Transforming activities from workbooks for preschoolers into computer games

Monika Tomcsányiová, *tomcsanyiova@fmph.uniba.sk*
Dept of Informatics Education, Comenius University Bratislava, Slovakia

Abstract

Children develop rather quickly. Admittedly, their mathematical, logical and mental skills can be influenced by books, which they are exposed to. There are some interesting workbooks for young children, which train their observation, give them first ideas about counting, colours, shapes, similarities and differences. A programmer can easily see a potential to convert parts of the workbooks into computer activities. The computer can enhance the activity by animations and sounds or by the ability to move the pictures around using the mouse or the keyboard. Computer games could become parts of such workbooks. Some activities can be made both in the workbook and in the computer. For other activities it is better to do them only in the workbook in order not to forget also about skills to paint using a real pencil or a crayon or to write and draw lines.

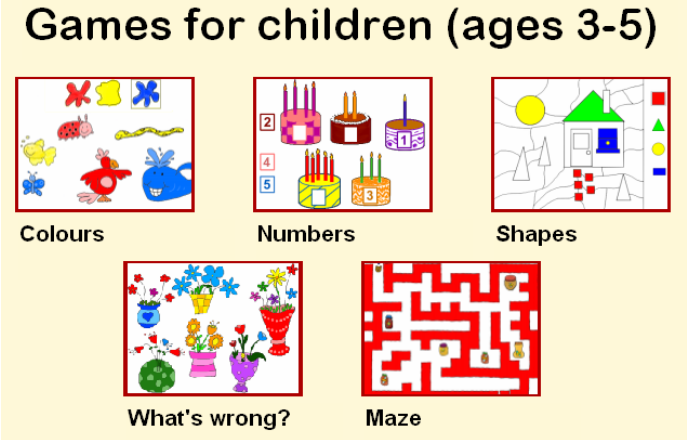

Figure 1. Choose the game

In our article we will describe some interesting programs for 3 to 5 years old children, which we made by transforming them from workbook activities.

From the very beginning our 4 years old son Thomas helped us. It seems that he understood how the computer can enhance the activities. He has seen how static activities from the workbook started suddenly moving and he was able to design colours of objects, give his voice to them or modify them. He was tirelessly colouring the pictures, he learned how to select them and move them to their proper place. Surprisingly, later when we worked with other activities in paper workbooks he was thinking how to transform them into computer activities, he even came with suggestions how the computer could check whether the child solved the assignments correctly.

Keywords

math, kids, computer games for kids, critical thinking

Motivation

3 to 5 years old children are given by the kind of curiosity and learning ability, which older children lack. We must certainly support these attributes by various types of activities. The most traditional way is to use workbooks, which allow the children developing their skills of using pencils and crayons. Such workbooks are often made by reputable authors and their applicability is tested by psychologists. The pictures are drawn by children's illustrators, so these workbooks are often not only suitable for children but also very appealing to them.

Figure 2. Workbooks for children
taken from http://matthewsmediallc.com/items/books-4-kids/list.htm?1=1

We have several such workbooks at home. Our son especially likes the Gold Starts series of workbooks because of its colourful and beautiful pictures. It has also an original idea of sticking stars to pages, which were well done by the child.

Nowadays many households own computers. The child sees how the parents are using the computer for their work and/or leisure. So it is very natural to seek for computerised activities for children, too. There are tons of activities, which their authors claim to be suitable for preschoolers. Can we believe them? We can compare them to our favourite workbooks. Or we can try to create computer activities, which resemble the activities in our favourite workbooks.

We tried to select such activities, where the computer can help making them more interesting and appealing. In the same time we wanted to make a set of activities covering several important skills and knowledge of preschoolers - distinguishing colours and shapes, counting up to 5, spotting differences and similarities, recognising the otherness. From the computer skills we focused especially to improving the coordination of eyes and hands when working with the keyboard.

Preparing the pictures

My son Thomas is 4 years old. He is already able to control the mouse, drag pictures around the screen, using the arrows on the keyboard. He learned all this by using the computer previously. He has been interested about the computer from his age of 3, so he already knew the graphical editor. He used it for creating his own very abstract paintings and drawings or he coloured pictures that we downloaded from the Internet.

To implement our new activities we needed at first to draw some pictures. We understood that the good activities attract children also because they are colourful and their pictures of animals, people and everyday things are realistic and childish at the same time. I drew them in the graphical editor RNA (http://www.logo.com/rna). Our son often stood beside me and pushed me

to finish drawing. He often indicated that the recently drawn line is not correct. So I had to use the undo ability of RNA quite often to make the pictures look well.

Figure 3. Thomas at work

Thomas understood quickly that he was not able to draw the pictures precisely enough. But he could colour them. So he was working with unbelievable patience, he blended colours until he made the right one. He coloured alone nearly all flowers in the *What's wrong*? and *Colours* activities.

We started with pictures for one activity. Then I transferred them into Imagine Logo and added the program so that the activity started to work according to my and Thomas's taste.

Programming

To implement the activities we chose Imagine Logo (http://www.logo.com/imagine). Not only because we know it very well, but also because it offers programming means, which are especially suitable when creating these kinds of activities.

Colours

The most well-known activities form children's workbooks are probably those for teaching the child to recognise colours. Children see colours every day, but it is not so easy to name them and to tell one colour from another one. The task of the child in this activity is to select a colour blot and then click to a picture of the same colour.

After preparing the pictures of blots and animals I was able to start programming. In the first phase, when defining the classes of objects used in the game, there is no visible result for a quite long time. My son started to be impatient and urged me. He wanted to see at least something working...

I created two classes: one of them is **Blot** and it represents the blots and the other, named **myPict** represents the coloured pictures. The **onLeftDown** event of the class **Blot** changes the frame of all blots to first one (which shows the blot not selected) and then changes the frame of the actually clicked blot to the second one (selected):

```
ask allOf "Blot [setFrame 1] setFrame 2
```

We must also create a variable named **frameMode** with value **true**. Otherwise it would not be possible to set the frames manually.

The **onLeftDown** event of **myPict** class must check whether the user clicked to a picture of the correct colour. In order to make testing simple we name the yellow blot **myYellow** and all yellow pictures will have names, which are different only by adding another character **myYellow1**, **myYellow2** etc.

When we added all yellow pictures, we found out that we must think out what should happen if the child clicks to a correct picture and what should happen if the click is incorrect. My son suggested that the computer will say "Yes" if the click is correct and "No" otherwise. He suggested the English words because they have some elementary English education in the kindergarten and he has already seen games, which react like this. Although in Imagine Logo it would be possible to make the speech synthesiser say the words by a computer's voice, it is more fun if Thomas can record his own voice for this.

Recording voice is quite easy in Imagine Logo. Just push the Multimedia button on the toolbar and in its drop-down menu select Wave sound. Then in the popping-up window just push its Create new button. You will get the standard Windows Sound Recorder application and you can start recording. After several trials (some trials were too loud or too silent or some noise was recorded along with the voice) Thomas was able to record the words Yes and No into files **Yes.wav** and **No.wav**.

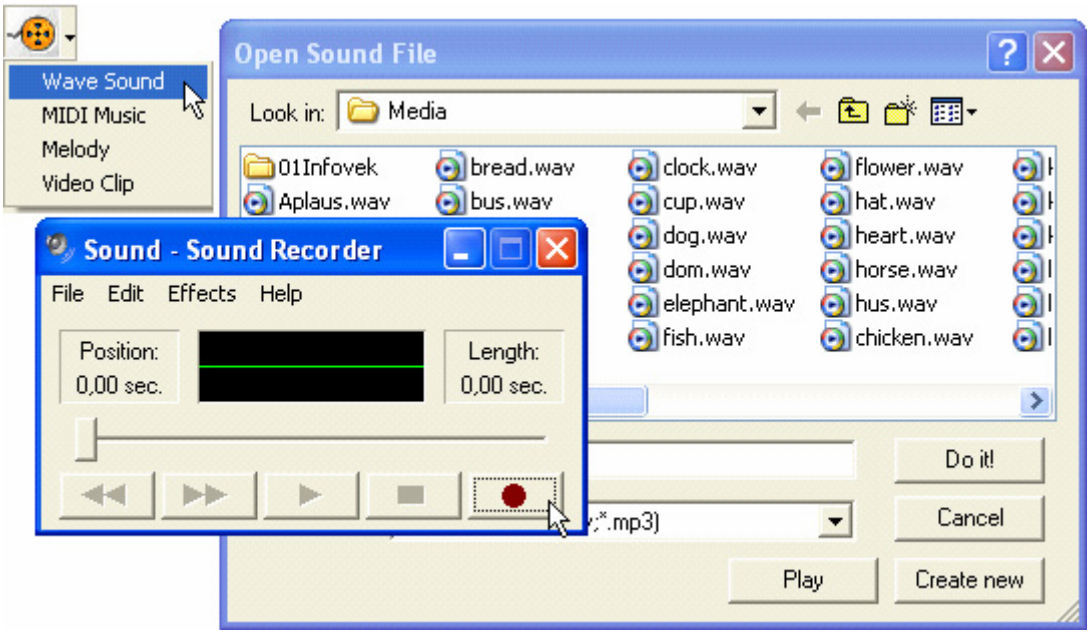

Figure 4. Record your voice

So the final version of **onLeftDown** event of the class **myPict** is:

```
ifElse (ask bl myName [frame]=2) [playW "yes] [playW "no]
ask bl myName [setFrame 1]
```

Numbers

A four years old child should be able to count up to 5 as well as know the numerals 1 to 5. We chose an activity where the task is to assign a numeral to a birthday cake, which has the same number of candles.

After creating the pictures, I started again with two classes. This time I named them **Cake** and **myNumber**. **Cake** is rather simple – it needs only define one variable **pen** with value **pu**.

The main piece of code lies in the **onLeftUp** event of **myNumber** class. Its effect is that the numeral being dragged either touches-down to the cake (if it is the correct numeral) or jumps off the cake to its home position:

```
ifElse (last last overlapped) = (last myName)
 [setPos ask overlapped [pos]] [home]
```

Note that testing is again based on the fact that the name of the cake and the name of the corresponding numeral are "similar". In this case their last character is equal (it is the numeral itself): The turtle representing the digit 1 is named **t1** and the cake with one candle is named **cake1**. The last two things to set for the class are two variables. One is **autoDrag** having value **true** and another is **pen** having value **pu**.

When the activity starts we want to rearrange the cakes so that the child could not remember each cake by its position on the screen and force it to take care even when doing the activity for several times in a row.

```
to start
 repeat 10 [
 let "mT1 pick allOf "cake
 let "mT2 pick allOf "cake
 let "p ask :mT1 [pos]
 ask :mT1 [setPos ask :mT2 [pos]]
 ask :mT2 [setPos :p]
 ]
end
```

Shapes

The task is to spot, identify and colour basic geometrical shapes (squares, circles, triangles and rectangles), which a scene consists of. Each type of shape should be coloured to the colour that is prescribed for them by one such object lying outside the scene at its border.

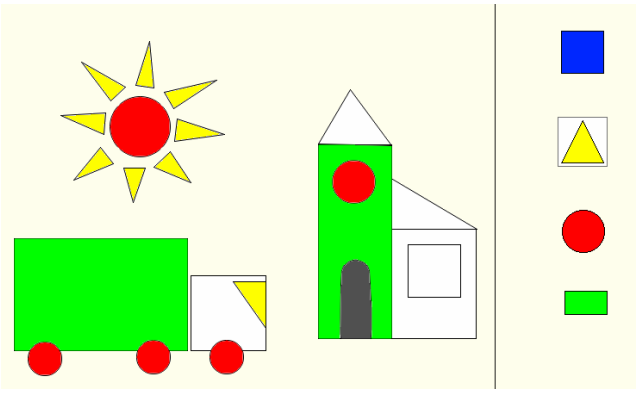

Figure 5. Can you spot another triangle?

This activity is in its structure nearly identical with the activity *Colours*. The objects on the border of the picture should behave in the same way as blots do in *Colours*. It means that they must

select themselves when clicked and deselect all other objects of the same kind. The picture consists of objects of the other kind. When they are clicked they must play a sound, then check if they were clicked correctly (i.e. they are of the same shape as the clicked prototype. If so then change their shape to a coloured one.

What's wrong?

It is a well-known puzzle to find a picture which does not belong to a group with other pictures. It can be as simple as an activity for preschoolers or as complex as an adult IQ test.

When creating the activity for such young children, the most important point is to find good pictures. They must show things, which are well-known to children and they can understand after a while which one does not fit into the group without lengthy explanations. Our picture shows several vases with flowers. Each vase contains one flower, which does not belong to the others in the same vase.

This is a kind of one purpose activity because it is not possible to mix the positions of flowers before the activity starts - the size and shape of vases and size and shape of flowers' leaves does not allow it. So we have only a background picture and one class of turtles named `Wrong`. Turtles of this class (shaped as flowers) are placed to all the places where the background picture contains a flower not belonging to that vase. The class `Wrong` has several variables: `shape`, `frameMode` and `transparentClick` (both set to `true`). It has also an event `onLeftDown`, which only changes the frame of the turtle to indicate that you spot correctly the wrong flower.

So the last point to finish the program is to place objects of class `Wrong` to the places where they belong to.

This project can be easily adopted for other kinds of objects. You just need to replace the background picture and to place instances of the class `Wrong` to correct places.

Maze

All the activities up to now contain one fixed assignment (even if we can mix things a bit as you have seen in the *Numbers* activity). The end-user cannot create its own versions of the assignment without some knowledge of Imagine Logo and an ability to draw new pictures.

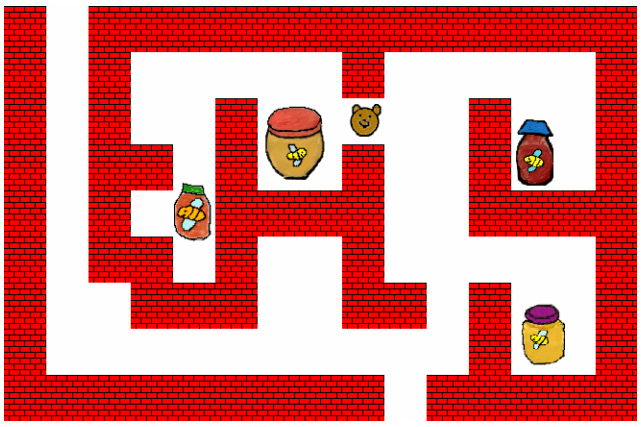

Figure 6. The Maze microworld

This activity is more flexible. It allows not only solving the task (collect all items in the maze and find the exit) but it also allows the child to create its own assignments (a maze with objects to collect positioned into it).

On the first page of this project there is a prepared maze. The player can move a bear inside the maze by using the arrow keys on the keyboard. The task is to collect all bottles with honey and then exit the maze.

When our son played the game he suggested that the computer should say something when the bear finds a bottle of honey. He remembered that recording had been very easy so he started to search for the correct button...

Later Thomas came with another suggestion: It should be not possible to exit the maze until the bear collected all the bottles. He suggested that the exit should be closed by a door and open only after collecting the last bottle. Another improvement was to forbid the bear to leave the maze by the same door by which he entered the maze.

The second page of this project contains a simple microworld, which allows the child to build a maze around the bear.

Clicking to an empty place you build a wall there. Clicking on a piece of wall it disappears. This way you can build a simple two dimensional maze. At the top of the page there are the bottles with honey. They can be dragged to some places in the maze.

The page has an event `onLeftDown`. It creates a new instance of the class `Wall`. It is important to position the walls to positions divisible by the sizes of the wall's picture:

```
new "Wall
 [pos ([22 24] + list
  44 * ( div first mousePos 44 )
  48 * ( div last mousePos 48 ))]
 if ask lastName [overlapped] <> [] [eraseObject lastName]
```

Also the class has its event. It just deletes the piece of wall which has been clicked on:

```
eraseObject myName
```

The bottles of honey are simple turtles having corresponding shapes, their pen is up and their `autoDrag` feature is switched on.

Conclusion

When creating individual programs we tried to implement activities from colourful workbooks for preschoolers of age 3 to 5 in the simplest possible way. The *Maze* microworld has been created as a specifically computer activity which should help the child to orientate himself on the two-dimensional surface of a computer screen.

We have also shown that Imagine Logo is a highly suitable tool, which allows to implement several activities with identical or just slightly different code. It enables a parent (who can program in Imagine Logo in some extent) to create several interesting activities for his/her child with very little effort. Each activity can be changed by changing only the pictures.

The child took part in the process of designing the programs. And it is very important to mention that the child was able to work differently with the program than with the same kind of activity in the workbook. He was able to prepare and colour the pictures, record sounds, suggest effects indicating good or wrong solution of the tasks.

The computer allowed us to make the activities more vivid by adding movements, animations and sounds. Imagine Logo has proved to be nice towards such kinds of activities as well as towards parents and children, who want to spend interesting and useful moments together next to the computer.

There are many similar activity books on the market. A parent, who is able to program, can find plenty of activities, which are destined to be re-implemented into the computer in the way, which we outlined in this paper. The computer can increase their attractiveness by including more variability, interactivity, animation and sounds.

References

Patilla, P. (2004) *Hviezda v matematike pre deti od 3 do 4 rokov*, Slovart, Bratislava, ISBN 80-7145-864-3

Patilla, P. (2002) *Gold Stars Math Ages 3-4*, Paragon, Bath, ISBN 0752594664

Tomcsányiová, M. (1999) *Logo programs for our 3 years old daughter Janka*, Eurologo99: Proceedings of the 7th European Logo Conference Eurologo'99, Sofia 1999, pp. 350-355, ISBN 954-9582-03-5.

Kalas, I. and Blaho, A. (2001) *Object metaphor helps create simple Logo projects*, Eurologo2001: Proceeding of the 8th European Conference. Wien Osterreichische Computer Gesellschaft, 2001, pp. 55-65, ISBN 3-85403-156-4

Van den Daele, Ch. (2002) *Čáry-máry žlté čiary*, Editions Caramel, Bratislava 2002, ISBN 80-85327-96-1

Macková, S. (2003) *Tak i tak*, časopis Silentium, Bratislava 2003

Kauka, R. (2005) *Macko Pusik*, časopis pre deti, Magnet-press Slovakia, s.r.o., Bratislava

Stoppard, M. (1992) *Test your Child*, Dorling Kindresley Limited, London 1991, ISBN 80-85186-46-2

Nováková, M. (1999) *Škôlkárova zima*, Nomi s.r.o. Košice 1999, ISBN 80-88958-17-2

Gibson, R. (1997) *I can draw animals*, Usborne Publishing Ltd. London, ISBN 80-89083-12-9

Tomcsányiová, M. (2001) *Niekoľko aktivít v prostredí Imagine*, Didinfo 2001, pp. 171-175, ISBN 80-8041-02364-8

Tomcsányiová, M. (2001) *Imagine – nová hračka pre rodičov s deťmi*, Poškole 2001, Praha 2001, pp. 171-175, ISBN 80-01-02364-8