **Black conditions are true while something lasts:** *is overlapping, repeat while a key is pressed, score of Player 1 > 100*. Their actions happen again every tick, sixty times a second, for as long as the condition stays true.
- A red **X** means the condition is negated: the test is turned round, so *is overlapping* becomes *is not overlapping*.

If something happens far too often, a black condition is usually why. Use a green condition instead, or add **Special**

- **Limit conditions** ► **Only one action when event loops**. That makes the actions happen once each time the event becomes true.

■ Making an event

- 1 Click **New condition** on the toolbar, or in the empty row at the bottom.
- 2 Click an icon. The top rows are Special, Storyboard controls, Timer, Create new objects, Mouse & keyboard and the players; your objects and families follow.
- 3 Choose a condition from the menu. Green entries happen at a moment.
- 4 If the condition needs settings, such as which key, a box asks for them.
- 5 In the new row, right-click the box under each object that should do something, and choose an action.

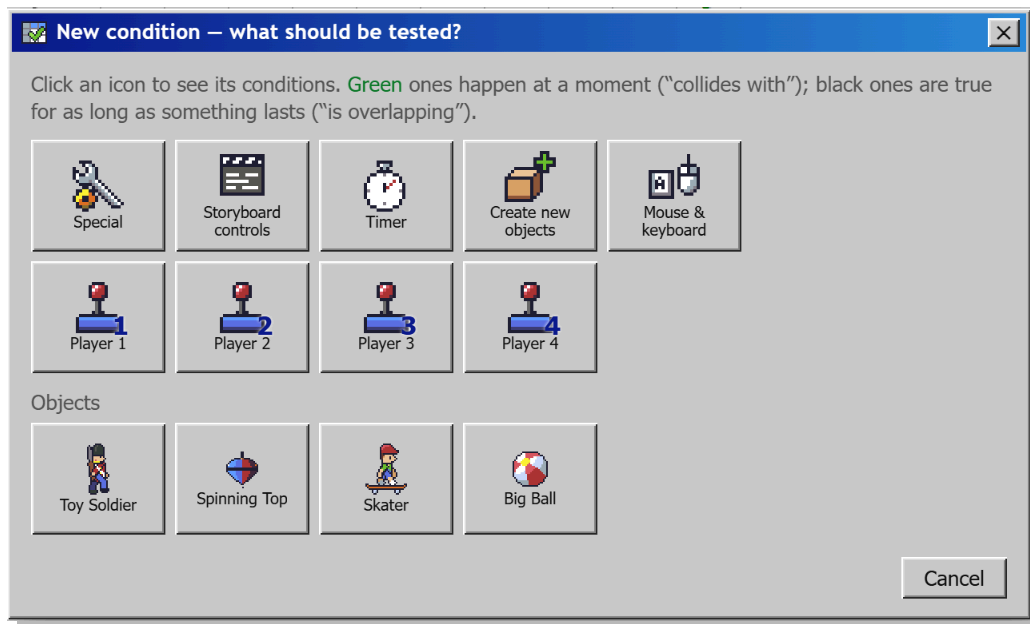


FIG 6.1 The New condition box: system icons on top, then the objects

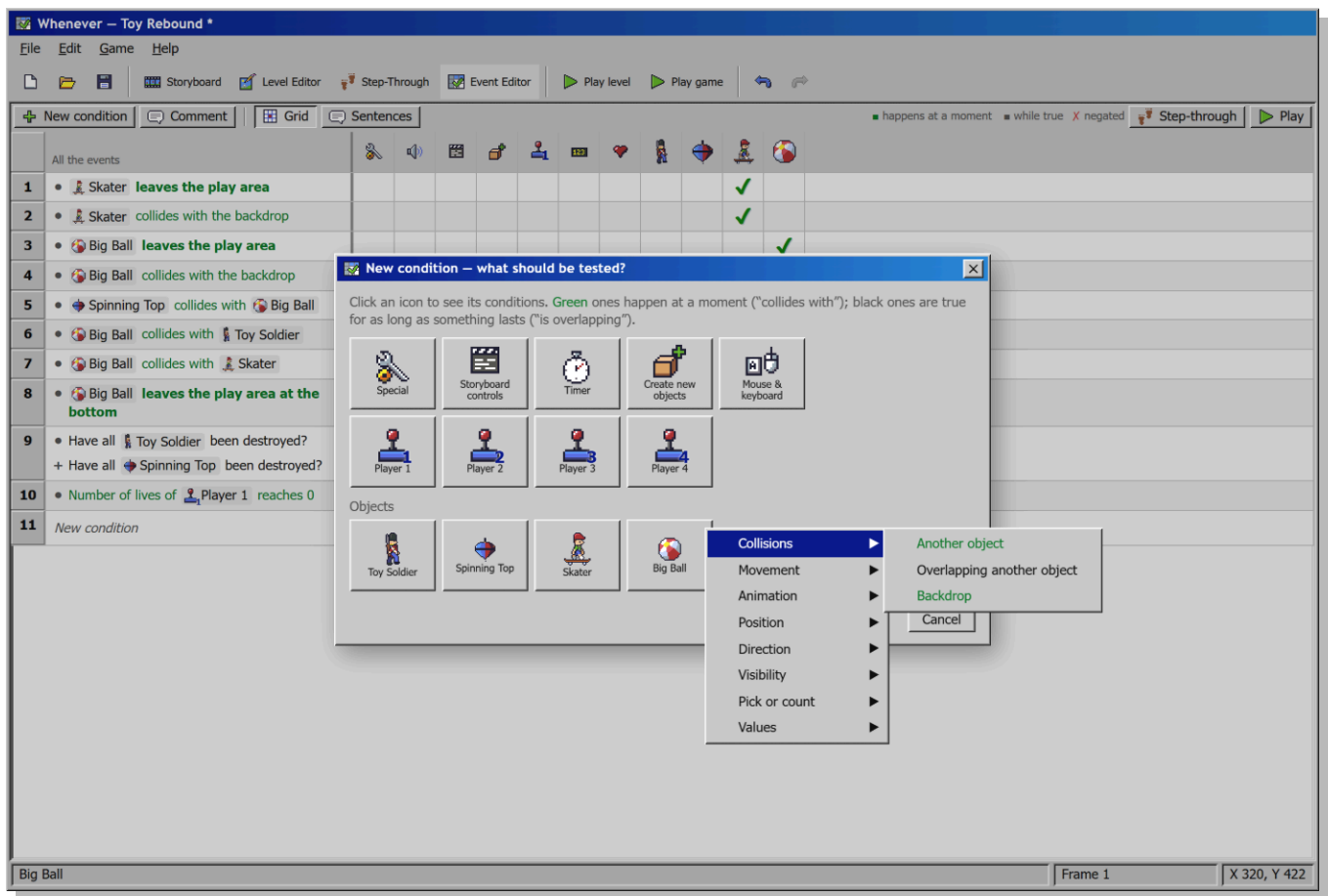


FIG 6.2 The Big Ball's condition menu, open at Collisions

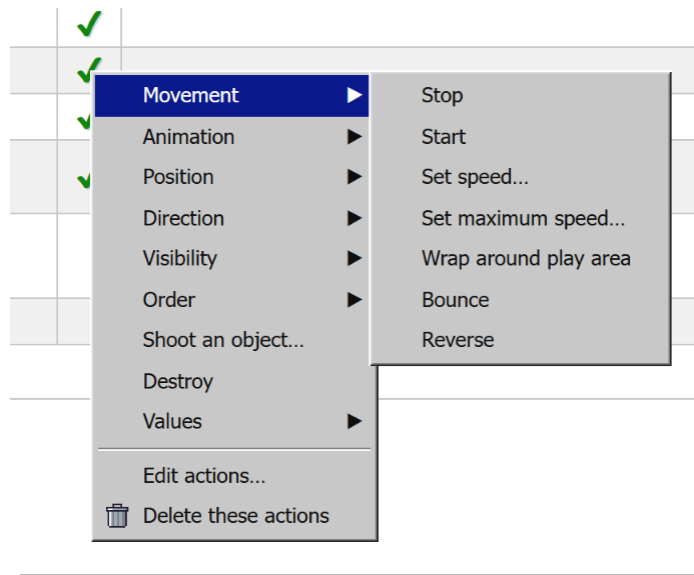


FIG 6.3 Right-clicking a box under the Big Ball opens its action menu

All the conditions in one event must be true together: they mean *and*. For *or*, make two events with the same actions.

Actions run in the order you added them, not the order of the columns. Click a ticked box to see the **Actions for ...** list, where you can change the order, edit or delete them.

■ Settings boxes

- **Numbers** can be calculations: `50`, `Random(10) + 1` or `X("Big Ball")`. **Calc...** opens the expression calculator. A red border means Whenever can't read it, and a message says why.
- **Positions** are either X and Y on the play area (or click the little picture), or relative to an object's hot spot or action point.
- **Zones** are a box you drag on the little picture.
- **Comparisons** are buttons: `=` `≠` `<` `≤` `>` `≥`.

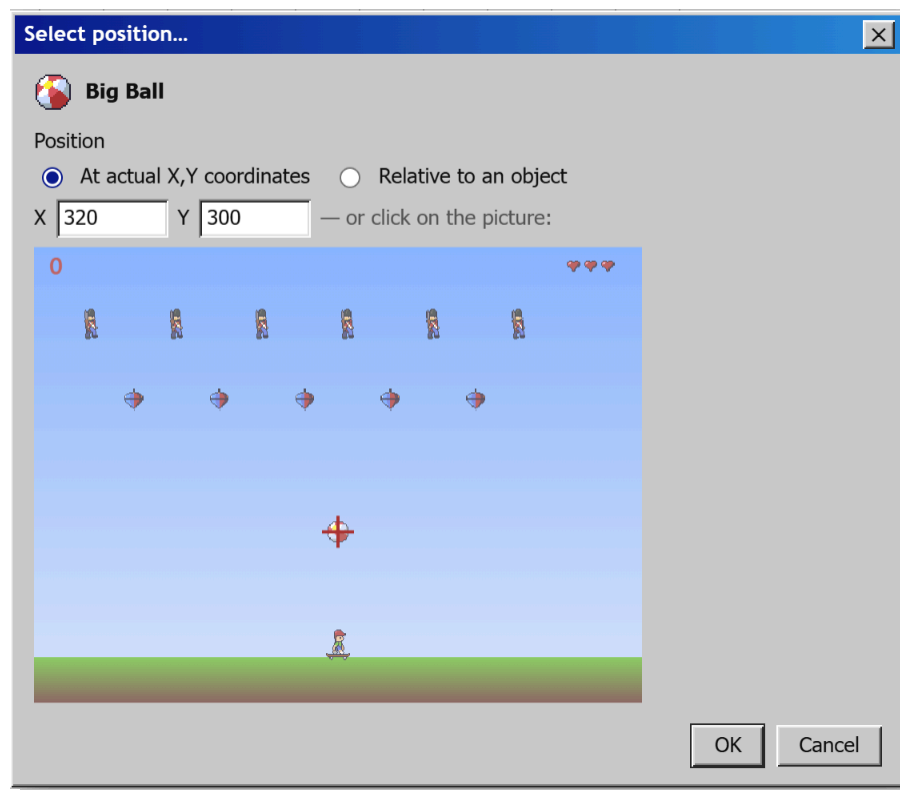


FIG 6.4 Setting a position: X and Y, or a click on the picture of the frame

■ Which copy does it act on?

When a condition tests an object, the actions act only on the copies that passed the test. *Big Ball collides with Toy Soldier* → *Destroy Toy Soldier* destroys the one soldier that was hit, not all six. If no condition mentions an object, its actions apply to every copy. **Pick or count** ► **Pick one at random** chooses a single copy.

■ Changing events

Right-click on	You can
An event's number	Add a new condition, insert an event above, put it in a group, add or edit a comment, duplicate, disable, move up or down, delete.
A condition	Edit, replace, negate, insert a condition before it, move it, delete it.
A box	Add an action; edit or delete the actions already there.
A column heading	Show only the events about that object; edit the object.
A group heading	Add an event to the group, rename it, choose whether it starts switched on, fold it up, ungroup.

- **Double-click** a condition to change its settings.
- **Drag** an event's number to move the event. **Drag** a tick to another row in the same column to copy those actions.
- **Comment** adds a note row. Comments are for people; the game ignores them. A comment row gets a number like an event, so the numbers never skip.
- **Disable event** skips an event without deleting it. It is handy while testing.
- **Click a column heading** to show only the events about that object. Shift-click or Ctrl-click adds more. Click the **Filtered** button to show everything again.

■ Sentence view

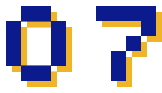
Click **Sentences** on the toolbar to read every event as plain English: **Whenever** ... *and* ..., followed by its actions. Moments are green and negated conditions say NOT in red. Click **Grid** to go back.

■ How a game runs

A running game repeats these steps sixty times a second. Each repeat is called a tick.

- 1 Every object moves, following its movement.
- 2 Animations move on a frame.
- 3 Whenever checks for collisions and for objects at the edges of the play area.
- 4 The events are read from top to bottom. For each event whose conditions are all true, its actions run.
- 5 If the window follows an object, it moves.
- 6 Destroyed objects disappear, and if an action asked for another frame, it starts.

Order matters: an event lower down sees the changes made by the events above it, such as a new score.



The Step-Through Editor

The Step-Through Editor writes events while you play. It runs the current frame at double size and stops the first time something happens that your events don't handle yet. Then you choose what should happen, and play on.

What makes it stop

- Two objects collide.
- An object hits a solid backdrop.
- An Active object leaves the play area on one side.

Each new situation stops the game once per run. Keys, timers and counting can't stop it, so make those events in the Event Editor.

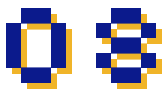
Choosing what happens

- 1 The **Game control panel** on the right shows what happened, such as *Spinning Top collides with Big Ball*.
- 2 Below it are icons for everything that can act. The objects involved come first, outlined.
- 3 Click an icon and choose an action from its menu. The first action you choose makes a new event; the others join it.
- 4 The **Selected actions** list shows them. The bin button removes one.
- 5 Click **Continue game**. If you chose nothing, the game ignores the situation for the rest of this run.

■ The other buttons

Button or key	What it does
	Stops the game.
Restart game	Plays the frame again from the start.
Event Editor	Shows all the events in the grid.

Events you add or change in the Step-Through Editor work in the running game straight away. Below the panel is the debugger: see *Finding bugs*.



Pictures and animations

An Active object shows animations: named runs of pictures, with a set for each direction it can face. Double-click an Active object to open the Animation editor. Double-click one of its frames to draw it in the Picture editor.

The Animation editor

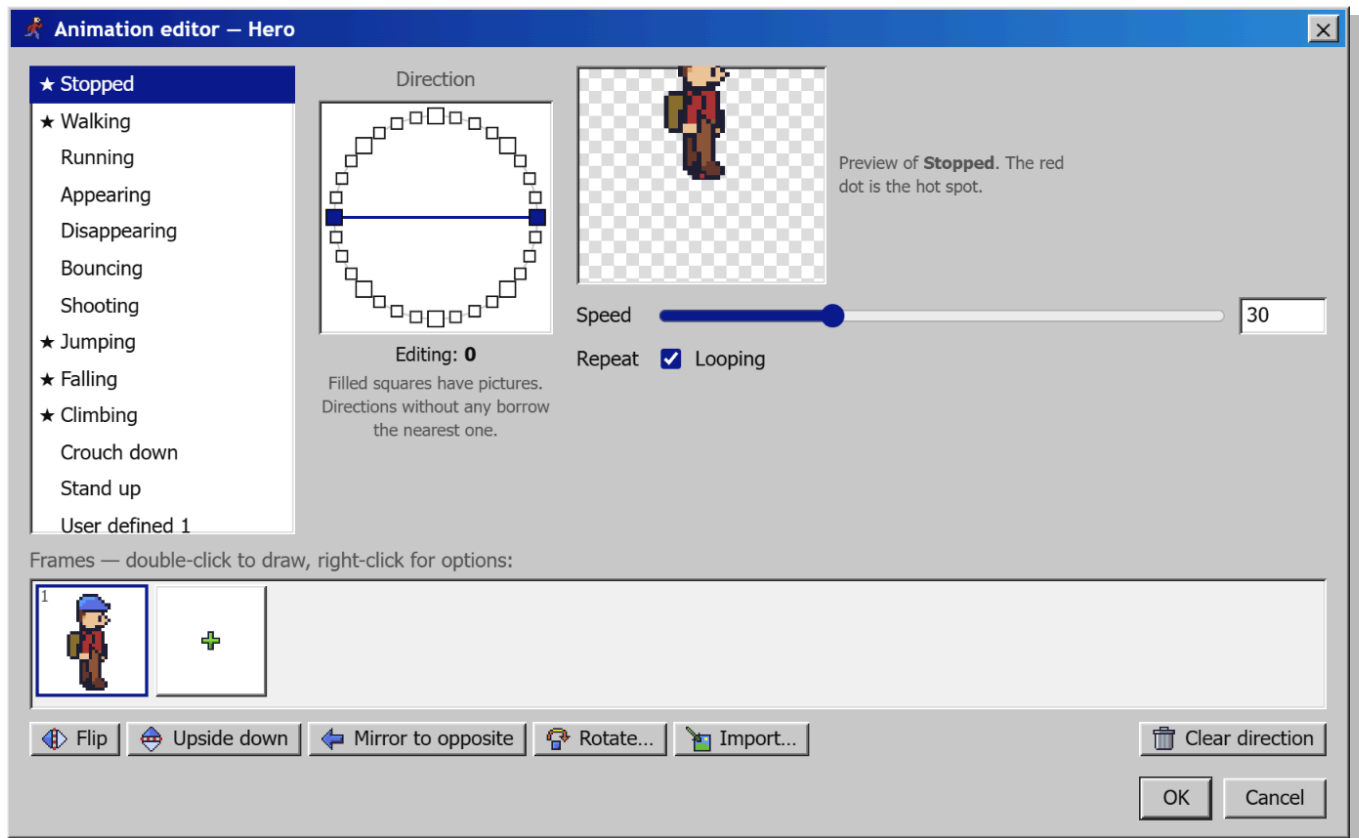


FIG 8.1 The Animation editor for the Hero, showing the Walking animation

- **The list** on the left names the animations. A ★ marks the ones that have pictures. The last four, User defined 1 to 4, can be renamed with a double-click.

- **The direction wheel** picks which of the 32 directions you are editing. Filled squares have pictures. Directions without pictures borrow the nearest one, so a single picture facing right is enough for many objects.
- **The preview** plays the animation. The red dot is the hot spot.
- **Speed** sets how fast it plays. Moving animations have a **Min speed** and a **Max speed** instead: slow when the object creeps, fast at top speed.
- **Repeat**: tick **Looping**, or untick it to play a set number of times. **Loop back to frame** plays an intro once and then loops from that frame.

Whenever picks some animations by itself. Moving objects show **Walking** and stopped ones show **Stopped**. Platform movers also use **Jumping**, **Falling** and **Climbing**. **Appearing** plays when an object is created and **Disappearing** when it is destroyed. The others play when an action asks: **Animation** ▶ **Change** ▶ **Animation sequence...**

■ Frames and directions

The strip along the bottom holds the frames. Double-click one to draw it. **+** adds a copy of the selected frame. Right-click a frame for **Edit frame...**, **Insert a copy**, **Resize zoom...** (a growing or shrinking sequence), **Move left**, **Move right** and **Delete**.

Button	What it does
Flip	Mirrors every frame in this direction.
Upside down	Turns every frame upside down.
Mirror to opposite	Makes the opposite direction as a left-right mirror image: walking left from walking right. Straight up and down have no left-right opposite, so it is greyed out there.
Rotate...	Turns this direction's pictures to fill 4, 8, 16 or all 32 directions. Good for cars and spaceships.
Import...	Adds a picture file, or cuts a sprite sheet into frames (you say how many columns and rows).
Clear direction	Removes every frame from this direction.

■ The Picture editor

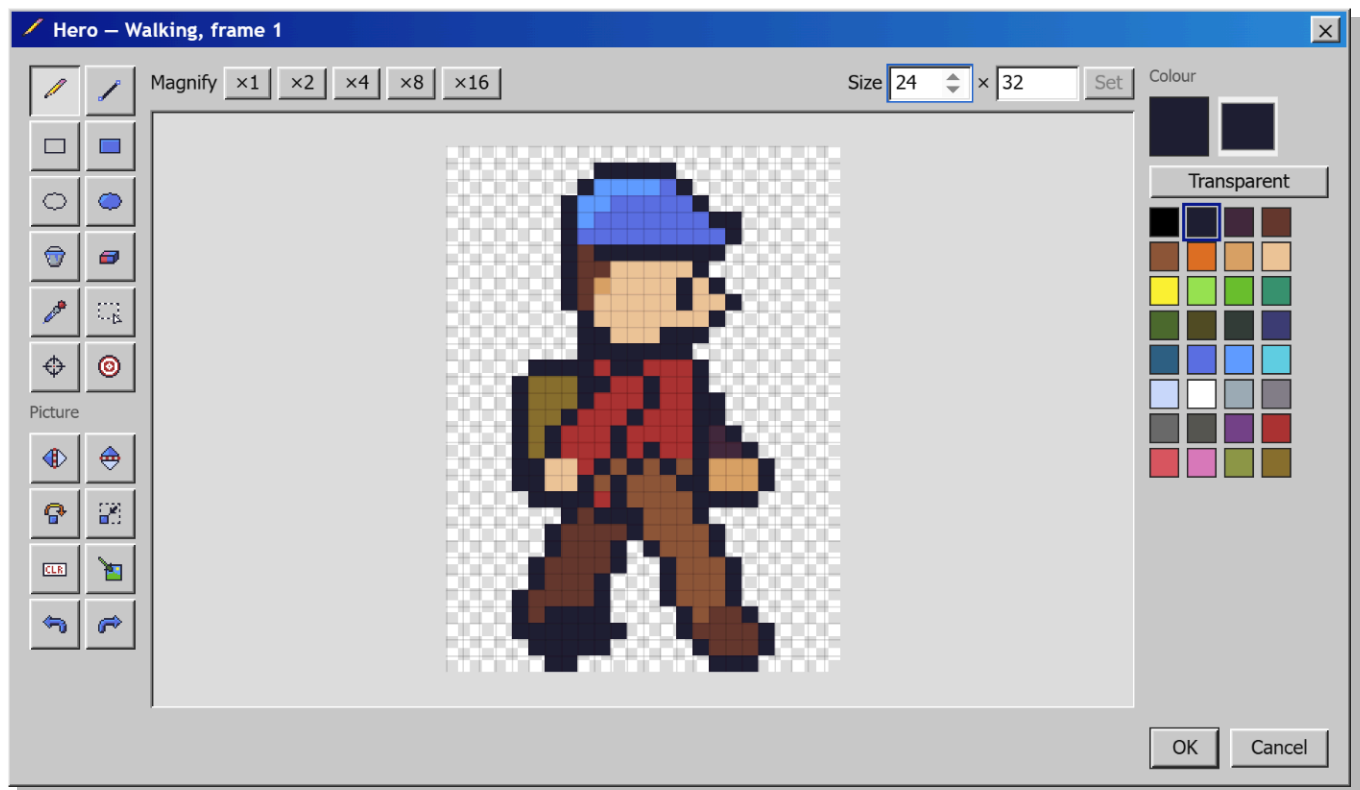


FIG 8.2 The Picture editor drawing one frame of the Hero

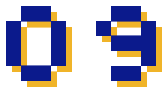
Tool	What it does
Draw	Plots single pixels. With most tools, a right-click picks up the colour under the pointer.
Line, Rectangle, Filled rectangle	Drag to draw.
Circle, Filled circle	Drag outwards from the centre.
Paint pot	Fills an enclosed area.
Eraser	Paints with the transparent colour.
Pick a colour	Takes a colour from the picture.
Select	Drag to mark an area, then drag inside it to move those pixels. Delete clears it.
Hot spot	Sets the object's handle (see below).
Action point	Sets where shots and new objects come from.

The buttons beside the tools flip, rotate, **Shrink** (trim the empty border), clear, and import a picture file. **Undo** is `Ctrl+Z` and **Redo** is `Ctrl+Shift+Z`. **Magnify** zooms from x1 to x16, with a pixel grid from x8. **Size** changes the picture's width and height. The colour panel has a 32-colour palette, a colour mixer and the **Transparent** colour.

■ Hot spot and action point

An object's position is the position of its hot spot. Put it at the feet of a character, so it stands on the ground, and in the middle of a ball. With the Hot spot tool chosen, the number keys jump it to the edges like a numeric keypad: 7 is the top-left corner, 5 the centre and 3 the bottom-right.

The action point is where a **Shoot** action fires from, and where objects created *relative to* this one appear. Put it at the end of a gun, or the front of a ship.



Sounds

Sounds play when an action in the **Sound** column says so. Make them with the built-in sound maker, or import your own files.

■ Choosing a sound

Every action that plays a sound has a sound box: the sound's name, a loudspeaker button to listen, and **Choose...**. Choose opens this box:

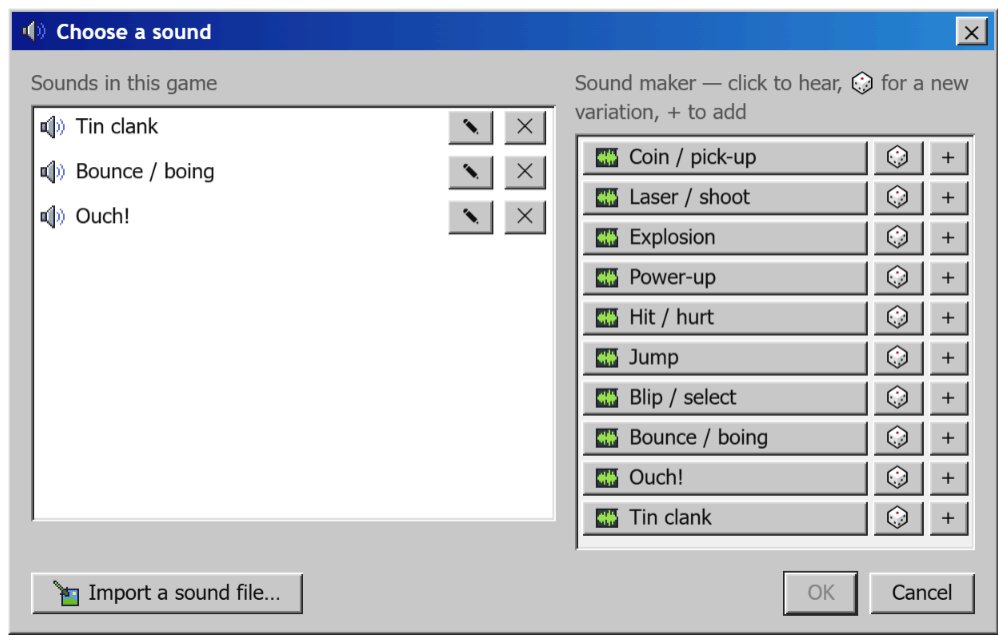


FIG 9.1 The Choose a sound box: the game's sounds on the left, each with rename and delete buttons, and the sound maker on the right

- **Sounds in this game**, on the left: click one to hear it, double-click to choose it, to rename it and to delete it from the game. If events use the sound, Whenever asks first, then removes those actions too.
- **The sound maker**, on the right, makes retro sound effects from a few numbers. There are ten kinds: *Coin / pick-up*, *Laser / shoot*, *Explosion*, *Power-up*, *Hit / hurt*, *Jump*, *Blip / select*, *Bounce / boing*, *Ouch!* and *Tin clank*. Click one to hear it, for a different variation, and **+** to add it to your game.
- **Import a sound file...** adds a WAV, MP3 or OGG file of up to 4 MB. Short sounds keep games quick.

■ Sound actions

Action	What it does
Play sample	Plays a sound once. Several samples can play at the same time. Tick Uninterruptible to stop other samples starting until it finishes.
Play and loop sample	Plays a sound over and over.
Stop a sample, Stop any sample playing	Stops one sample, or all of them.
Play music, Play and loop music	Plays a longer piece. Only one piece of music plays at a time.
Stop any music playing	Stops the music.

Players can switch samples and music off from the **Options** menu of a running game. Stand-alone games include only the sounds your events use.



Expressions

Anywhere a setting asks for a number, you can type a calculation instead.

`Random(6) + 1` rolls a die; `X("Big Ball") - 16` is 16 pixels left of the ball. Whenever works out the answer each time the action runs.

The expression calculator

Click **Calc...** beside any number box to open the calculator.

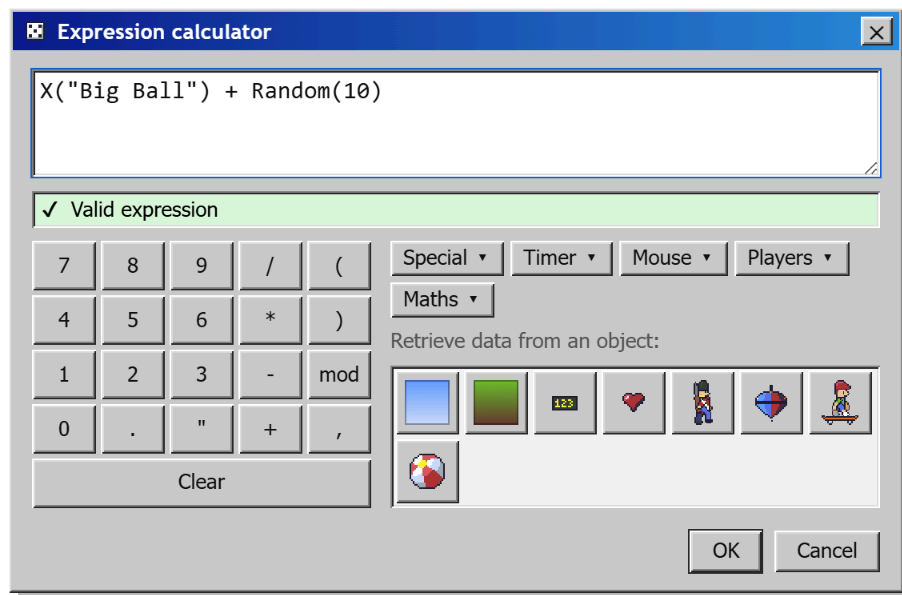


FIG 10.1 The expression calculator, with `X("Big Ball") + Random(10)` typed in

- Type in the box, or click the keypad buttons.
- The menus **Special**, **Timer**, **Mouse**, **Players** and **Maths** insert functions.
- Retrieve data from an object** lists your objects. Click one for its functions, such as its X position or its speed.
- The line under the box says **✓ Valid expression**, or explains what is wrong. **Where?** highlights the problem.

Writing expressions

- Arithmetic: `+`, `-`, `*`, `/`, `mod` (the remainder), `^` (power), and brackets.

- Object names go in quotes: `X("Big Ball")`. Upper and lower case don't matter in function names.
- Words in quotes are text. `+` joins text to text or numbers: `"Score: " + Score(1)`.
- When an event has picked one copy of an object, its functions read that copy. Otherwise they read the first one.
- Dividing by zero gives 0, not an error.

■ Functions about objects

`"Obj"` stands for an object's name, in quotes.

Function	Gives
<code>X("Obj")</code> , <code>Y("Obj")</code>	Its position.
<code>Width("Obj")</code> , <code>Height("Obj")</code>	Its size.
<code>Speed("Obj")</code>	Its speed, 0 to 100.
<code>Dir("Obj")</code>	The direction it faces, 0 to 31.
<code>Frame("Obj")</code>	Which animation frame is showing, from 1.
<code>Count("Obj")</code>	How many copies there are.
<code>Value("Obj", "Health")</code>	An alterable value, by name, letter or number.
<code>Counter("Obj")</code>	A counter's value.
<code>Paragraph("Obj")</code>	Which paragraph a text object shows, from 1.
<code>Distance("A", "B")</code>	Pixels between two objects.
<code>DirTo("A", "B")</code>	The direction from A to B, 0 to 31.

■ Other functions

Function	Gives
<code>Score(1)</code> , <code>Lives(1)</code>	A player's score or lives.
<code>Global("Best")</code>	A global value.
<code>Random(n)</code>	A whole number from 0 to $n - 1$.
<code>RandomRange(a, b)</code>	A whole number from a to b.
<code>Min(a, b)</code> , <code>Max(a, b)</code>	The smaller or larger.
<code>Abs(n)</code> , <code>Round(n)</code> , <code>Floor(n)</code> , <code>Ceil(n)</code>	Without the sign; rounded; rounded down; rounded up.
<code>Sqrt(n)</code>	The square root.
<code>Sin(deg)</code> , <code>Cos(deg)</code> , <code>Tan(deg)</code> , <code>Atan2(y, x)</code>	Trigonometry, in degrees.
<code>Timer</code>	Seconds since the frame started.
<code>MouseX</code> , <code>MouseY</code>	The mouse pointer's position.
<code>ObjectCount</code>	How many Active objects there are.
<code>FrameWidth</code> , <code>FrameHeight</code>	The play area's size.
<code>WindowWidth</code> , <code>WindowHeight</code>	The game window's size.
<code>DisplayX</code> , <code>DisplayY</code>	The window's left and top edge in a scrolling level.
<code>Pi</code>	π .

11

Beyond 1994

The original Klik & Play had none of the features in this chapter; they come from its successors, or from games now being played on tablets. The **Coin Quest** example (**File** ► **Open an example** ► **Coin Quest**) uses all of them.

■ Scrolling levels

A level can be bigger than the game window. The window then scrolls to follow the player.

- 1 Open **Frame set-up** and make the **Play area size** bigger than the window. The quick-size button on the right makes it three windows wide.
- 2 Under **Window follows**, pick the player's object. The window keeps it in the middle, and stops at the edges of the play area.

The Level Editor draws the window as a dashed blue box where the game starts: centred on the followed object, but kept inside the play area. It is labelled *Window (follows Hero)*.

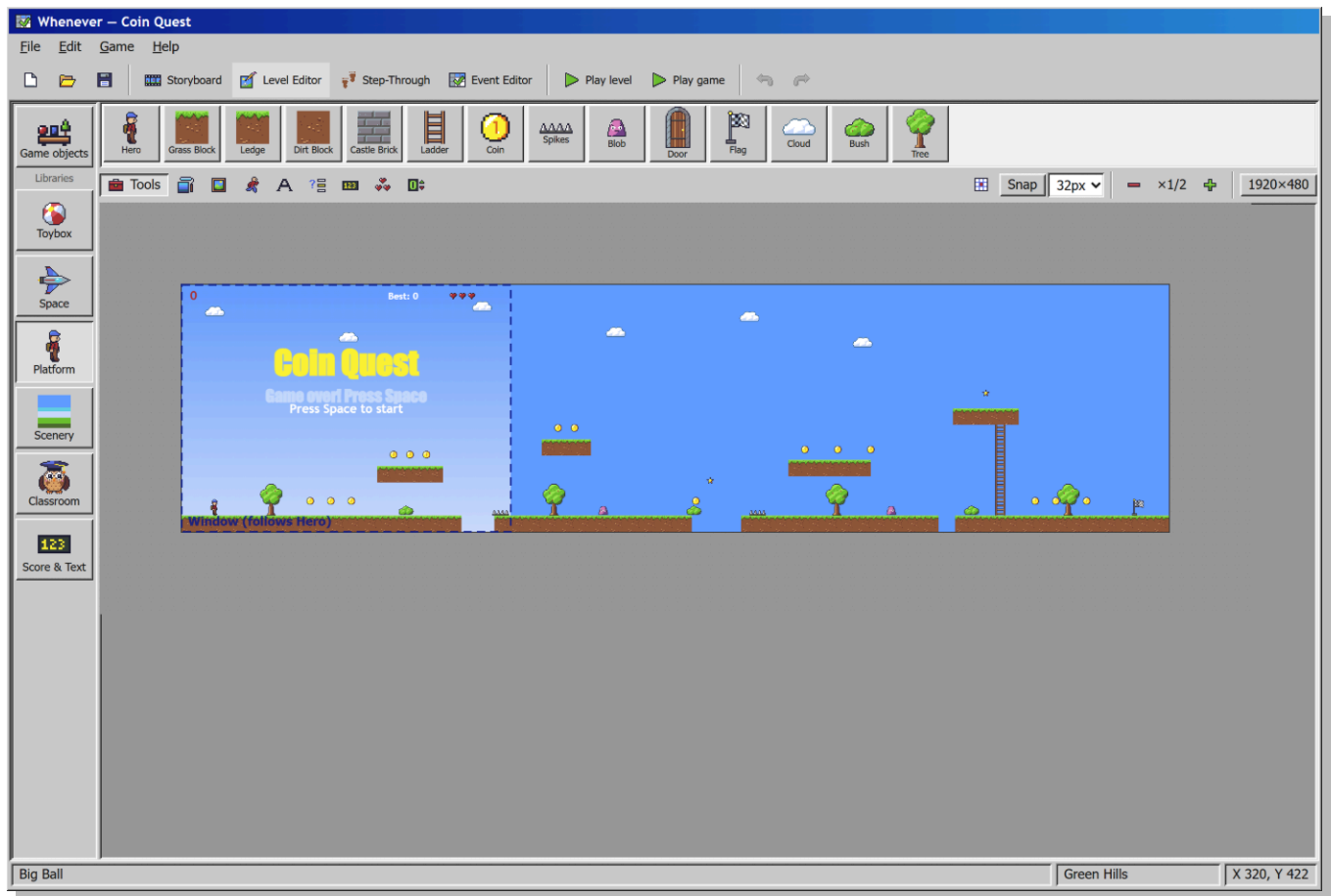


FIG 11.1 Coin Quest's level, three windows wide, with the dashed window box at the left

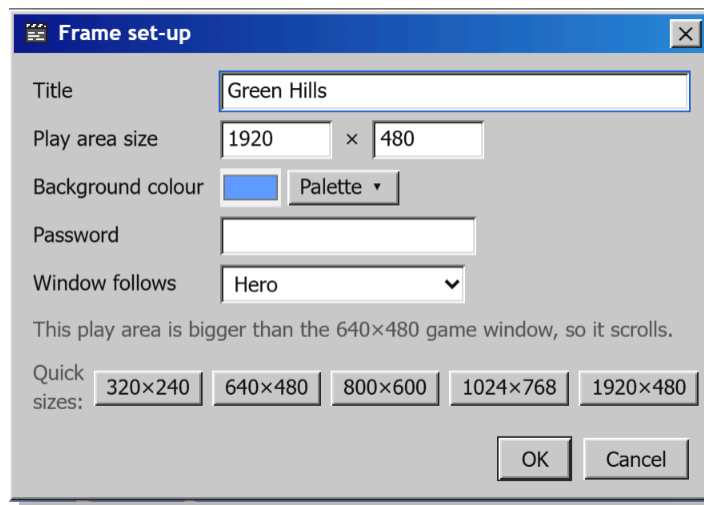


FIG 11.2 Frame set-up for a scrolling level: 1920 × 480, and the window follows the Hero

Right-click any object and use **Scrolling** to choose how it moves when the window scrolls:

Choice	Use it for
Moves with the level	Almost everything. This is normal.
Stays on the screen	Scores, lives, buttons and titles. Score, Lives, Counter and Question objects start this way.
Far away (half speed)	Distant scenery such as clouds. Moving more slowly than the level gives a feeling of depth.
Very far away (quarter speed)	Mountains, the moon.

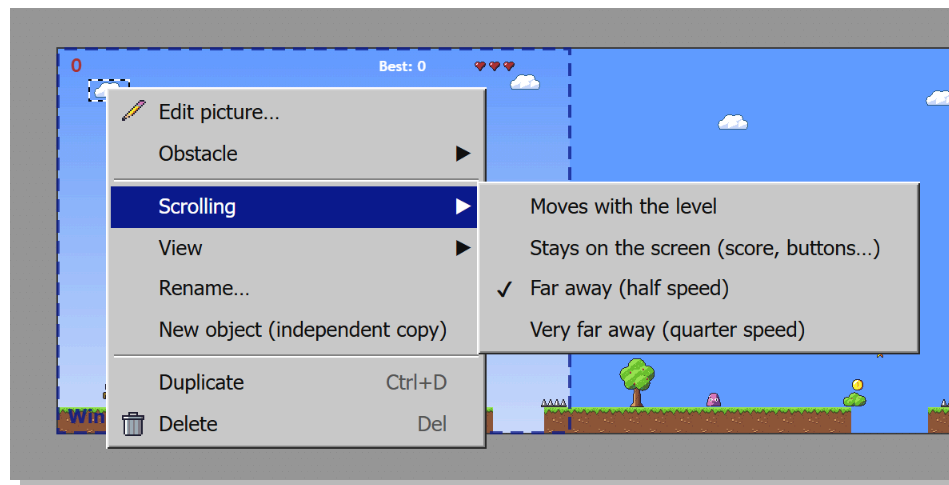


FIG 11.3 The Scrolling menu for a cloud, set to Far away

Two actions under **Storyboard controls** ► **Scrolling** move the window yourself: **Centre the window on an object** and **Scroll the window to a position**. The expressions `DisplayX` and `DisplayY` give the window's left and top edge.

■ Groups of events

A group is a set of events you can switch on and off while the game runs. Events in a group that is off are skipped. Groups are how a game has modes: a title screen, playing, paused, game over.

- **Make one:** right-click an event's number and choose **Group** ► **New group...**. Drag other events onto it, or use the same menu.
- **The heading row** shows the group's name, how many events it holds, and whether it is *on at start* or *off at start*. Click the triangle to fold it up.
- **Right-click the heading** for **New event in this group**, **Rename group...**, **Switched on at the start of the frame**, **Fold up** and **Ungroup (keep the events)**.
- **Actions: Special** ► **Groups of events** ► **Switch a group on** and **Switch a group off**.
- **Conditions: Is a group of events switched on?**, and the green **When this group is switched on**. It happens once, straight after the group switches on, which makes it the place to set a mode up. The green "After this

group has been on for..." happens once, that many seconds after the group switched on: right for traffic lights, countdowns and "Get ready" screens.

Switching a group off takes effect at once. A group switched on starts running on the next tick, so one key press can't run straight through several modes.

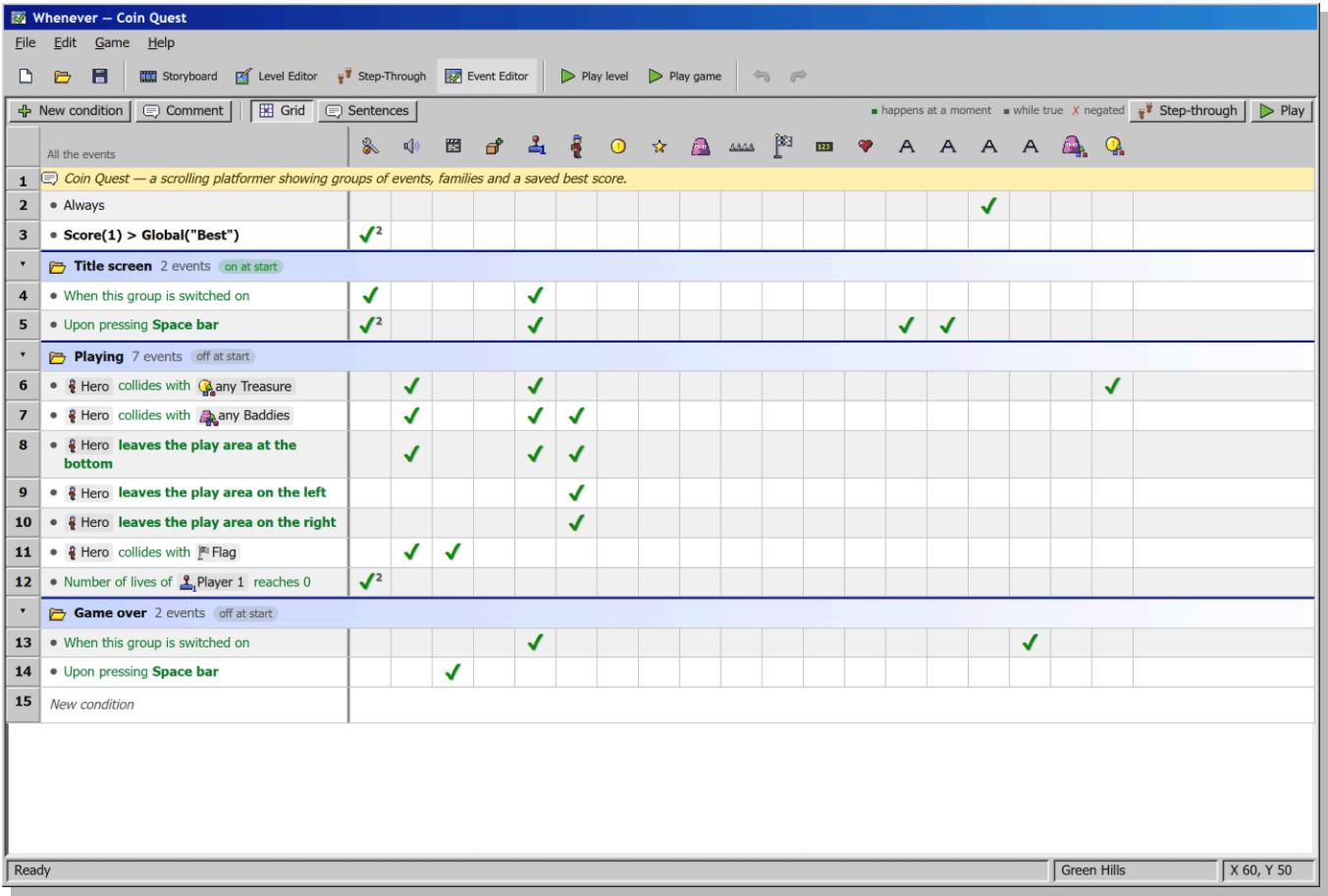


FIG 11.4 Coin Quest's events in three groups: Title screen, Playing and Game over

Coin Quest group	At start	What it does
Title screen	On	Ignores the controls. Space hides the title, switches this group off and switches Playing on.
Playing	Off	Coins, baddies, falling off, and the flag. When the lives run out it switches to Game over.
Game over	Off	Shows <i>Game over</i> . Space restarts the game.

Families

A family is a name shared by several Active objects. In Coin Quest, the Coin and the Star are *Treasure*, and the Blob and the Spikes are *Baddies*. Then one event, *Hero collides with any Baddies*, works for all of them.

- Right-click an object and choose **Families....** Tick a family, or type a new name and click **Add**.

- Families appear in the New condition box and in object lists, written *any Baddies*. A family used in an event gets its own column.
- Actions on a family act on the members the conditions picked. *Destroy any Treasure* removes only the coin or star the Hero touched.

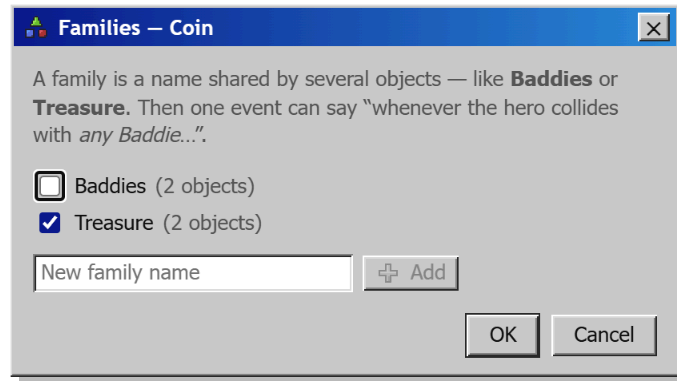


FIG 11.5 The Families box for the Coin: Treasure is ticked

■ Global values

Global values are numbers the whole game remembers. Unlike a score, you name them, and they carry on from frame to frame: the level number, gems collected, the best score.

- Add them in **Game set-up** under **Global values**, with a name and a starting number. Any setting that asks for a global value also has a **New...** button.
- **Actions**, under **Special** ► **Global values: Set, Add to, Subtract from, Save on this computer** and **Load from this computer**.
- **Condition: Compare a global value. Expression:** `Global("Best")`.
- **Save on this computer** keeps the value in the browser, even after the game is closed. **Load** gets it back next time. That is how Coin Quest remembers its best score. Each game title keeps its own saved values.

Game set-up

Title: Coin Quest

Author: the Whenever examples

Window size: 640 × 480 ☐ resize frames too

Border colour: Palette ▾

Players start with 0 points and 3 lives

Options:

- ☒ Game has a menu bar (Game / Options / Help)
- ☒ Crisp pixel scaling
- ☒ Panic key (F5)

Touch joystick: on touch screens ▾

Players' controls

	Up	Down	Left	Right	Fire 1	Fire 2
Player 1	Up arrow	Down arrow	Left arrow	Right arrow	Space bar	Ctrl
Player 2	W	S	A	D	Q	E
Player 3	I	K	J	L	U	O
Player 4	Numpad 8	Numpad 5	Numpad 4	Numpad 6	Numpad 7	Numpad 9

Global values (numbers the whole game remembers)

Best starts at 0

Use them for things like Level, Gems or High score. They carry on from frame to frame.

How to play (shown from the game's Help menu)

Run with the left and right arrows, jump with Space, climb with up and down.
 Collect the treasure, dodge the baddies and reach the flag!

On a tablet, use the on-screen joystick and button 1.

FIG 11.8 Game set-up for Coin Quest, with the global value Best

■ Touch joystick

On a tablet or phone, a game shows an on-screen joystick and two buttons, **1** and **2**. They press Player 1's direction keys and fire keys.

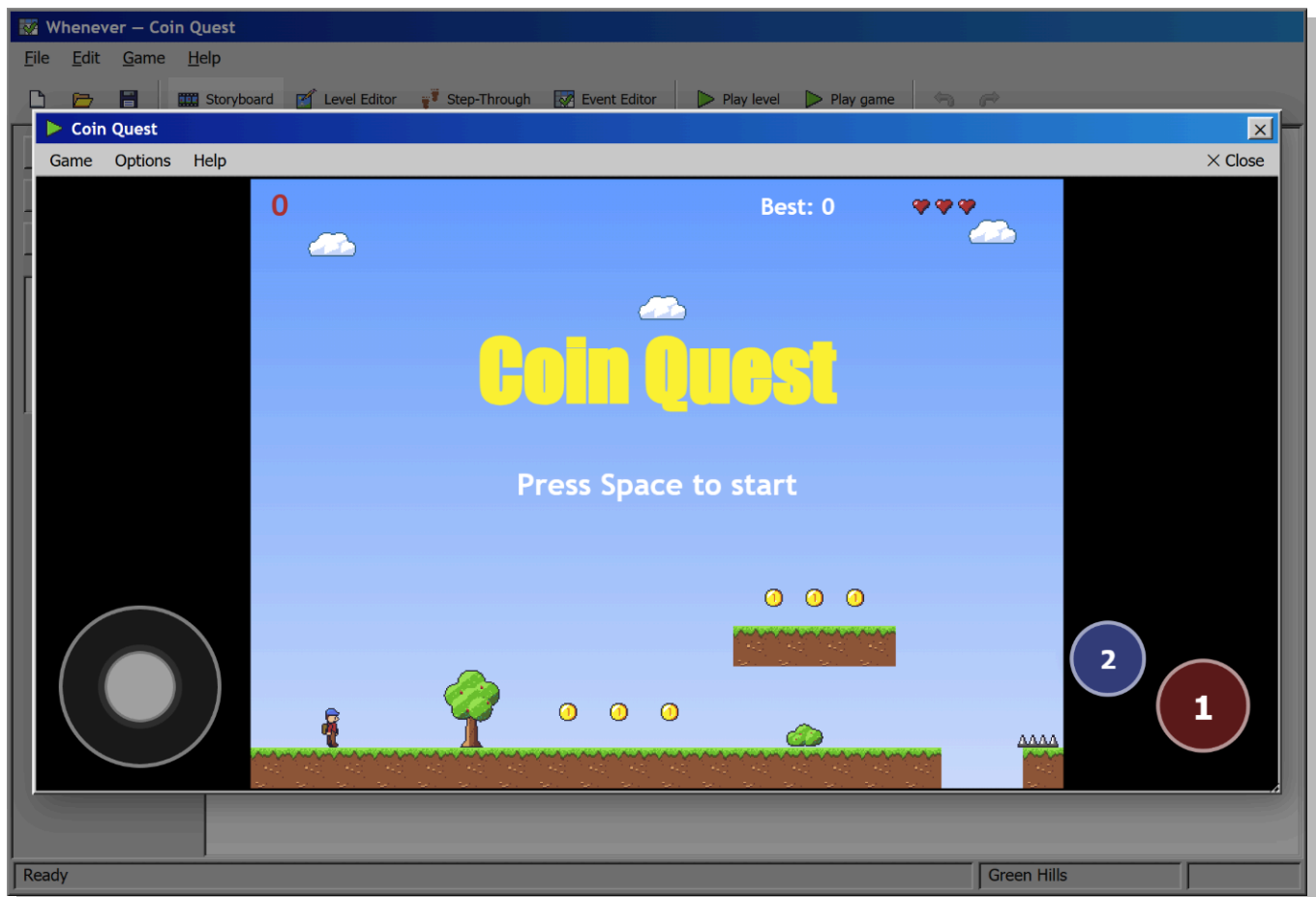


FIG 11.7 Coin Quest on a touch screen, with the joystick on the left and buttons 1 and 2 on the right

Game set-up ▶ **Touch joystick** decides when it appears. *On touch screens* (the usual choice) shows it on touch devices, when the game has an object steered by the keys or tests keys or the joystick. *Always* and *never* override that.

If a screen waits for a key, such as *Press Space to start*, make that key Player 1's **Fire 1** in Game set-up. Then button 1 presses it too. Coin Quest does this.

■ Opening a web page

Special ▶ **The internet** ▶ **Open a web page** opens an address in a new browser tab: your website, a credits page, the next chapter of a story. Browsers only allow new tabs straight after a click or key press, so use it with a condition such as **User clicks on an object**.

12

Finding bugs

When a game doesn't do what you meant, the debugger shows which events are running and what the numbers are. It sits at the bottom of the Step-Through Editor's panel.

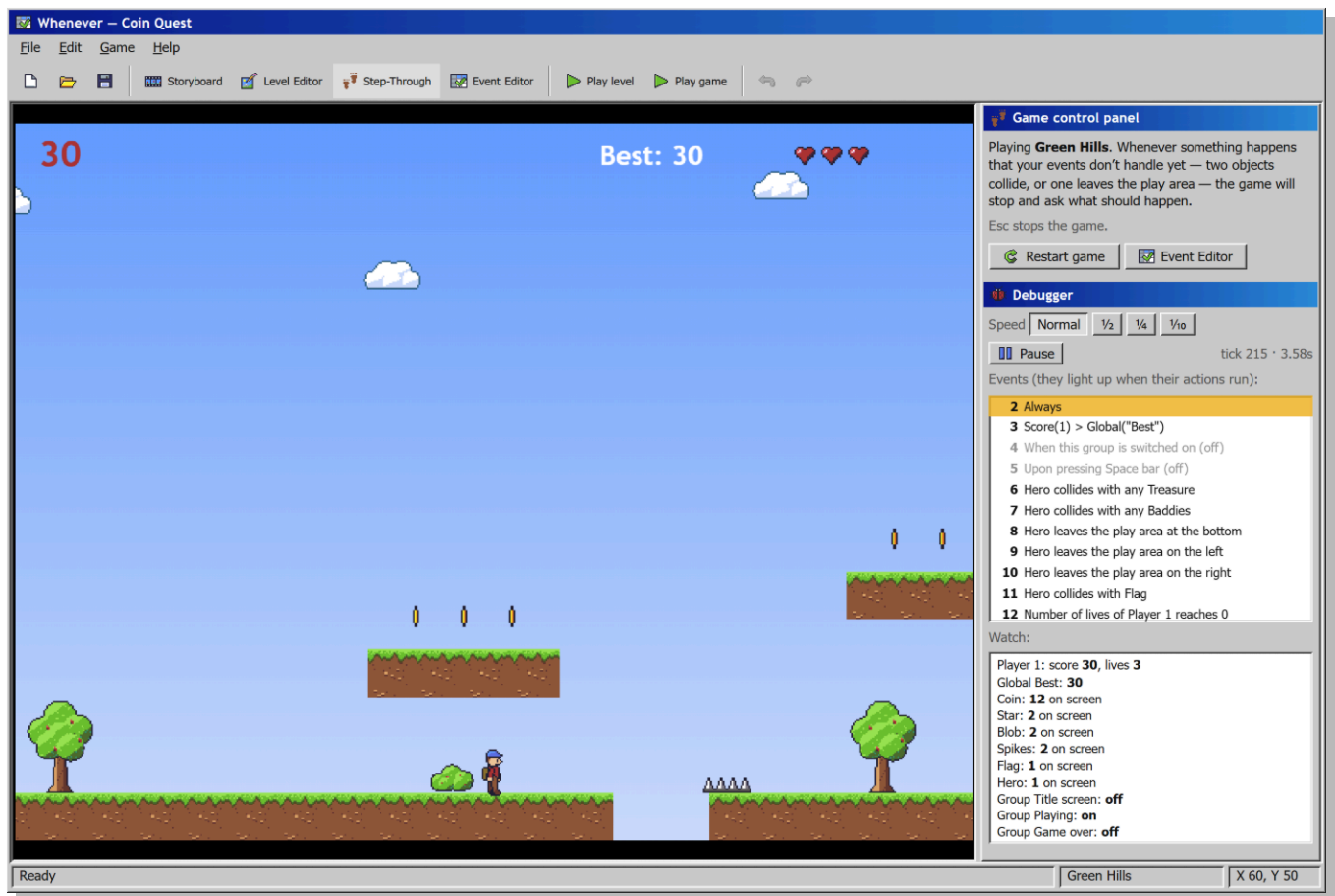


FIG 12.1 The debugger while Coin Quest plays: event 2 glowing, and the Watch list

■ The controls

Control	What it does
Speed: Normal, ½, ¼, 1/10	Plays in slow motion.
Pause	Freezes the game so you can look around.
One tick	Runs exactly one tick (one sixtieth of a second) and stops again.
Continue	Carries on playing.
tick 221 · 3.68s	How long the frame has been running.

■ Events that light up

The debugger lists every event, with the same number it has in the Event Editor. Comment rows are left out, so a number may be missing. An event glows amber when its actions run. Disabled events, and events in a group that is switched off, are faded and marked (*off*).

■ Watch

The **Watch** list shows Player 1's score and lives, every global value, how many of each Active object there are, and which groups are on.

■ A way to hunt bugs

- 1 Say exactly what should happen, and what happens instead.
- 2 Find the event that should make it happen, and watch it while you play.
- 3 If it never lights up, its conditions are never all true together. Check each one, and check its group is on.
- 4 If it glows all the time, a black condition is repeating it. Use a green condition, or add **Only one action when event loops**.
- 5 If it lights up but nothing seems to change, another event may undo it, or its action may be acting on a different copy.
- 6 Slow the game down, or pause it and step one tick at a time through the moment it goes wrong.

Two more tricks: **Disable event** switches an event off while you test, and a Text object with **Set text to a value...** can show any number on the screen.

■ Common bugs

What you see	Likely cause	What to do
The score shoots up by hundreds.	A black condition, such as <i>is overlapping</i> , repeats every tick.	Use <i>collides with</i> , or add Only one action when event loops .
Every enemy vanishes when one is hit.	The action is on an object the conditions didn't test.	Make a condition test that object, such as <i>Bullet collides with Enemy</i> .
An event never happens.	Its conditions can't all be true at once, or its group is off.	Check the conditions one at a time; check the groups in Watch.
Things happen in the wrong order.	Events run from top to bottom.	Drag the event higher or lower.
An object won't move.	Moving at start is unticked, or it starts touching a solid backdrop.	Tick it, or move the object, or change the backdrop's Obstacle setting.
A ball won't fall out of the bottom.	The floor is solid.	Right-click the floor: Obstacle ► No .
A <i>Press Space</i> screen ignores a tablet's buttons.	Space isn't one of Player 1's keys.	Make Space Player 1's Fire 1 in Game set-up.

13

Playing and sharing

Test your game at any time with **Play level** (**F7**) or **Play game** (**F8**).
When it is finished, **File** ► **Save stand-alone game...** turns it into a single web page that anyone can play, without Whenever.

■ The play window

The game opens in a window over the editor. Drag its corner to resize it; the game scales to fit. **X** or **Esc** stops it. If the game has a menu bar, it offers:

Menu	Command	Key
Game	New game	F2
Game	Password... (only when a level has a password)	
Game	Pause	Ctrl+P
Game	Quit	Esc
Options	Play samples	Ctrl+S (stand-alone games)
Options	Play music	Ctrl+M
Options	Hide the menu	F8
Help	How to play	F1
Help	About...	

Opening a menu pauses the game. **F5**, the panic key, hides the game behind a blank screen and stops the sound; press it again to come back. When the game ends, a box offers **Play again**.

■ Saving your work

- Whenever saves one game in the browser automatically, a moment after each change. That copy lives only in this browser on this computer, and clearing the browser's data removes it. If a game grows too big for the

browser to keep (pictures and sounds take lots of room), Whenever says so once: from then on, use File ▶ Save game.

- **File ▶ Save game** (`Ctrl+S`) downloads a `.whenever` file. **Save game as...** (`Ctrl+Shift+S`) asks for a new name.
- **File ▶ Load game...** (`Ctrl+O`) opens a `.whenever` or `.json` file.
- **New game, Load game** and **Open an example** replace the game in the browser. Whenever asks first, so you can save.

■ Stand-alone games

Choose **File ▶ Save stand-alone game...** and give the file a name. Whenever downloads a single `.html` file holding your game and a player for it: about 95 KB, plus your pictures and sounds. Pictures and sounds the game doesn't use are left out.

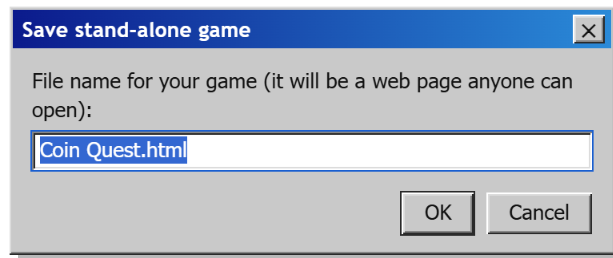


FIG 13.1 Saving a stand-alone game: the file-name box

Double-click the file to play it in any web browser, on any computer, tablet or phone. It fits itself to the window and shows the touch joystick on touch screens. You can email it, copy it to a memory stick, or put it on a website.

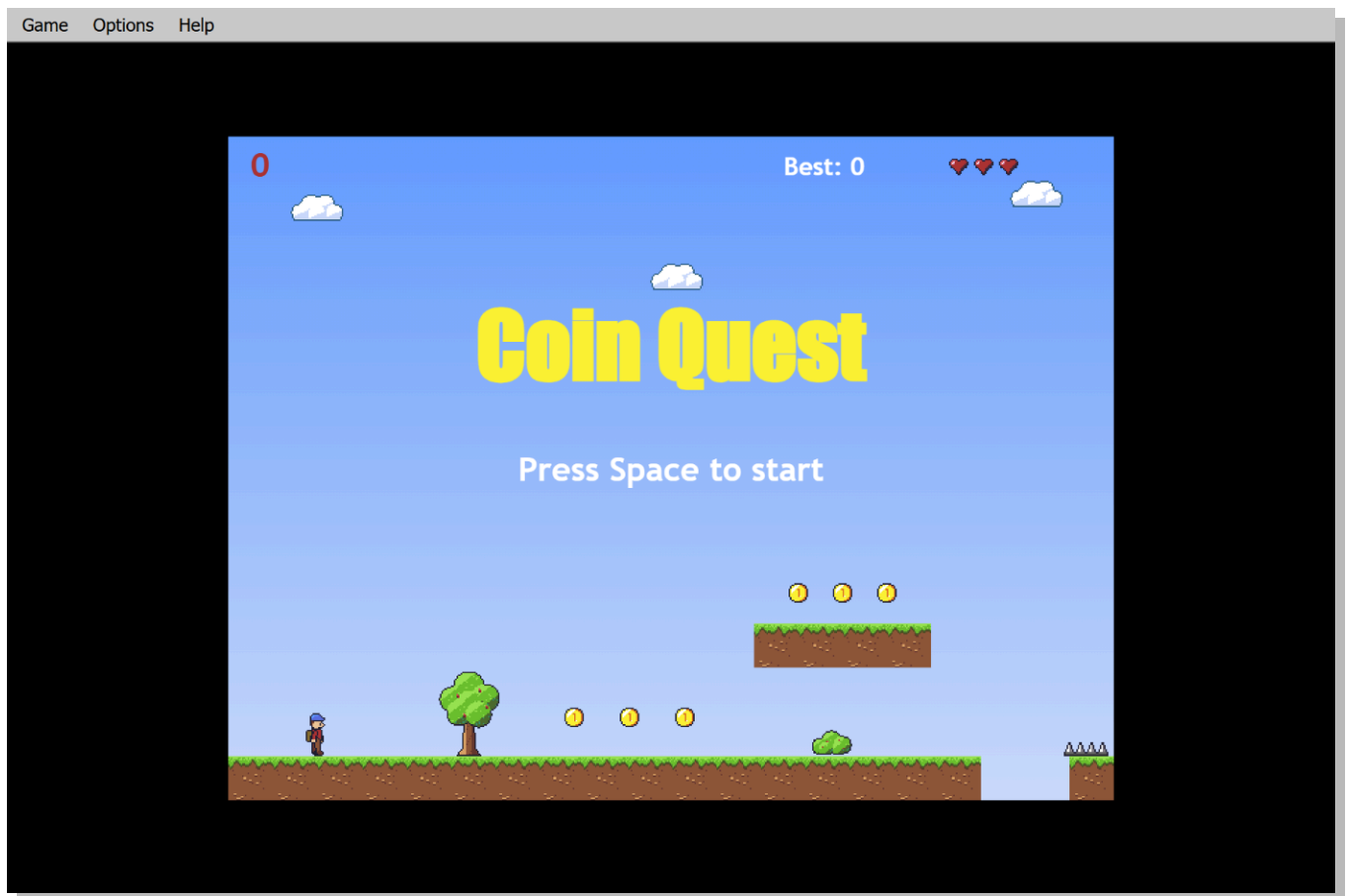


FIG 13.2 Coin Quest as a stand-alone web page, with its Game, Options and Help menus

■ Putting games online

A stand-alone game needs no server. Upload the `.html` file to any website that serves plain files: Neocities, GitHub Pages, or a school website. To put Whenever itself on a website, see *Putting Whenever on a website* in the Reference.

14

Reference

Shortcuts, words, every condition and action, and how to put Whenever itself on a website. The condition and action tables are taken from the program's own list, so they match its menus.

Keyboard shortcuts in the editor

Keys	What they do
<code>Ctrl+S</code> , <code>Ctrl+Shift+S</code>	Save game; save game as.
<code>Ctrl+O</code>	Load game.
<code>Ctrl+Z</code> ; <code>Ctrl+Y</code> or <code>Ctrl+Shift+Z</code>	Undo; redo.
<code>Ctrl+L</code> , <code>Ctrl+E</code>	Level Editor; Event Editor.
<code>F7</code> , <code>F8</code>	Play this level; play the whole game.
<code>Esc</code>	Cancel a box, close a menu, stop placing, stop a game.
<code>Enter</code>	OK in a box.
<code>Ctrl+C</code> , <code>Ctrl+X</code> , <code>Ctrl+V</code> , <code>Ctrl+D</code> , <code>Ctrl+A</code> , <code>Delete</code>	Copy, cut, paste, duplicate, select all, delete (Level Editor).

The keys for a running game are in *Playing and sharing*.

Words

Word	Meaning
Frame	One screen or level of a game.
Object	A thing in the game. Every copy of an object on the play area shares its pictures, movement and events.
Event	A rule: conditions on the left, actions in the boxes.
Condition	Something to test, such as a collision or a key press.
Action	Something to do, such as destroy, bounce or add to the score.
Moment (green)	A condition true for one instant.
While true (black)	A condition true for as long as something lasts.
Tick	One pass through the game: everything moves once and the events are read once. There are sixty a second.
Hot spot	The point of an object that its X and Y position refer to.
Action point	Where an object's shots and created objects appear.
Obstacle	A backdrop moving objects can't pass: a wall, a platform or a ladder.
Alterable value	A number each copy of an object remembers.
Global value	A number the whole game remembers, across frames.
Group	A set of events you can switch on and off.
Family	A name shared by several objects, so one event covers them all.

Conditions

Object is the object whose icon you clicked. Words in italics are settings.

Icon	Menu	Reads as	Kind
Special	Compare two general values	<i>value</i> =, <, >... <i>value</i>	While true
Special	Always / Never ▶ Always	Always	While true
Special	Always / Never ▶ Never	Never	While true
Special	Limit conditions ▶ Run this event once	Run this event once	Limit
Special	Limit conditions ▶ Only one action when event loops	Only one action when event loops	Limit
Special	Limit conditions ▶ Restrict actions for...	Restrict actions for <i>time</i> seconds	Limit

Icon	Menu	Reads as	Kind
Special	Global values ▶ Compare a global value	Global value <i>global</i> =, <, >... <i>value</i>	While true
Special	Groups of events ▶ Is a group of events switched on?	Group <i>group</i> is switched on	While true
Special	Groups of events ▶ When this group is switched on	When this group is switched on	Moment
Special	Groups of events ▶ After this group has been on for...	This group has been on for <i>time</i> seconds	Moment
Storyboard controls	Start of frame	Start of frame	Moment
Storyboard controls	End of frame	End of frame	Moment
Timer	Every...	Every <i>time</i> seconds	Moment
Timer	Is the timer equal to a certain value?	Timer reaches <i>time</i> seconds	Moment
Timer	Compare the timer to a value	Timer =, <, >... <i>time</i> seconds	While true
Create new objects	Compare to total number of objects	Total number of objects =, <, >... <i>value</i>	While true
Create new objects	Test for no objects in a zone	No objects in <i>zone</i>	While true
Create new objects	Compare to number of objects in a zone	Number of objects in <i>zone</i> =, <, >... <i>value</i>	While true
Mouse & keyboard	The keyboard ▶ Upon pressing a key	Upon pressing <i>key</i>	Moment
Mouse & keyboard	The keyboard ▶ Repeat while key is pressed	Repeat while <i>key</i> is pressed	While true
Mouse & keyboard	The mouse ▶ Check for mouse pointer in a zone	Mouse pointer is in <i>zone</i>	While true
Mouse & keyboard	The mouse ▶ Check for mouse pointer over an object	Mouse pointer is over <i>object</i>	While true
Mouse & keyboard	The mouse ▶ User clicks	User clicks with <i>button</i> button	Moment
Mouse & keyboard	The mouse ▶ User clicks within a zone	User clicks with <i>button</i> button in <i>zone</i>	Moment
Mouse & keyboard	The mouse ▶ User clicks on an object	User clicks with <i>button</i> button on <i>object</i>	Moment

Icon	Menu	Reads as	Kind
Players	Joystick ▶ Pressed...	<i>Player presses joystick</i>	Moment
Players	Joystick ▶ Repeat while held...	Repeat while <i>Player</i> holds <i>joystick</i>	While true
Players	Compare to player's score	Score of <i>Player</i> =, <, >... <i>value</i>	While true
Players	Compare to player's number of lives	Lives of <i>Player</i> =, <, >... <i>value</i>	While true
Players	When number of lives reaches zero	Number of lives of <i>Player</i> reaches 0	Moment
Active objects	Collisions ▶ Another object	<i>Object</i> collides with <i>another object</i>	Moment
Active objects	Collisions ▶ Overlapping another object	<i>Object</i> is overlapping <i>another object</i>	While true
Active objects	Collisions ▶ Backdrop	<i>Object</i> collides with the backdrop	Moment
Active objects	Movement ▶ Is the object stopped?	<i>Object</i> is stopped	While true
Active objects	Movement ▶ Is the object bouncing?	<i>Object</i> is bouncing	While true
Active objects	Movement ▶ Compare speed to a value	Speed of <i>Object</i> =, <, >... <i>value</i>	While true
Active objects	Movement ▶ Has it reached a node in the path?	<i>Object</i> has reached a node in the path	Moment
Active objects	Movement ▶ Has it reached the end of its path?	<i>Object</i> has reached the end of its path	Moment
Active objects	Animation ▶ Which animation is playing?	<i>Object's</i> animation is <i>animation</i>	While true
Active objects	Animation ▶ Has an animation finished?	<i>Object's</i> animation has finished	Moment
Active objects	Animation ▶ Compare current frame to a value	Animation frame of <i>Object</i> =, <, >... <i>value</i>	While true
Active objects	Position ▶ Test position of the object	<i>Object</i> leaves, enters, is inside or is outside the play area	Moment
Active objects	Position ▶ Is the object in a zone?	<i>Object</i> is in <i>zone</i>	While true
Active objects	Direction ▶ Compare direction of the object	<i>Object</i> is facing <i>directions</i>	While true
Active objects	Pick or count ▶ Pick one at random	Pick a <i>Object</i> at random	While true
Active objects	Pick or count ▶ Have all been destroyed?	Have all <i>Object</i> been destroyed?	While true
Active objects	Pick or count ▶ Compare to the number of these objects	Number of <i>Object</i> =, <, >... <i>value</i>	While true
Active objects	Pick or count ▶ Test for none in a zone	No <i>Object</i> in <i>zone</i>	While true
Active objects	Pick or count ▶ Compare to the number in a zone	Number of <i>Object</i> in <i>zone</i> =, <, >... <i>value</i>	While true

Icon	Menu	Reads as	Kind
Active objects	Values ▶ Compare to an alterable value	<i>alterable value of Object =, <, >... number</i>	While true
Any object	Position ▶ Compare X position to a value	X position of <i>Object</i> =, <, >... <i>value</i>	While true
Any object	Position ▶ Compare Y position to a value	Y position of <i>Object</i> =, <, >... <i>value</i>	While true
Any object	Visibility ▶ Is the object visible?	<i>Object</i> is visible	While true
Any object	Visibility ▶ Is the object invisible?	<i>Object</i> is invisible	While true
Counter	Compare the counter to a value	<i>Object</i> =, <, >... <i>value</i>	While true
Text	Compare the current paragraph number	Paragraph of <i>Object</i> =, <, >... <i>value</i>	While true
Question	Is the answer correct?	<i>Object</i> : the answer is correct	Moment
Question	Is the answer wrong?	<i>Object</i> : the answer is wrong	Moment
Question	Is the answer equal to a certain value?	<i>Object</i> : answer <i>value</i> is chosen	Moment

■ Actions

Column	Menu	Reads as
Special	Global values ▶ Set a global value	Set global value <i>global</i> to <i>value</i>
Special	Global values ▶ Add to a global value	Add <i>value</i> to global value <i>global</i>
Special	Global values ▶ Subtract from a global value	Subtract <i>value</i> from global value <i>global</i>
Special	Global values ▶ Save on this computer	Save global value <i>global</i> on this computer
Special	Global values ▶ Load from this computer	Load global value <i>global</i> from this computer
Special	Groups of events ▶ Switch a group on	Switch on group <i>group</i>
Special	Groups of events ▶ Switch a group off	Switch off group <i>group</i>
Special	The internet ▶ Open a web page	Open the web page <i>address</i>
Storyboard controls	Next frame	Go to the next frame
Storyboard controls	Previous frame	Go to the previous frame
Storyboard controls	Jump to frame...	Jump to frame <i>frame</i>
Storyboard controls	Restart the current frame	Restart this frame
Storyboard controls	Restart the game	Restart the game
Storyboard controls	End the game	End the game

Column	Menu	Reads as
Storyboard controls	Scrolling ▶ Centre the window on an object	Centre the window on <i>object</i>
Storyboard controls	Scrolling ▶ Scroll the window to a position	Scroll the window to (X, Y)
Create new objects	Create object	Create <i>object</i> at position
Players	Score ▶ Set score	Set score of <i>Player</i> to value
Players	Score ▶ Add to score	Add <i>value</i> to the score of <i>Player</i>
Players	Score ▶ Subtract from score	Subtract <i>value</i> from the score of <i>Player</i>
Players	Number of lives ▶ Set number of lives	Set lives of <i>Player</i> to <i>value</i>
Players	Number of lives ▶ Add to number of lives	Add <i>value</i> to the lives of <i>Player</i>
Players	Number of lives ▶ Subtract from number of lives	Subtract <i>value</i> from the lives of <i>Player</i>
Players	Affect input ▶ Ignore control	Ignore the controls of <i>Player</i>
Players	Affect input ▶ Restore control	Restore the controls of <i>Player</i>
Sound	Play sample	Play sample <i>sound</i>
Sound	Play and loop sample	Play and loop sample <i>sound</i>
Sound	Stop a sample	Stop sample <i>sound</i>
Sound	Stop any sample playing	Stop all samples
Sound	Play music	Play music <i>sound</i>
Sound	Play and loop music	Play and loop music <i>sound</i>
Sound	Stop any music playing	Stop the music
Active objects	Movement ▶ Stop	Stop <i>Object</i>
Active objects	Movement ▶ Start	Start <i>Object</i>
Active objects	Movement ▶ Set speed...	Set speed of <i>Object</i> to <i>value</i>
Active objects	Movement ▶ Set maximum speed...	Set maximum speed of <i>Object</i> to <i>value</i>
Active objects	Movement ▶ Wrap around play area	Make <i>Object</i> wrap around the play area
Active objects	Movement ▶ Bounce	Bounce <i>Object</i>
Active objects	Movement ▶ Reverse	Reverse <i>Object</i>
Active objects	Animation ▶ Stop	Stop the animation of <i>Object</i>

Column	Menu	Reads as
Active objects	Animation ▶ Start	Start the animation of <i>Object</i>
Active objects	Animation ▶ Change ▶ Animation sequence...	Change animation of <i>Object</i> to <i>animation</i>
Active objects	Animation ▶ Change ▶ Direction of animation...	Change animation direction of <i>Object</i> to <i>directions</i>
Active objects	Animation ▶ Change ▶ Speed of animation...	Set animation speed of <i>Object</i> to <i>value</i>
Active objects	Animation ▶ Restore ▶ Animation sequence	Restore the animation of <i>Object</i>
Active objects	Animation ▶ Restore ▶ Direction of animation	Restore the animation direction of <i>Object</i>
Active objects	Animation ▶ Restore ▶ Speed of animation	Restore the animation speed of <i>Object</i>
Active objects	Position ▶ Swap position with another object	Shuffle the positions of <i>Object</i>
Active objects	Direction ▶ Select direction...	Set direction of <i>Object</i> to <i>directions</i>
Active objects	Direction ▶ Look in direction of...	Make <i>Object</i> look at <i>object</i>
Active objects	Direction ▶ Turn by...	Turn <i>Object</i> by <i>value</i> directions
Active objects	Shoot an object...	<i>Object</i> shoots <i>object</i> how at speed <i>speed</i>
Active objects	Values ▶ Set value	Set <i>alterable value</i> of <i>Object</i> to <i>number</i>
Active objects	Values ▶ Add to value	Add <i>number</i> to <i>alterable value</i> of <i>Object</i>
Active objects	Values ▶ Subtract from value	Subtract <i>number</i> from <i>alterable value</i> of <i>Object</i>
Active objects	Values ▶ Set to a random number	Set <i>alterable value</i> of <i>Object</i> to a random number below <i>number</i>
Active objects	Values ▶ Spread a value (0, 1, 2...)	Spread <i>alterable value</i> of <i>Object</i> starting at <i>number</i>
Any object	Position ▶ Select position...	Set position of <i>Object</i> to <i>position</i>
Any object	Position ▶ Set X coordinate...	Set X position of <i>Object</i> to <i>value</i>
Any object	Position ▶ Set Y coordinate...	Set Y position of <i>Object</i> to <i>value</i>
Any object	Visibility ▶ Make object invisible	Make <i>Object</i> invisible
Any object	Visibility ▶ Make object reappear	Make <i>Object</i> reappear
Any object	Visibility ▶ Flash object	Flash <i>Object</i> every <i>time</i> seconds

Column	Menu	Reads as
Any object	Order ▶ Bring to front	Bring <i>Object</i> to the front
Any object	Order ▶ Send to back	Send <i>Object</i> to the back
Any object	Destroy	Destroy <i>Object</i>
Text	Display text...	Display <i>Object</i> at <i>position</i>
Text	Erase text	Erase <i>Object</i>
Text, Counter, Score, Lives	Set colour of text...	Set the colour of <i>Object</i> to <i>colour</i>
Text	Set paragraph...	Show paragraph <i>value</i> of <i>Object</i>
Text	Next paragraph	Show the next paragraph of <i>Object</i>
Text	Previous paragraph	Show the previous paragraph of <i>Object</i>
Text	Set text to a value...	Set the text of <i>Object</i> to <i>value</i>
Counter	Set counter	Set <i>Object</i> to <i>value</i>
Counter	Add to counter	Add <i>value</i> to <i>Object</i>
Counter	Subtract from counter	Subtract <i>value</i> from <i>Object</i>
Question	Ask the question	Ask <i>Object</i>

Files

File	What it is
<code>.whenever</code>	A saved game, made by File ▶ Save game . Load it with File ▶ Load game...
<code>.json</code>	The same thing under another name. Some web hosts accept <code>.json</code> but not <code>.whenever</code> .
<code>.html</code>	A stand-alone game, made by File ▶ Save stand-alone game... Opens in any browser.

Putting Whenever on a website

Whenever is a set of plain files, so any static web host can serve it, including a subfolder of a Neocities site. Building it needs Node.js on your computer.

- 1 In the Whenever folder, run `npm install`, then `npm run build`.
- 2 Upload everything inside the new `dist` folder: `index.html`, `player.js`, `favicon.svg` and the `assets` folder. Any folder name works, such as `/whenever/`.
- 3 To check the build first, run `npm run preview`. It uses the same security rules as a free Neocities site. Add `-- --host` to try it on a phone or tablet on the same Wi-Fi.

Games are saved in each person's browser, not on the website, so a whole class can use one copy of the site. A browser holds one autosaved game at a time: on shared computers, have pupils use File ► Save game at the end of each session.