suma+ 70
Artículos
Julio 2012
pp. 65-73

Matemáticas para jugar

Mario Rafael Gil Martín

Si los alumnos son capaces de analizar, razonar y comunicar eficazmente cuando enuncian, formulan y resuelven problemas matemáticos que son reales y cotidianos decimos que son competentes en Matemáticas. En este artículo se justifica la necesidad de resolver problemas cuando diseñamos y creamos nuestros propios juegos de ordenador. ¿Son conscientes nuestros alumnos de las Matemáticas que hay que saber para crearlos?

Palabras clave: Innovación didáctica, Resolución de problemas, Juegos, Actitudes, Secundaria.

Mathematics for Playing
It is said that students are mathematically competent if they are able to analyse, think over and communicate when they enunciate, formulate and solve real and everyday life mathematical problems. In this article we justify the need to solve problems when we design and create our own computer games. Are our students aware that they need to know maths in order to create them?

Key words: Educational Innovation, Problem Solving, Games, Attitudes, Secondary Education.

Ya no sé qué hacer con los jóvenes. Parecen completamente desmotivados. Ya sé que siempre se puede hacer más, pero pienso que hago todo lo que puedo: busco problemas atractivos, intento hablar con ellos, trato de adaptarme a su ritmo... «Creo que, en parte, tiene que ver con los problemas que les proponemos. Normalmente no les interesan. No les ven utilidad futura. Seguimos pidiéndoles que hallen la altura de un castillo conocido un cierto ángulo de visión. Pero ¡muy pocos viven en castillos! Algunos ni siquiera han visto uno» (Chamoso y Rawson, 2003).

Nuestros alumnos, nativos digitales, están familiarizados con las tecnologías de la información y la comunicación, pero usan principalmente internet para entretenerse y, secundariamente, para aprender. Según un estudio que puede consultarse en el ITE (Instituto de Tecnologías Educativas), el 35% de los jóvenes lo usa para chatear y el 27% para jugar.

«Los juegos constituyen, un buen instrumento para desarrollar el idioma matemático, para hacer matemáticas, para interiorizar los procesos propios del pensar matemático. Y además, por sí mismos, y comparándolos con las actividades habituales de las clases de Matemáticas, tienen un atractivo aña-

Artículo recibido en *Suma* en octubre de 2010 y aceptado en noviembre de 2011

dido: apetece dedicarse a ellos. No hay que empujar a los alumnos para que comiencen al análisis de los mismos; lo hacen voluntariamente» (Corbalán, 1994).

Durante las horas de guardia en mi centro, he constatado cómo los alumnos quieren usar los ordenadores normalmente para jugar online. Suelen ser juegos de acción y están realizados con tecnología Flash. Les divierte pasar el rato jugando, pero ¿se han preguntado cómo se hacen? Y, lo más sorprendente para ellos aún, ¿saben las Matemáticas que hay detrás de esos juegos?

Así surgió la idea de este artículo donde diseño un sencillo juego interactivo en *Flash* y hago un recorrido por distintas cuestiones matemáticas que necesitamos saber para comprender y avanzar en la creación del juego.

El propósito de los problemas es planteárselos a los alumnos, para que sepan hacer y aprendan a aprender en un contexto real y relevante para ellos.

Según Carrillo (2001) la resolución de problemas debe estar presente en la enseñanza. Y no lo debe hacer de forma anecdótica, sino como caracterizadora del proceso de formación matemática de los alumnos.

Es sorprendente que, a pesar de la simplicidad del juego, hayan surgido en él conceptos de Aritmética, Álgebra, Geometría, Funciones y Probabilidad. Es decir, ¡todos los bloques que incluye cualquier libro de texto de Matemáticas de secundaria!

A lo largo de este trabajo incluyo código, que debe entenderse más bien como pseudocódigo, donde doy unas pautas lógicas de cómo se programaría. No es mi propósito que la sintaxis del lenguaje Actionscript que usa Flash desenfoque los problemas matemáticos, verdadero eje vertebrador de este trabajo.

Una versión del juego y del archivo fuente necesario para modificarlo estarán disponibles para todos aquellos compañeros que quieran experimentar con sus alumnos en clase:

<http://www.juntadeandalucia.es/ averroes/centros-tic/ 21700502/helvia/sitio/upload/index.swf>

Cargando el juego

Nuestros alumnos suelen jugar *on-line*, lo que hace que el juego tarde un tiempo en cargarse en el navegador. Este tiempo depende, principalmente, de la velocidad de conexión y de la respuesta del ordenador que se usa. Y es habitual que el juego, en esa espera, incluya un cargador o *preloader* que se expresa mediante un porcentaje. ¡Antes de empezar a jugar ya estamos usando las Matemáticas!

Necesitamos saber el tamaño en *bytes* del juego, que llamaremos total, y definir dos variables que llamaremos cargados y porcentaje y que se relacionan como sigue:

$$\text{Porcentaje} = \text{Parte entera}\left(\frac{\text{cargados}\cdot 100}{\text{total}}\right)$$

Normalmente, el cargador se representa gráficamente con una barra que crece y un porcentaje. La función parte entera, que en Flash es «Math.floor», tiene por objeto que el porcentaje mostrado no tenga decimales.

El escenario del juego

El juego se suele desarrollar en una región rectangular, conocida en Flash como el escenario, donde se ubican los objetos y personajes del juego y donde transcurren las acciones.

Es fundamental elegir un sistema de coordenadas que nos permita identificar de forma única cada punto de nuestro escenario de juego, esto es lo que en Matemáticas llamamos plano cartesiano, eligiéndose así un sistema de referencia formado por dos ejes perpendiculares y un punto de origen.

Pero según la situación de estudio, puede resultar más útil elegir los ejes de forma diferente a como los usamos normalmente en nuestras clases. Así en Física, cuando se estudia la caída de objetos por un plano inclinado en un campo gravitatorio, se suele tomar un sistema de referencia ligado al plano y se eligen como ejes los que vienen dados por la dirección de movimiento del objeto y la recta normal al plano.

En *Flash*, el sistema de referencia que se usa cambia respecto al que usamos en Matemáticas, lo que a mi entender es más interesante para trabajar estos conceptos con los alumnos.

Vamos a considerar que nuestro juego se desarrolla en un rectángulo de dimensiones a y b, de manera que la región a la que llamamos escenario queda definida del siguiente modo:

$$R = \left\{ (x, y) \in \mathbb{R}^2 ; 0 \leq x \leq a ; 0 \leq y \leq b \right\}$$

Movimiento del personaje

En Flash, para referirnos a las coordenadas *(x,y)* de un objeto debemos escribir:

```
objeto._x     objeto._y
```

Nuestro personaje, héroe en adelante, va a ser un cuadrado de lado *L* que vamos a situar inicialmente en el centro del escenario y que el jugador va a poder mover mediante las teclas de los cursores en las cuatro direcciones (norte, sur, este y oeste). La siguiente sentencia condicional muestra uno de los movimientos:

```
CODIGO PARA MOVER AL HÉROE
EN LA DIRECCIÓN POSITIVA DEL EJE X
Si el jugador pulsa la tecla derecha, entonces:
heroe._x=heroe ._x+1
```

¿Tiene sentido esta igualdad en Matemáticas? Algebraicamente representa una ecuación sin solución, sin embargo, aquí y en la mayoría de los lenguajes de programación, más que una igualdad es una asignación y debe entenderse dinámicamente, queriendo decir en realidad:

```
Si el jugador pulsa la tecla de la derecha debes mover el
personaje 1 pixel a la derecha, de manera que:
_xnueva=_xanterior+1
```

Esto nos recuerda a una sucesión definida por recurrencia. En este caso tendríamos una progresión aritmética de diferencia *d=1* y que en la práctica del juego originaría un movimiento rectilíneo uniforme (MRU) en el eje x positivo. Hay que caer en la cuenta de que cuando se pulse la tecla de ir hacia abajo debemos incrementar la coordenada *y*. Esto es debido al sistema de referencia que usa *Flash*.

Usemos la variable *d* para expresar estos cambios o desplazamientos, quedando la movilidad del héroe como sigue:

```
CÓDIGO PARA MOVER AL HÉROE
Si el jugador pulsa la tecla derecha, entonces:
heroe._x=heroe ._x+d
Si el jugador pulsa la tecla izquierda, entonces:
heroe._x=heroe ._x-d
Si el jugador pulsa la tecla arriba, entonces:
heroe._y=heroe ._y+d
Si el jugador pulsa la tecla abajo, entonces:
heroe._y=heroe ._y-d
```

Así, en cualquiera de las direcciones, nuestro personaje ocuparía posiciones que vienen dadas por progresiones aritméticas de primer orden de diferencia *d*.

Problema: ¿A qué velocidad se mueve nuestro personaje?

Cuando creamos un archivo nuevo en Flash, automáticamente se crea una velocidad de reproducción de 12 fotogramas por segundo (fps, en lenguaje técnico), aunque es modificable. Y, de hecho, como algunos programadores de juegos aconsejan aumentar esta velocidad de fotogramas a 30 fps, así lo hago. Según esta velocidad, si dejamos pulsada una tecla, nuestro héroe se moverá 30 veces en un segundo. Si cada vez que se mueve recorre d píxeles, recorrerá 30 ·d píxeles por segundo. Pero, ¿cuántos píxeles hay en un centímetro? Debemos entonces considerar la resolución y el tamaño del monitor. Acabo de comprobar, mirando en la configuración, que tengo una resolución de 1400 x 1050 píxeles. Esto quiere decir que mi pantalla está dividida en 1400 líneas verticales y 1050 líneas horizontales.

Mi monitor tiene 22 y mide, aproximadamente, 48cm de largo por 26,8cm de ancho. Comprobémoslo aplicando el Teorema de Pitágoras:

$$\text{Diagonal del monitor}=\sqrt{(48\text{ cm})^2+(26{,}8\text{ cm})^2}=$$
$$=\sqrt{3022{,}24\text{ cm}^2}=54{,}97\text{ cm}=21{,}64\text{ pulgadas}$$

¡Es muy aceptable teniendo en cuenta que lo he medido por el exterior! Por tanto, las componentes del vector velocidad de nuestro héroe son:

$$v_x=\frac{\text{espacio}}{\text{tiempo}}=\frac{30d\text{ píxeles}}{1\text{ segundo}}=$$
$$=\frac{30d\text{ píxeles}}{1\text{ segundo}}\cdot\frac{48\text{ cm}}{1400\text{ píxeles}}\cong 1{,}03d\ \frac{\text{cm}}{\text{s}}$$

$$v_y=\frac{\text{espacio}}{\text{tiempo}}=\frac{30d\text{ píxeles}}{1\text{ segundo}}=$$
$$=\frac{30d\text{ píxeles}}{1\text{ segundo}}\cdot\frac{26{,}8\text{ cm}}{1050\text{ píxeles}}\cong 0{,}77d\ \frac{\text{cm}}{\text{s}}$$

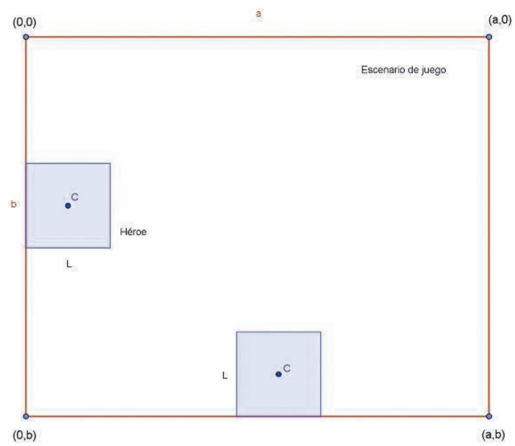
¿Se cumple lo anterior? ¡Experimenta con tu monitor!

Problema: Te has preguntado ¿qué pasaría si el jugador deja el dedo pulsado en algunas de las teclas de dirección?

Efectivamente el personaje saldría de la pantalla de juego y ¡no te echaría de menos! Seguiría moviéndose y en ningún juego ocurre eso. Básicamente propongo dos posibles soluciones.

Solución 1: El personaje no puede salirse de la pantalla de juego, por ejemplo, porque haya paredes que lo impidan.

Hay que tener presente que nuestro héroe es un modesto cuadrado de longitud L y que cuando hablamos de su posición hablamos de su centro. ¿Cómo cambiamos el código que rige el movimiento? Para ello debemos considerar el siguiente dibujo donde se ha representado el cuadrado tangente con el borde del escenario.



Debemos hacer que se pare cuando su centro esté a una distancia del borde menor o igual que la mitad de su lado.

CÓDIGO PARA IMPEDIR QUE
NUESTRO HÉROE ABANDONE
LA PANTALLA DE JUEGO
Si (heroe ._x<= L/2) o (heroe ._x>=a- L/2), entonces que no se mueva.
Si (heroe ._y<= L/2) o (heroe ._y>=b- L/2), entonces que no se mueva.

Solución 2: El escenario tiene condiciones de contorno periódicas.

¿Has jugado alguna vez al comecocos? Los de mi generación tuvimos la suerte de conocer, entre otros, juegos como éste. El comecocos salía por una puerta de la derecha y, por arte de magia, entraba por otra de la izquierda. Los investigadores que estudian la dinámica por computador de sistemas líquidos de N moléculas que interaccionan con un potencial determinado, tienen que limitar el estudio a una celda cúbica muy pequeña, sólo hay que recordar lo grande que es el número de Avogadro (partículas en un mol de cualquier sustancia: $6{,}022 \cdot 10^{23}$). Considerar celdas mayores haría crecer mucho el tiempo de cálculo y, por consiguiente, la obtención de resultados. Entonces, establecen lo que ellos llaman condiciones de contorno periódicas, que consiste en suponer que si una molécula sale por la derecha entra otra por la izquierda. ¿Cómo conseguimos esto en nuestro juego?

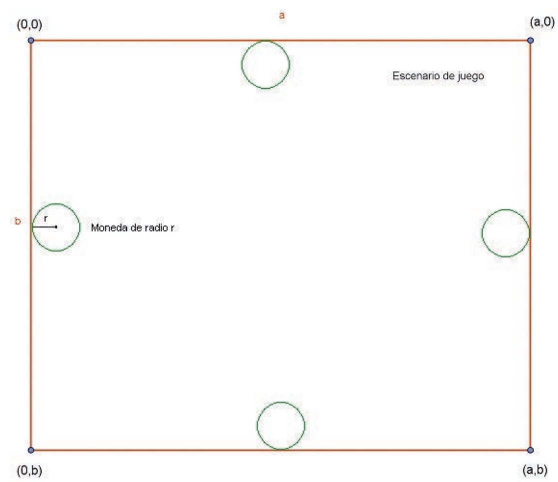
```
CÓDIGO PARA CONSEGUIR
UNA PANTALLA DE JUEGO PERIÓDICA
Si heroe._x< L/2 entonces heroe._x= a- L/2
Si heroe._x> a- L/2 entonces  heroe._x= L/2
Si heroe._y< L/2 entonces heroe._y= b- L/2
Si heroe._y> b- L/2 entonces  heroe._y= L/2
```

Objetivo del juego

Evidentemente, el juego sería muy aburrido si sólo consistiera en mover un cuadrado a velocidad constante por una pantalla rectangular en las cuatro direcciones que marcan los cursores. Vamos a crear un objetivo que consista en recolectar monedas. Estas monedas van a tener un radio r y se van a crear de forma aleatoria en el escenario de juego.

En las calculadoras científicas existe una función implementada para generar un número aleatorio entre cero y uno. Por cierto, ¿le habéis pedido alguna vez a vuestros alumnos que usen esta tecla para simular el lanzamiento de una moneda o el de un dado? Igualmente, en la hoja de cálculo de *OpenOffice*, la que yo uso, esta función se llama ALEATORIO( ); en *Flash* disponemos del método matemático Math.random( ) para el mismo propósito.

Cada vez que nuestro héroe recoja una moneda se generará una nueva en una posición aleatoria de la pantalla de juego. Debemos considerar el rectángulo de juego y el radio *r* de la moneda y, por supuesto, debemos evitar que se posicione fuera de la pantalla de juego. La siguiente imagen muestra la moneda tangente con el límite del escenario de juego.



Esto lo conseguiremos de la siguiente manera:

```
CÓDIGO PARA GENERAR UNA
NUEVA MONEDA CUANDO EL HÉROE RECOJA UNA
mayor=Math.max(L,2*r)

Si (heroe ._x=moneda._x) y (heroe ._y=moneda._y),
                                        entonces:]
moneda._x=Math.random( )*(a-mayor)+mayor/2;
moneda._y=Math.random( )*(b-mayor)+mayor/2;
```

Creamos la variable mayor para comparar los tamaños del héroe y de la moneda con el fin de que la moneda sea generada en un área rectangular del escenario donde el cuadrado pueda llegar. Para que el código sea eficiente la comparación entre el lado del cuadrado y el diámetro de la moneda ha de hacerse sólo una vez durante el juego.

Por ejemplo, si definimos $a=550$, $b=400$, $L=30$, $r=10$ y $mayor=30$, la moneda se situará de tal manera que su centro quede dentro de la siguiente región rectangular:

$$\left\{(x,y)\in\Re^2;15\le x\le 535;15\le y\le 385\right\}$$

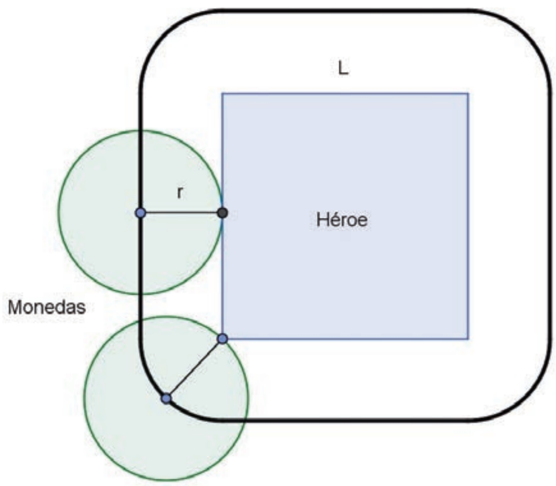
Problema: ¿Qué posibilidades hay de que la moneda se cree encima del personaje? ¿Debemos preocuparnos por ello?

No sería deseable que la moneda se crease encima de nuestro héroe. De hecho, si ganamos puntos recogiendo monedas habríamos sumado puntos sin mover un dedo. Pero, ¿hasta qué punto, matemáticamente, debe preocuparnos que la moneda circular pise a nuestro héroe cuadrangular?

Dado que todas las posibles posiciones de la moneda son equiprobables, la regla de Laplace nos dice que la probabilidad de que eso ocurra viene dada por:

$$\text{Probabilidad de solapamiento moneda-héroe} = \\ = \frac{\text{Superficie de contacto moneda-héroe}}{\text{Superficie posible donde puede crearse la moneda}}$$

Donde, suponiendo que *L>2r*, la superficie posible donde puede aparecer el centro de la moneda es *(a-L)*(b-L)*. La superficie donde habría contacto entre la moneda y nuestro héroe viene dada por la siguiente figura geométrica:



$$\text{Superficie de contacto} = (L+2r)^2 - \left((2r)^2 - \pi r^2\right) = \\ = L^2 + 4Lr + \pi r^2$$

Por tanto la probabilidad sería:

$$P = \frac{L^2 + 4Lr + \pi r^2}{(a-L)\cdot(b-L)}$$

Evaluemos la expresión para las siguientes condiciones: *L=30*, *r=10*, *a=550*, *b=400*.

$$P = \frac{30^2 + 4\cdot 30\cdot 10 + \pi\cdot 10^2}{(550-30)\cdot(400-30)} = 0{,}0125$$

Es decir, sólo en un 1.25 % de los casos se dará ese contacto. ¡No es, por tanto, preocupante!

Sistema de puntuación

Un sistema de puntuación hará más atractivo el juego. Vamos a hacer que obtenga puntos cuando recoja monedas y pierda cuando choque con los bordes del escenario.

Consideremos que gana 5 puntos por moneda recogida y pierde 3 puntos por colisión con las paredes. Bastará modificar el último código como sigue:

CÓDIGO MODIFICADO PARA GANAR PUNTOS

```
Si (heroe._x=moneda._x)
y (heroe ._y=moneda._y), entonces:
moneda._x=Math.random( )*(a-mayor)+mayor/2;
moneda._y=Math.random( )*(b-mayor)+mayor/2;
puntos=puntos+5;
```

CÓDIGO MODIFICADO PARA PERDER PUNTOS

```
Si (heroe ._x<= L/2) o (heroe ._x>=a- L/2) entonces:
que no se mueva y puntos=puntos-3;
Si (heroe ._y<= L/2) o  (heroe ._y>=b- L/2) entonces:
que no se mueva y puntos=puntos-3;
```

Lógicamente, debemos poner un fin al juego. El jugador habrá superado el juego si llega, por ejemplo, a 100 puntos y habrá perdido si su puntuación se hace negativa.

CÓDIGO PARA ACABAR EL JUEGO

```
Si puntos>=100, entonces muestra mensaje:
«Enhorabuena, lo conseguiste».
Si puntos<0, entonces muestra mensaje:
«Has perdido, otra vez será».
```

Problema: ¿Puede ser la puntuación del jugador 17?, ¿qué puntuaciónes pueden darse en el juego?

La puntuación conseguida viene dada por la expresión algebraica Puntos=5m-3c, donde m y c indican el número de monedas recolectadas y el número de colisiones con las paredes, respectivamente. La puntuación será 17 si la siguiente ecuación tiene solución en N:

$$-3c + 5m = 17 \ ; \ c, m \in \mathbb{N}$$

Se trata pues de una ecuación diofántica lineal de la forma ax+by=n que tiene soluciones enteras si y sólo si d=m.c.d.(a,b) divide n. En nuestro caso, puesto que a y b son primos entre sí, d=1 y siempre va a dividir a n. Por lo tanto, no sólo se obtendrá alguna vez la puntuación 17, sino cualquier otra, como comprobaremos más adelante.

Con los alumnos podemos usar una hoja de cálculo para buscar soluciones, para lo cual despejamos *m*, por ejemplo:

$$m = \frac{17 + 3c}{5}$$

Y damos valores a *c* en busca de soluciones naturales:

c	m	puntos
0	3,4	17
1	4	17
2	4,6	17
3	5,2	17
4	5,8	17
5	6,4	17
6	7	17
7	7,6	17
8	8,2	17
9	8,8	17
10	9,4	17
11	10	17
12	10,6	17
13	11,2	17
14	11,8	17
15	12,4	17
16	13	17
17	13,6	17
18	14,2	17
19	14,8	17
20	15,4	17
21	16	17

Observamos que en infinitas ocasiones, durante el desarrollo del juego, puede darse una puntuación de 17. Y las soluciones vienen dadas por los pares:

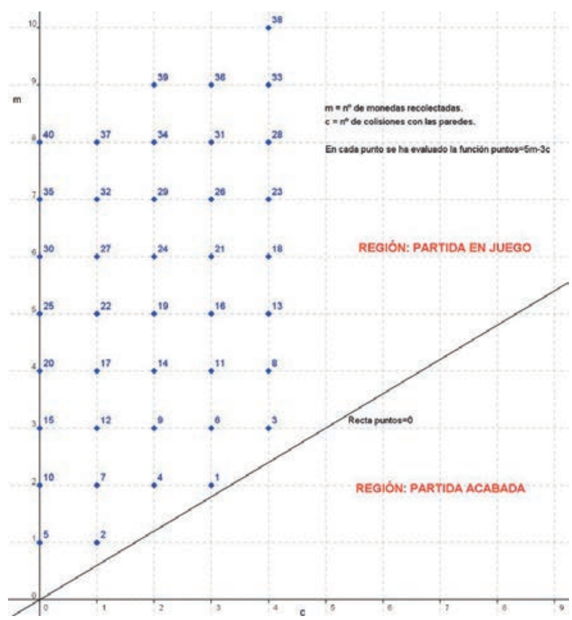
$$(c, m) = (1 + 5n, 4 + 3n) \quad \text{con} \quad n = 0, 1, 2, 3, \ldots$$

¡Otra vez progresiones aritméticas! Efectivamente, la teoría nos dice que la solución general de la ecuación diofántica ax+by=n tiene la forma:

$$\left\{ x_0 + \frac{tb}{d}, y_0 - \frac{ta}{d} \right\}, \ t \in \mathbb{Z}, \ (x_0, y_0): \text{solución particular}$$

Es fácil comprobar que concuerda con lo obtenido en la hoja de cálculo, ya que: $(x_0, y_0) = (1, 4)$, $d = 1$, $a = -3$ y $b = 5$. Pero, ¿se podrá obtener cualquier otra puntuación? Puesto que m.c.d.(5, 3) = 1, donde 5 y 3 son los puntos que se pueden ganar o perder durante el juego, efectivamente se puede obtener cualquier puntuación. Evaluemos la función definida como puntos = $5m - 3c$ en cada par (c, m).

La siguiente gráfica, realizada con Geogebra, muestra los pares (c, m) que permiten obtener los puntos desde 1 hasta 40:



La misma información obtenida con la hoja de cálculo:

c	m	puntos
3	2	1
1	1	2
4	3	3
2	2	4
0	1	5
3	3	6
1	2	7
4	4	8
2	3	9
0	2	10
3	4	11
1	3	12
4	5	13
2	4	14
0	3	15
3	5	16
1	4	17
4	6	18
2	5	19
0	4	20
3	6	21
1	5	22
4	7	23
2	6	24
0	5	25
3	7	26
1	6	27
4	8	28
2	7	29
0	6	30
3	8	31
1	7	32
4	9	33
2	8	34
0	7	35
3	9	36
1	8	37
4	10	38
2	9	39
0	8	40

Obsérvese que basta considerar las familias de puntos obtenidas desde c=0 hasta c=4 para generar cualquier puntuación:

c	m	puntos	
0	1	5	
0	2	10	
0	3	15	
0	4	20	$\{5n; n\in\mathbb{N}\}$
0	5	25	
0	6	30	
0	7	35	
0	8	40	
1	1	2	
1	2	7	
1	3	12	$\{5n+2; n=0,1,2,3,\ldots\}$
1	4	17	
1	5	22	
1	6	27	
1	7	32	
1	8	37	
2	2	4	
2	3	9	
2	4	14	
2	5	19	$\{5n+4; n=0,1,2,3,\ldots\}$
2	6	24	
2	7	29	
2	8	34	
2	9	39	
3	2	1	
3	3	6	
3	4	11	
3	5	16	$\{5n+1; n=0,1,2,3,\ldots\}$
3	6	21	
3	7	26	
3	8	31	
3	9	36	
4	3	3	
4	4	8	
4	5	13	
4	6	18	$\{5n+3; n=0,1,2,3,\ldots\}$
4	7	23	
4	8	28	
4	9	33	
4	10	38	

Sugerencias de trabajo con los alumnos

Las siguientes son algunas actividades matemáticas que se pueden proponer a los estudiantes con relación al juego desarrollado:

A

Cuando el juego se está cargando verás una barra que progresa de 0 % a 100 %. Elige las magnitudes que consideres oportunas y representa gráficamente este cambio.

B

Hemos visto que los movimientos del héroe vienen dados por progresiones aritméticas de diferencia *d*.

¿Qué cambios podríamos hacer para que las progresiones fueran geométricas? ¿Qué razón elegirías? ¿Qué efecto tendría eso en el movimiento?

C

Para dar un poco de emoción vamos a añadir algunos efectos. Estudia cómo añadirías la existencia de viento, rozamiento y gravedad.

D

Si elegimos que la pantalla de juego sea periódica y suponemos que dejamos pulsada la tecla de ir a la derecha durante un largo período de tiempo, ¿cómo representarías la posición que ocupa el héroe en función del tiempo? ¿y su velocidad?

E

Te propongo investigar una tercera solución para que el héroe no se pierda de la pantalla de juego. Cuando choque contra ella apliquemos el principio de acción-reacción y que el héroe rebote.

F

Intercambiemos la geometría de los protagonistas, es decir, el héroe será un círculo de radio r y la moneda será un cuadrado de lado L. ¿Qué cambios habría que hacer?

G

Cambiemos el sistema de puntuación de manera que el héroe gane 6 puntos por moneda recogida y pierda 3 puntos al chocar con las paredes. ¿Se obtendrán alguna vez 17 puntos? ¿Qué puntuaciones se pueden obtener ahora? Usa la hoja de cálculo.

H

Queremos que en pantalla se muestre, además de la puntuación, la distancia al objetivo mediante un texto dinámico que irá cambiando según cambien las posiciones del héroe y de la moneda. El usuario podrá ver a qué distancia se encuentra el héroe de la moneda. ¿Cómo logramos esto?

Referencias bibliográficas

Bujalance, E.; Bujalance, J.; Costa, A. y Martínez, E. (1997): *Elementos de Matemática Discreta,* Sanz y Torres, Madrid.

Carrillo, J. (2001): *Otras Matemáticas para la clase de Secundaria. Resolución de problemas,* Consejería de Educación y Ciencia, Huelva.

Chamoso, J. y Rawson, W. (2003): *Matemáticas en una tarde de paseo,* Nivola, Madrid.

Corbalán, F. (1994): *Juegos matemáticos para Secundaria y Bachillerato,* Síntesis, Madrid.

<http://www.emanueleferonato.com/2006/10/29/flash-game-creation-tutorial-part-1/>

<http://ite.educacion.es/w3/recursos2/estudiantes/ocio/op_04.htm#02>

Mario Rafael Gil Martín
IES Guadiana
Ayamonte (Huelva)
<mariogil@telefonica.net>

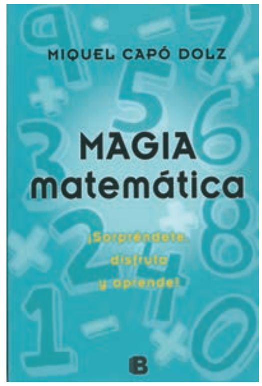
1 Instituto de Tecnologías Educativas.

2 fps en lenguaje técnico.

3 El número de Avogadro expresa el número de partículas que hay en un mol de cualquier sustancia y es igual a $6{,}022 \cdot 10^{23}$.

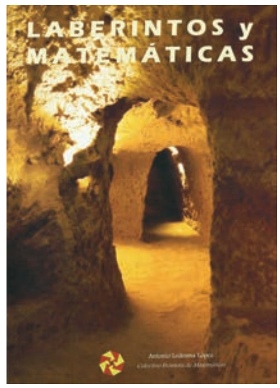
Publicaciones recibidas

Recreamáticas
Recreaciones matemáticas para jóvenes y adultos
Juan Diego Sánchez Torres
Editorial Rialp, SA
Madrid, 2012
168 páginas
ISBN 978-84-321-4178-2



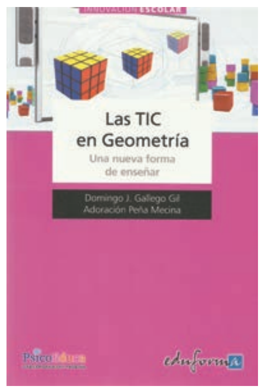
Mágia Matemática
Sorpréndete, disfruta y aprende
Miquel Capó Dolz
Ediciones B, S.A.
Barcelona, 2012
222 páginas
ISBN 978-84-666-5049-6

Publicaciones de la Facultad de Educación y Humanidades del Campus de Melilla
Universidad de Granada
Nº. 41, noviembre 2011
ISSN 1577-4147



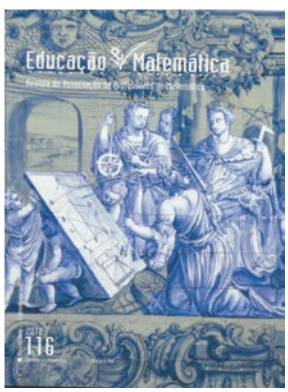
Laberintos y Matemáticas
Antonio Ledesma López
Colectivo Frontera de Matemáticas
Requena, 2012
ISSN 2254-2159

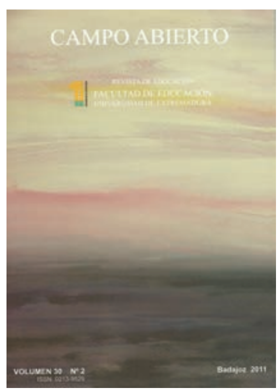
Tarea
Asociación de publicaciones educativas
Nº. 78, diciembre 2011
Lima, Perú
ISSN 0252-8819



Las Tic en Geometría
Una nueva forma de enseñar
Domingo J. Gallego Gil
Adoración Peña Mecina
Editorial MAD., S.L.
Alcalá de Guadaira
Sevilla, 2012
172 páginas
ISBN 978-84-676-5274-1

Educaçao e Matemática
Revista da Associaçao de Professores de Matemática
nº. 116, Janeiro-Fevereiro, 2012
Lisboa
ISSN 0871-7222





Campo Abierto
Facultad de Educación
Universidad de Extremadura
vol. 30, nº. 2, 2011
Badajoz
ISSN 0213-9529