Red Shell Challenge Cards

Teachers Notes:

This set of Challenge cards is designed to be used with the Blockly Turtle resources. The full collection of resources for teachers is found in the **Turtle** area on bebras.uk.

This set of cards is for pupils who have achieved their Red Shell Programmer award and are now working towards their Black award. These cards introduce lists. These are the final set of cards in the series. Lists are not normally taught in UK junior schools.

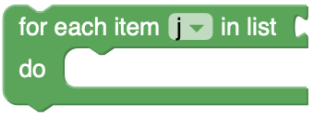
Preparation:

1. When the pupils login to their computers they should head to the ***Turtle Playground - Red***. They should be directed to:
   bebras.uk -> Turtle -> click on the Red turtle.
2. These cards should be printed out (size to:100% on A4 card, or “fill the paper” on A5 card) and laminated. Each pupil also needs their own Yellow Shell Record Card (which should not be laminated as they have to be written on). When a pupil completes a Challenge Card, its number can be written in their Record Card (in one of the clip boards).
3. In the first lesson, the teacher should show the students how to access ***Turtle Playground - Red*** and the *Lists - basics* video. At a later date students should watch *Lists and Loops*. Note ***Card 0*** is for the teacher to use with the class. Pupils can start with ***Card 1***.
4. Students should complete a minimum of 10 cards, from 14 available, so some choice is possible.

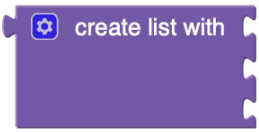
Red Shell Challenge Cards

Code Blocks introduced in **Turtle Playground - Red**:

There are two new blocks in the ***Loops*** folder in the toolbox:



A new ***Lists*** folder containing these blocks:

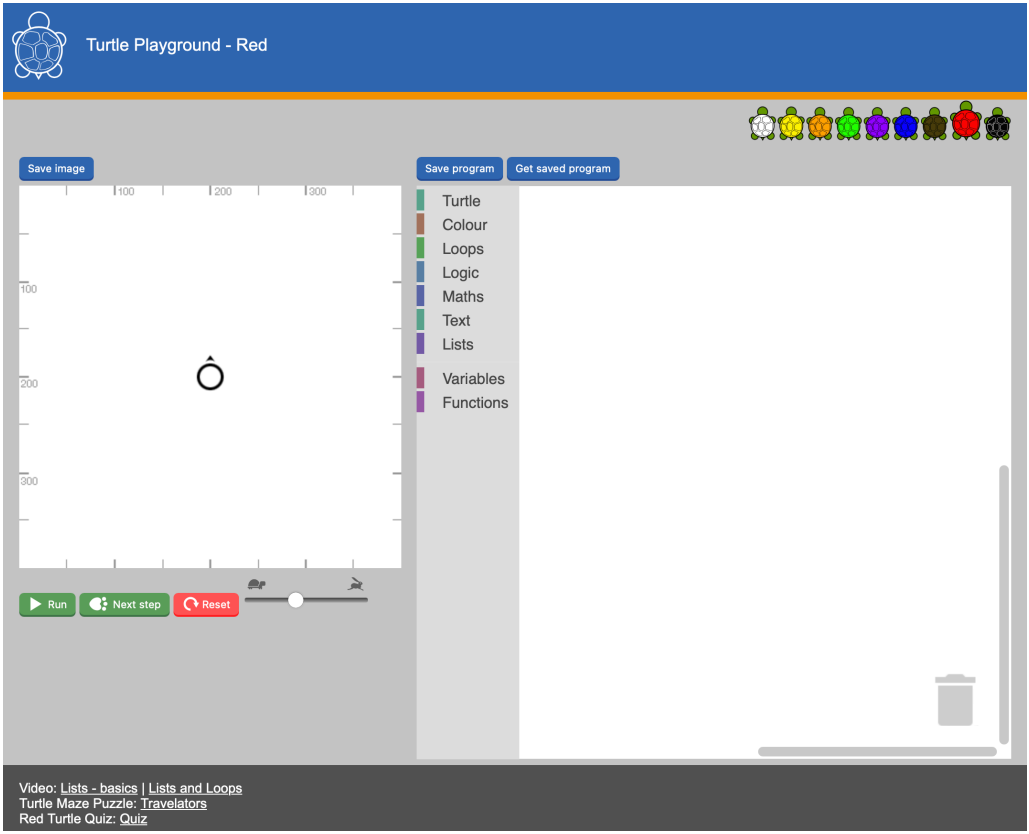


This Card is for teachers!

Red Shell Challenge Cards

0

1. Show your pupils how to go to ***Turtle Playground - Red***



2. Show your class the *Lists - basics* video.
3. Provide each pupil who has achieved their Red Shell award with their new Record Card.
4. Distribute these Challenge Cards.
5. Show pupils the *Lists and Loops* video when about to start Red Card 3.

Red Shell Challenge Cards

1

A colours list

Lists are containers that hold **items** of **data** that are similar to **variables**. Each item in the list has an **index** starting from 0:

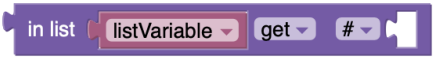
colours:	Red	Yellow	Green	Purple	Orange	Black
index:	0	1	2	3	4	5

So, Green is #2 in the *colours* list (said: “green is index 2 in the colours list”)

Challenge:

1. Make a list called “colours”. Add the above colours using these **code blocks**:

2. Edit your program so that it can **print** out “Green” using this block:



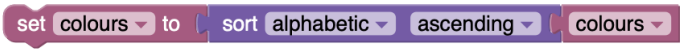
3. Print out all the items in the list *colours* in one go. (You will only need two blocks.)
4. Print out all the items in the list *colours* in alphabetical order.
5. Print out the index for “Purple" using:

Things to think about:

Do these blocks change the order of the the items in the list?

A)

B)



Red Shell Challenge Cards

2

Counting

Lists are containers that hold **items** of **data** that are similar to **variables**. Each item in the list has an **index** starting from 0:

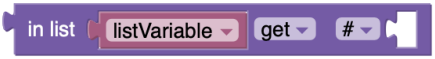
numbers:	1	2	3	4	5	6
index:	0	1	2	3	4	5

So, 2 is #1 in the *numbers* list (said: “2 is index 1 in the numbers list”)

Challenge:

1. Make a list called “numbers”. Add the above numbers with these **code blocks**:

2. Edit your program so that it can **print** out 2 using this block:



3. Print out all the items in the *numbers* list in one go. (You will only need two blocks.)
4. Print out all the items in the *numbers* list in reverse order.
5. Print out the index for 5 using:

(Your program should **return** 4.)

Things to think about:

Do these blocks change the order of the the items in the *numbers* list?

A)

B)

Red Shell Challenge Cards

3

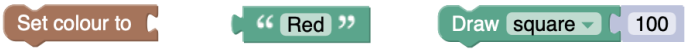
Words to colours

Lists are containers that hold **items** of **data** that are similar to **variables**. Each item in the list has an **index** starting from 0:

colours:	Red	Yellow	Green	Purple	Orange	Black
index:	0	1	2	3	4	5

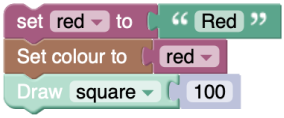
Challenge:

1. Try and make a program that **prints** out a red square with these **code blocks**:



   The *set colour* block doesn't accept text blocks!

2. Now make a variable called "red" and make this program:



   Does this work?

3. Write a program that **loops** through the *colours* list and draws a square for each colour, pausing one second between each square.

Extra:

Find out which colour words Blockly understands and one that it doesn't.

Roll the dice

Challenge:

1. Make a **list** containing the numbers 1, 2, 3, 4, 5 and 6.
2. Add **code blocks** to your program so that it selects and **outputs** a random number from your list.
3. Your program should use these blocks:

If you cannot find the correct block, look for it in the lists folder. It looks like this:

Of course we didn't need to put the numbers in a list to make this program. Having made it this way should help you to make the next two programs.

4. Make a list containing these **strings**: "rock", "paper", "scissors". Make a program that uses this list to select and **print** out one of them randomly. Now you can play this classic game against the computer!
5. When **run**, the *Fortune Teller* toy displays a sentence describing someone's future. Put 8 phrases into a list so that you can make a *Fortune Teller* toy.

   Example phrases: "It is going to be a great day today." "I would advise against getting out of bed this week." "I foresee great wealth in your life." "I'm sorry but I cannot see into your future at the moment."

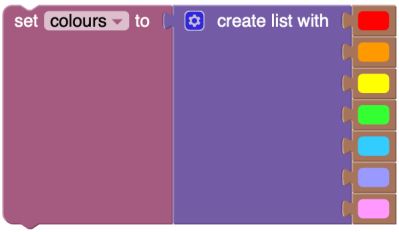
Red Shell Challenge Cards

5

Rainbow

Challenge:

1. Make this **list** with 7 colours of your choice.



2. Write a program to draw a rainbow that uses your list.

Hints:

- Draw circles on top of each other to make the rainbow.
- Draw a square that is the same colour as the background to hide the unwanted parts of the circles.
- Use the *for each item i in list* **code block** to select the colours

A list of names

A teacher wants a list of the children in their new class. She would like a program that asks the pupils to enter their names one a time and then prints out a list of their names.

Challenge:

1. Make an empty **list** called “names”.
2. Write a program that asks a **user** for their name and inserts it into the list *names*. This should happen repeatedly until one user (probably the teacher) enter: “end”.
3. Finally, add some **code blocks** that **print** out the list of names in *names*.

Hints:

- You might need to ask for a user’s name before starting the **loop** as well as inside the loop.
- You will need to use a **while loop** that loops while the entered name does not equal “end”
- You may find you have to remove “end” from the *names* list.

Extra:

Edit your program so that it prints out all the names and tells the teacher how many pupils are in the list, and who is first and last alphabetically.

Coded messages

Challenge:

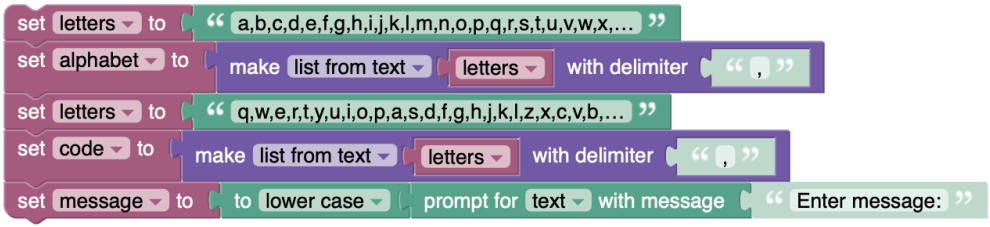
1. Make these two **lists**:
   alphabet: `a b c d e f g h I j k l m n o p q r s t u v w x y z`
   code: `Q W E R T Y U I O P A S D F G H J K L Z X C V B N M`

   Example: The message `turtle red shell` becomes `ZXKZST KTR LITSS`

   Note: It is traditional for messages to be in lowercase and for code to be in uppercase.

2. Write a program that encodes a message, supplied by a **user**, by finding the **index** in the *alphabet* list and then using the same index in the *code* list to build, and then **print**, the encoded message. Spaces should be ignored.

   Here is the start of a program you can use:



Extra:

1. Edit your program so that it decodes an encoded message.
2. Amend your program so that it firsts asks the user whether to encode or to decode. It should then either encode a provided message or decode a coded message.

Shift cypher

A shift cypher is a secret code that hides a message by substituting letters like this.
Shift = 2: (The code letters are all shifted 2 places to the right.)

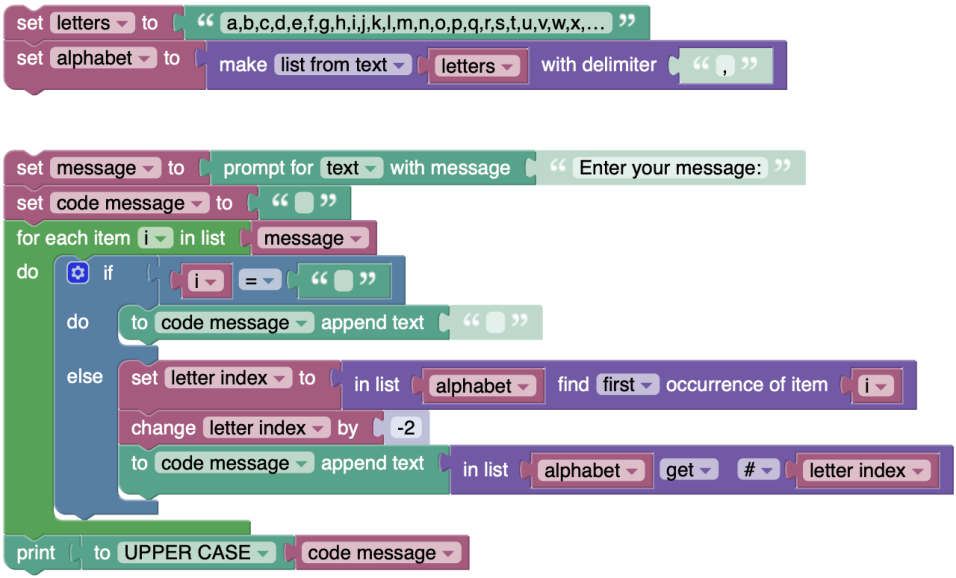
message letter: `a b c d e f g h I j k l m n o p q r s t u v w x y z`

code letter: `Y Z A B C D E F G H I J K L M N O P Q R S T U V W X`

Example: The message `turtle red shell` becomes `RSPRJC PCB QFCJJ`

Challenge:

1. A text **string** can be **iterated** through, one letter at a time, as if it is a **list**!
   Write this program that asks a **user** for a message and then **outputs** the coded message:



2. This program doesn't work with messages containing a or b. Can you fix it?
3. Make a version of this program that decodes a shift cypher with Shift = 5.

Red Shell Challenge Cards

9

Library index

Challenge:

1. Make these four **lists** with information about books in a library:

title:	`The Gruffalo, 1984, Thirteen, The Grinch, Emma`
author:	`Donaldson, Orwell, Hoyle, Seuss, Austen`
age:	`children, adult, young adult, children, adult`
rating:	`5, 4, 4, 4, 5`

Note how all the information for each book has the same **index**.
Example:
index 0 has the title: The Gruffalo, author: Donaldson, age: children, and rating: 5.

2. Write a program where a **user** can **input** a book title and the program then **outputs** all the information about that book.
3. Make sure your program lets the user know if they enter a book that isn't in the library.
4. Edit your program so that the information about the book is nicely **displayed** in the Turtle drawing area.

Save your program to use with Red Cards 10 and 11

Hint:

Returns the index of the first/last occurrence of the item in the list. Returns -1 if item is not found.

Adding books

If you have completed Red Card 9, and saved your program, you can import it so you don't need to do step 1 below.

Challenge:

1. Make these four **lists** of books in a library:

title:	`The Gruffalo, 1984, Thirteen, The Grinch, Emma`
author:	`Donaldson, Orwell, Hoyle, Seuss, Austen`
age:	`children, adult, young adult, children, adult`
rating:	`5, 4, 4, 4, 5`

   Note how all the information for each book has the same **index**.
   Example:
   index 0 has the title: The Gruffalo, author: Donaldson, age: children, and rating: 5.

2. Write a program where a **user** can be asked for the details for a new book. The program should then add the new details to the end of the four lists.
3. Add some code that checks this process was successful by **printing** out the information for this new book from the lists:

   "You just added <title>, <author>, <age>, <rating> to the library."

Book selection

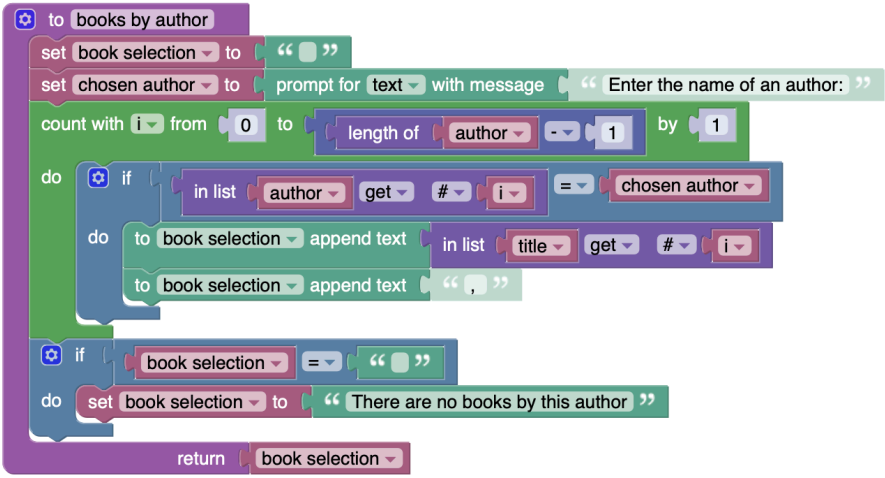
If you have completed Red Card 9, and saved your program, you can import it so you don't need to do step 1 below.

Challenge:

1. Make these four **lists** of books in a library:

title:	`The Gruffalo, 1984, Thirteen, The Grinch, Emma`
author:	`Donaldson, Orwell, Hoyle, Seuss, Austen`
age:	`children, adult, young adult, children, adult`
rating:	`5, 4, 4, 4, 5`

2. Add this **function** to your program:



3. Now make two more functions: "books by age" and "books by rating".
4. Write a program that asks the **user** which of these facilities they would like to use and then calls the appropriate function and **prints** out the results.

Red Shell Challenge Cards

12

Pick a card

There are 52 cards in a pack of standard playing cards. We could make a single **list** but it is probably better to make two:

suits: `clubs, diamonds, hearts, spades`

ranks: `Ace, King, Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3, 2`

Challenge:

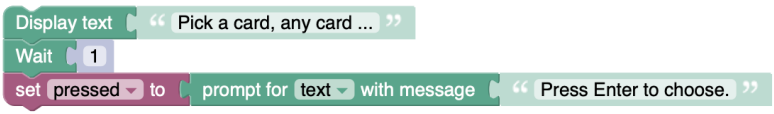
1. Make a program that randomly selects a playing card and then **displays** the **output** on the Turtle **drawing area**.

2. Now make a program that asks the **user** to pick a card and then displays it nicely with the message: “You have chosen: <card>”.

Save your program to use with Red Card 13

Hint:

You can ask the user to pick a card with these code blocks:



Red Shell Challenge Cards

13

Higher or lower?

There are 52 cards in a pack of standard playing cards. We could make a single **list** but it is probably better to make two:

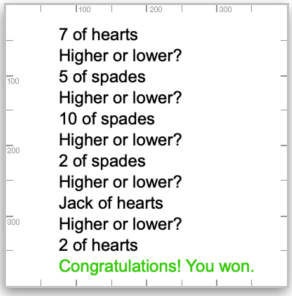
suits: `clubs, diamonds, hearts, spades`
ranks: `Ace, King, Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3, 2`

If you have completed Red Card 12 you can get your saved program and skip step 1.

Challenge:

1. Make a program that randomly selects a playing card and then **displays** the **output** on the Turtle **drawing area**.

2. The **user** should then be asked to guess whether the next card is “higher or lower?”
3. The next card is then displayed.
   - If they are correct, the question is repeated.
   - If they are wrong display the message: “Never mind, better luck next time!”
   - If it is their 5th correct guess display the message: “Congratulations! You won.”



Hints:

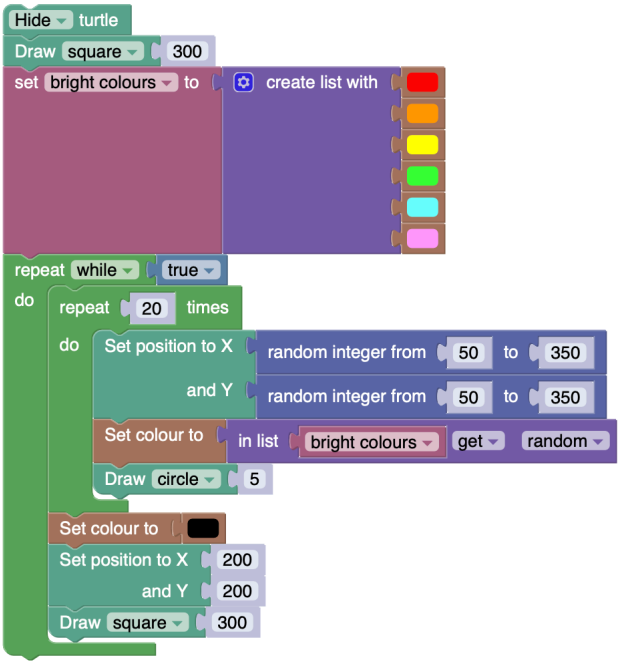
• Ignore the fact that the same card might be drawn twice!
Separate smaller tasks into functions:
• Make a “next line” function.
• Make a “draw card” function that returns the rank.
• Make a “get guess” function that returns the user’s guess.
• Make a “correct?” function that returns *true* or *false.*

Red Shell Challenge Cards

14

Decorations

Here is a program that draws twinkling circles. The twinkling never stops, thanks to the *while true* **loop**.



Challenge:

1. Make a copy of this and study how it works.
2. Now make a dark green Christmas tree out of triangles.
3. Put these **code blocks** into a *draw tree* **function**.
4. Give it a pot to sit in.
5. Add some lightbulbs in some good positions. Add this code to another function called *draw bulbs*.
6. Make the bulbs flash on the tree using these two functions.
7. You will probably want to add a star on top.

Hint:
Save your work regularly

Extra:

- Try out different ways of making the bulbs flash.
- Draw a house with lights decorating it and windows that occasionally light up. You could even have a night sky with twinkling stars!