MYnd Kraft

GUIDE TO PROGRAMMING CONCEPTS INTRODUCED IN THE GAME

Spoken languages share some basic concepts such as verbs, nouns, adjectives, adverbs etc. Similarly, programming languages also share some basic concepts and the Action Cards introduce these concepts in a fun and exciting way.

The various programming concepts introduced in the game are listed below and explained in more detail in the sections that follow.

VARIABLE

This is a storage location that contains information referred to as a value. The value of a variable could be changed.

ASSIGNMENT

This is the action of setting a value to a variable.

BOOLEAN LOGIC

This involves evaluating whether a statement is TRUE or FALSE.

BOOLEAN OPERATORS

Boolean operators are used to manipulate TRUE/FALSE values (i.e. they facilitate testing the truth or falsity of combinations of Boolean Logic values). The game introduces the AND and OR operators.

CONDITIONAL STATEMENT

A conditional statement is a set of rules where an action is performed if the specified rules are satisfied. The game introduces the IF/ELSE statement.

SWITCH-CASE STATEMENT

A switch-case statement is a chain of IF/ELSE statements where multiple conditions are tested and only one or none is expected to match.

LOOP

A loop is a sequence of instructions that are repeated either a fixed number of times or while a certain condition is satisfied. It allows programmers to perform the same action multiple times without having to do extra work. The game introduces the FOR-LOOP and the WHILE-LOOP.

DATA STRUCTURES

Data structures are a specialized format for organizing and storing data. The game introduces data structure concepts such as the ARRAY and DICTIONARY.

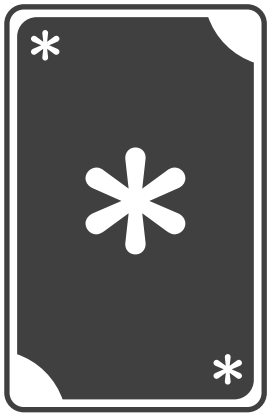
INSTRUCTIONS

Instructions are simple orders/commands given by a computer program to a computer processor.

FUNCTIONS

Functions are specific named sections of a computer program that perform specific tasks (i.e. they are procedures or routines that perform specific tasks). Simply put, a function is made up of multiple instructions. Once a function is defined, it helps reuse that part of the code instead of having to rewrite it.

Point to Note: This document simply highlights the various programming concepts introduced in the game. It does not explain how the game is played nor how the various Action Cards are solved using Number Cards. This is explained in detail in the Instructions Manual.



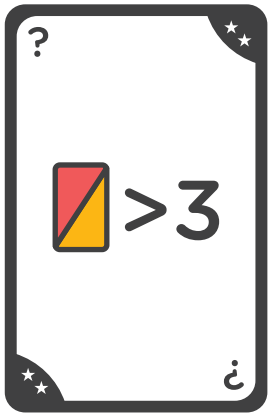
ASTERISK NUMBER CARD

Definition:

Asterisks Cards are Number Cards that can stand in to be any value ranging from 0-7 in any colour.

Programming Concepts Introduced:

Asterisk Cards are similar to **Variables** in programming as they can stand in to be any value ranging from 0-7 in any colour set by the player. So, when a player solves an Action Card by playing an Asterisk Card instead of a Number Card, the player is doing so by assigning a value and colour to the Asterisk Card. Assigning a value and colour in this manner to the Asterisk Card is an example of the concept of **Assignment** in programming.



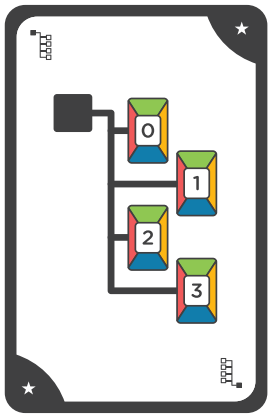
CONDITION ACTION CARDS

How to Solve:

Only one Number Card, which is red or yellow and greater than 3, could be played.

Programming Concepts Introduced:

- To determine whether the Number Card is of the correct colour, red/yellow (i.e. true) or blue/green (i.e. false), **'TRUE' or 'FALSE' Boolean Logic** is applied along with the **'OR' Boolean Operator** (i.e. red or yellow).
- Then, to determine if the Number Card is both red/yellow and greater than 3, the **'AND' Boolean Operator** is applied.
- The action of checking whether the condition specified on the Action Card is satisfied and playing the correct Number Card involves application of **Conditional Statements - IF** the Number Card is red or yellow and greater than 3 in value, **THEN** play the Number Card, **ELSE** do not play the Number Card.



SWITCH ACTION CARDS

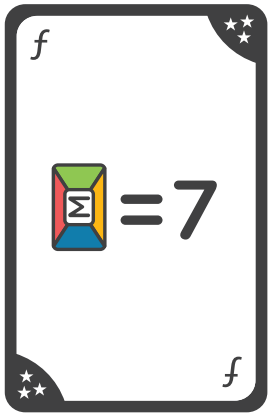
How to Solve:

Only one Number Card in one of the values specified (0, 1, 2, 3) irrespective of the colour could be played.

Programming Concepts Introduced:

Determining whether the Number Card is in one of the values specified - 0, 1, 2, 3 (i.e. true) or 4, 5, 6, 7 (i.e. false) - involves application of the **'TRUE' or 'FALSE' Boolean Logic** and this is tested in a chain of **Conditional Statements** as follows:

- **IF** value is 0, **THEN** play Number Card
- **ELSE** test **IF** value is 1, **THEN** play Number Card
- **ELSE** test **IF** value is 2, **THEN** play Number Card
- **ELSE** test **IF** value is 3, **THEN** play Number Card
- **ELSE** do not play Number card



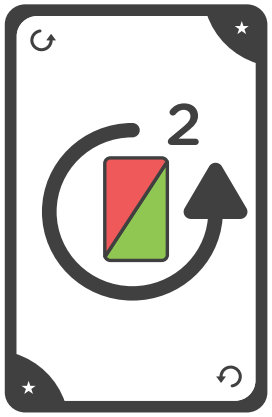
FUNCTION ACTION CARDS

How to Solve:

One or more Number Cards, irrespective of the colour as long as the values of the Number Cards add up to 7, could be played.

Programming Concepts Introduced:

The pre-defined task of identifying the correct combination of Number Cards that add up to 7 is an example of the application of the **function concept**. So, in this example, a function provides a convenient way of grouping together the multiple individual instructions involved in identifying the correct combination of Number Cards that add up to 7.



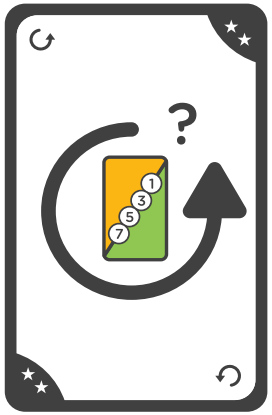
FOR-LOOP ACTION CARDS

How to Solve:

Only two Number Cards in red or green irrespective of the value could be played.

Programming Concepts Introduced:

- Determining whether the Number Card is of the correct colour, red/green (i.e. true) or yellow/blue (i.e. false), involves application of the **'TRUE' or 'FALSE' Boolean Logic** and also, the **'OR' Boolean Operator** (i.e. red or green).
- The action of checking whether the condition specified on the Action Card is satisfied and playing the correct Number Card involves application of **Conditional Statements** - **IF** the Number Card is red or green, **THEN** play the Number Card, **ELSE** do not play the Number Card.
- Playing the correct Number Cards a **fixed number of times** (i.e. twice in this example), demonstrates the **For-Loop** programming concept where a set of instructions is repeated a pre-determined number of times.



WHILE-LOOP ACTION CARDS

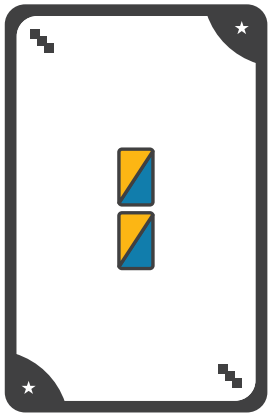
How to Solve:

Only green or yellow Number Cards in one of the values specified (1, 3, 5, 7) could be played. What's key to note is that more than one Number Card in each value and colour specified could be played.

Programming Concepts Introduced:

- Determining whether the Number Card is of the correct colour, yellow/green (i.e. true) or red/blue (i.e. false), involves application of the **'TRUE' or 'FALSE' Boolean Logic** and also, the **'OR' Boolean Operator** (i.e. yellow or green).
- Assessing if the Number Card is both yellow/green and in one of the values specified, requires application of the **'AND' Boolean operator**.
- The action of checking whether the **Conditional Statement** (i.e. the Number Card is yellow/ green and in one of the values specified) is satisfied and playing all Number Cards that satisfy this condition is an application of the **While-Loop** concept.

Point to Note: With the For-Loop, the action is only repeated a fixed number of times but with the While-Loop, the action is repeated as long as the condition is satisfied (i.e. it is not limited to a fixed number of iterations).



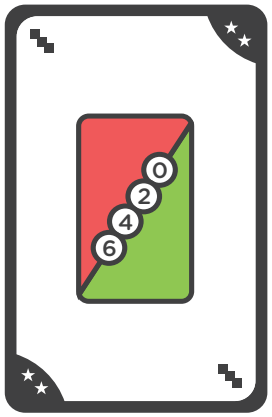
ARRAY ACTION CARDS

How to Solve:

Up to two Number Cards in either blue or yellow, irrespective of the value could be played.

Programming Concepts Introduced:

- Determining whether the Number Card is of the correct colour, yellow/blue (i.e. true) or red/green (i.e. false) involves application of the **'TRUE' or 'FALSE' Boolean Logic** and also, the **'OR' Boolean Operator** (i.e. yellow or blue).
- The action of checking whether the condition specified on the Action Card is satisfied and playing the correct Number Card involves application of **Conditional Statements** - **IF** the Number Card is red or green, **THEN** play the Number Card, **ELSE** do not play the Number Card.
- An Array is a format for organising and storing elements (i.e. two Number Cards in this example) that have the same data type (blue or yellow). Playing up to two Number Cards from this pre-defined data type is an example of the **Array Data Structure** concept.



DICTIONARY ACTION CARDS

How to Solve:

Only red or green Number Cards in one of the values specified (0, 2, 4 or 6) could be played. Unlike with the While-Loop where more than one Number Card in each value and colour specified could be played with the Dictionary Data Structure, only one Number Card in each value and colour specified could be played.

Programming Concepts Introduced:

- Determining whether the Number Card is of the correct colour, red/green (i.e. true) or yellow/blue (i.e. false) involves application of the **'TRUE' or 'FALSE' Boolean Logic** and also, the **'OR' Boolean Operator** (i.e. red or green).
- The action of checking whether the Number Card is red/green and in one of the values specified and playing the correct Number Card involves application of the **Conditional Statement**.

A Dictionary is a format for organising and storing data where there are unique identifiers known as 'keys' and associated objects known as 'values'. These concepts are similar to a standard dictionary where 'keys' are the words one looks up for the meaning and 'values' are the descriptions provided for those words. In programming, it's important to note that each 'key' should only be capable of identifying one unique 'value'.

With the Dictionary Action Card, 'keys' are the colour-value pairings specified on the Action Card (i.e. 0-green, 0-red, 2-green etc.) and 'values' are the corresponding Number Cards (i.e. 0 Number Card in green, 0 Number Card in red, 2 Number Card in green etc.). As every 'key' should only have one unique 'value' associated with it, each colour-value pairing could only have one unique corresponding Number Card. This is the reason only one Number Card in each value and colour specified could be played with Dictionary Action Cards.

- Playing only one Number Card for each key (i.e. colour-value pairing) is an example of the **Dictionary Data Structure**.