STEAM activity: Music Maestro

Description of the activity:	Hands on activity on Light Graffiti, as an example on how the process of inquiry and the design process are integrated.
Target group(s):	Primary teachers
Keywords:	Underlying principles, models, template STEAM activity
Duration of activity:	90’
Description of activity environment and materials needed:	Reading through this powerpoint provides tools to carry out a real STEAM activity with your pupils in primary school. It also shows you the underlying principles and models on integrated STEAM education in order to reflect upon the activity.

Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

The Development of STEAM education is co-funded by the Erasmus+ Programme of the European Union (Grant Agreement No. 2020-1-LT01-KA201-077935). The content of this document is the sole responsibility of the organizer and does not represent the opinion of the European Commission (EC), nor is the EC responsible for any use that might be made of information contained.

Rhythm is a dancer

An example of a STEAM activity for primary education

Students explore tempo, rhythm and melody in a piece of music

Students make their own piece of music by programming tempo, rhythm and melody in scratch

Materials needed:

- template with rhythm codes
- Music installation (in order to play a song)
- Computers – scratch
- Makey MAkey (optional)
- Objects that can make sounds
- In order to construct their own music instrument, the children prepare a list of materials they need to make the instruments and use for example trash materials (optional)

Problems to be tackled:

Listen and distinguish sounds with different tempo and rhythm.
Defining tempo as a unit that indicates at which speed a composition needs to be played.
Distinguish and apply musical codes to define the rhythm of a piece of music.
Distinguish and apply musical notes to define the melody of a piece of music.
Set up, apply, check and adjust a simple algorithm to solve a specific task or achieve a goal such as programming a piece of music.
Applying fractions while analysing the rhythm patterns of songs.

www.steam-ct.org

Engage

- Did you like the music? Why? What exactly did they like? (melody, rhythm, tempo…)
- How did you feel about the music?
- What makes you move to the music?
- What’s the difference between *tempo* and *rhythm*

Tempo: 81 bpm

Tempo: 103 bpm

Tempo: 120 bpm

Investigate

To keep up with ***the tempo*** of a song: use a metronome

https://www.google.com/search?q=metronoom

Let's now focus ***on the rhythm***!
Rhythm is also in all of us…
Try to move AND clap on this song:

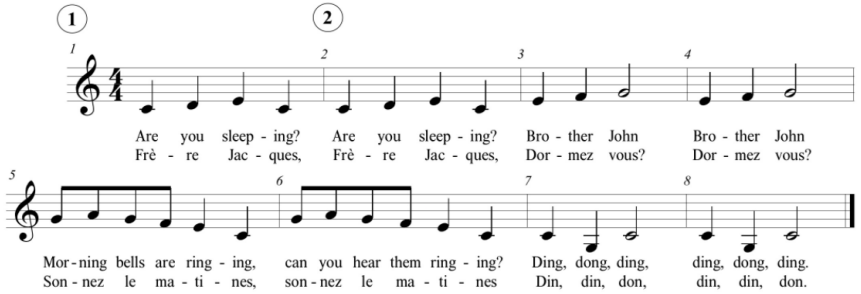
https://www.youtube.com/watch?v=WA4iX5D9Z64&t=2s

If you find yourself tapping or nodding your head along with music, it's probably the beat you are following and you are synchronized with.
As you clap along in time with the music, you're keeping track of the tempo.
It's instinctive: a skill we generally develop naturally in early childhood as we start listening to music, and dancing, clapping or singing along with it.

So Rhythm is not the same as tempo… But of course both terms are linked with each other.
For example musicians can play a song at a different tempo, but the rhythm of the song will stay the same.
Actually you can think of it as a kind of beat that supports the song.

Investigate

Investigate the rhythm of the song ‘Father John’



1 bar

1x klappen per 4 tellen = TU (= hele noot) = 1

Play 1 ‘clap’ per 4 beats = **TU (= full note = 1)**

1x klappen per 2 tellen = TO (= halve noot) = 1/2

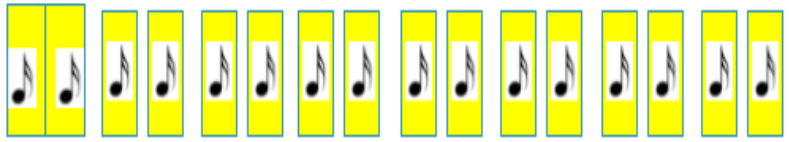
Play 1 ‘clap’ per 2 beats = **TO (= half note = 1/2)**

1x klappen per tel = TA (= kwartnoot) = 1/4

Play 1 ‘clap’ per beat = **TA** (= quater note = ¼)

2x klappen per tel = TI TI (2 achtste noten duren samen één tel) = 1/8

Play 2 ‘claps’ per beat = **TITI (= eighth note = ⅛)**



4x klappen per tel = TIRITIRI (4 zestiende noten duren samen één tel) = 1/16

Play 4 ‘claps’ per beat = **TIRITIRI (= sixteenth note = 1/16)**

Investigate

Some exercises: Try to clap or sing the Rhythm
(use the metronome to set a tempo: The faster the tempo, the more difficult it will be to follow the rhythm

Investigate

E.g. The teacher plays some short and repeated rhythms: e.g.

GREY-RED-RED = TO-TA-TA or

RED-GREEN-GREEN-GREY = TA-TITI-TITI-TO,

a really challenging one:
GREEN-GREEN-YELLOW-YELLOW-YELLOW-YELLOW-RED-GREEN-GREEN = TITI-TITI-TIRITIRI-TIRITIRI-TIRITIRI-TIRITIRI-TA-TITI-TITI.

The group in turns repeat the same rhythm by clapping and/or singing.
It’s helpful to keep using the metronome.

Investigate

Try to analyse the rhythm of this piece of music

The solution

bars 1 & 2:

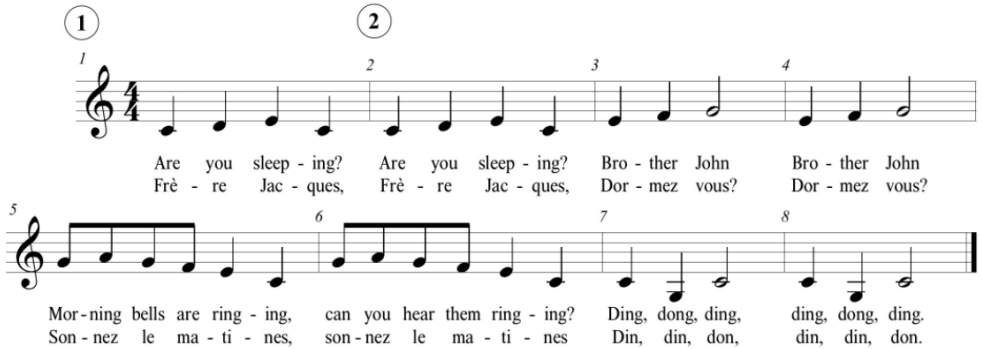
bars 3 & 4, 7 & 8:

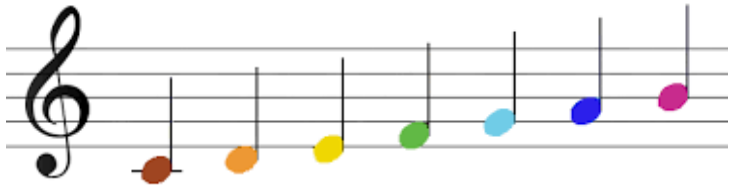
bars 5 & 6:

Investigate

….and what about the melody?

Can you recognize the notes of this song by comparing?





Do Re Mi Fa Sol La Si

in some countries, they also use :
A(la)-B(si)-C(do)-D(re)-E(mi)-F(fa)-G(sol)

Conclude

What makes a piece of music work?

Which elements are important to think of when creating a piece of music

- tempo:
- rhythm:
- Notes (melody)

How does each of them affects the music?

Create

Music Composer!

TASK (in small groups)
Let's compose/program our own song!

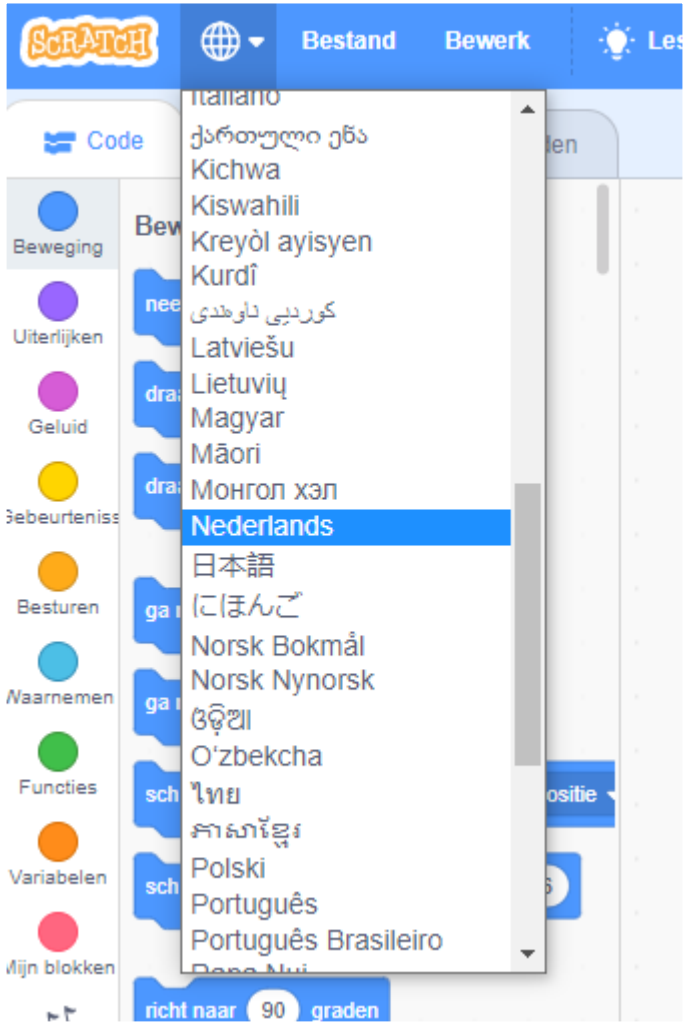
The song needs to last for 8 'bars'. Each bar takes 4 beats.
You can use all the notes do-re-mi-fa-sol-la-si.
Try to search for a catchy melody by using the notes and the rhythms we have learned!

In order to help you compose the song, you can use scratch!
http://scratch.mit.edu

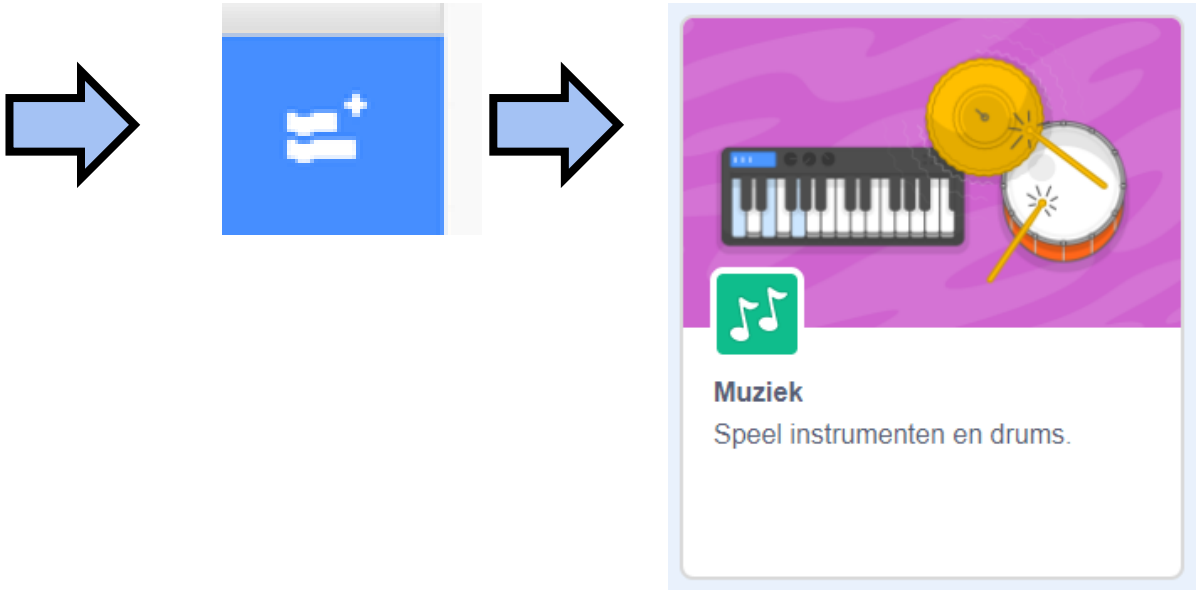
….BUT BEFORE
Let's explore SCRATCH a bit!

Investigate

Let’s start exploring ‘scratch’ together



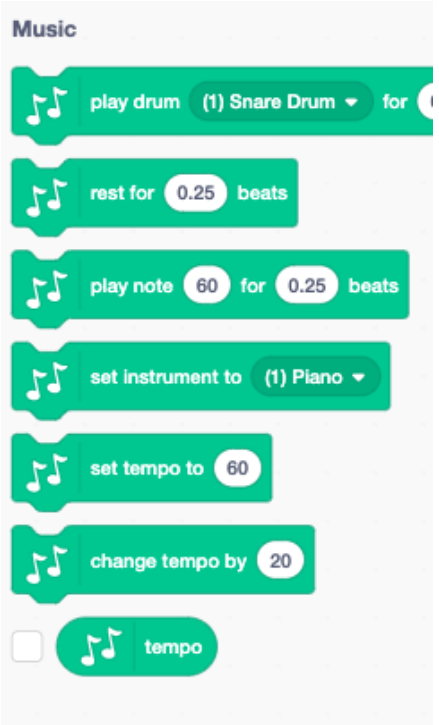
Add the ‘music’ extension tool

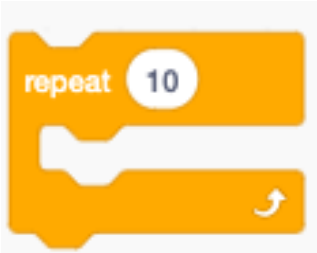


Similar work is that of Bell and Bell (2008),
Bell, Judith, and Tim Bell. "Integrating computational thinking with a music education context." Informatics in Education 17.2 (2018): 151-166.

Investigate

The commands which are important in this extension can be found here:

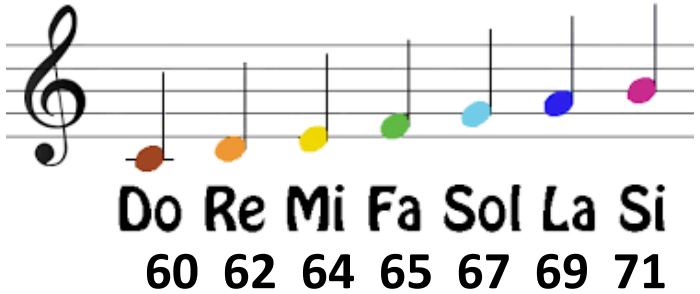




This table you will need to program the notes:

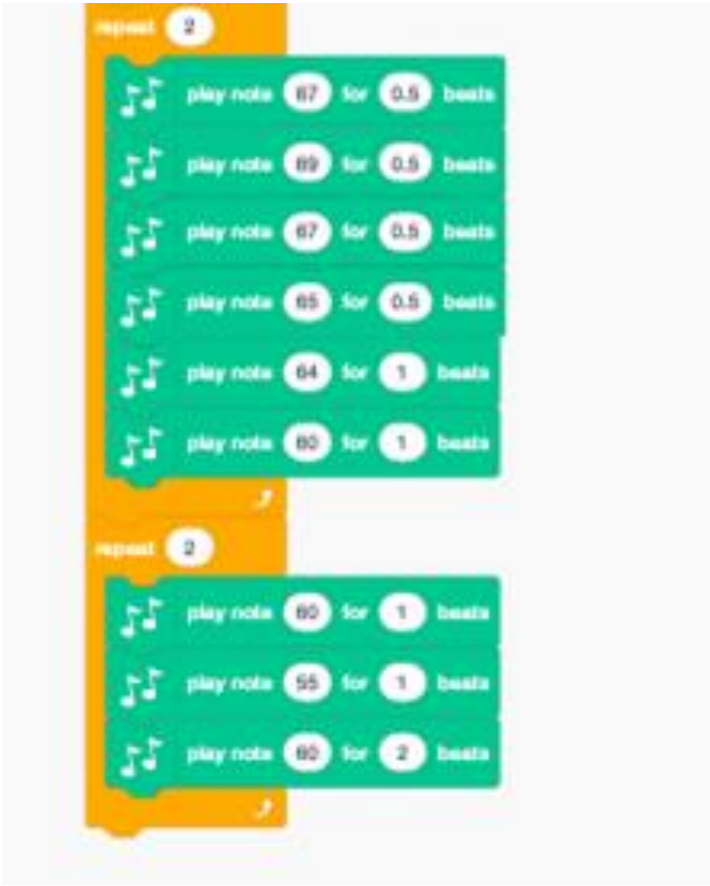
do = 60
re = 62
mi = 64
fa = 65
sol = 67
la = 69
si = 71

Octave	Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-2	0	1	2	3	4	5	6	7	8	9	10	11
-1	12	13	14	15	16	17	18	19	20	21	22	23
0	24	25	26	27	28	29	30	31	32	33	34	35
1	36	37	38	39	40	41	42	43	44	45	46	47
2	48	49	50	51	52	53	54	55	56	57	58	59
3	60	61	62	63	64	65	66	67	68	69	70	71
4	72	73	74	75	76	77	78	79	80	81	82	83
5	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107
7	108	109	110	111	112	113	114	115	116	117	118	119
8	120	121	122	123	124	125	126	127				



Investigate

Example of Brother John in Scratch (with tempo 120 bpm).
If the children are not familiar with scratch, you can use this example to explain the program and how it is used in order to compose and play songs.



Plan

Before starting to compose in scratch,
use the rhythm blocks to compose the rhythm of your song!

Create

Let’s compose/program our own song!
The song needs to last for 8 ‘bars’. Each bar takes 4 beats.
You can use all the notes do-re-mi-fa-sol-la-si.
Try to search for a catchy melody by using the notes and the rhythms we have learned!

Report

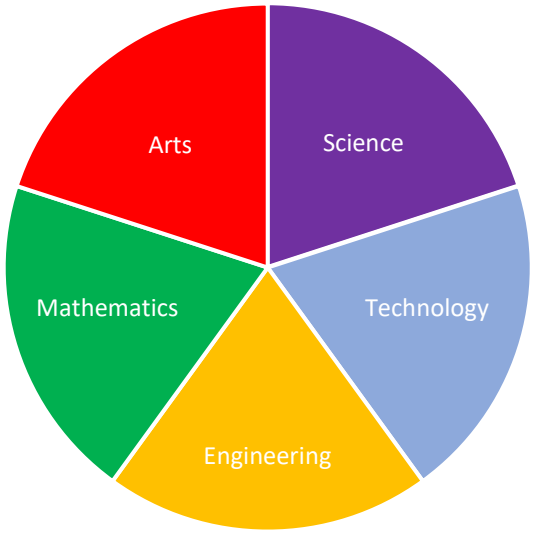
Let’s play it!

Play the song in scratch!
The other groups listen to the song and in group we can discuss and analyze the different songs.

Underlying principles, models

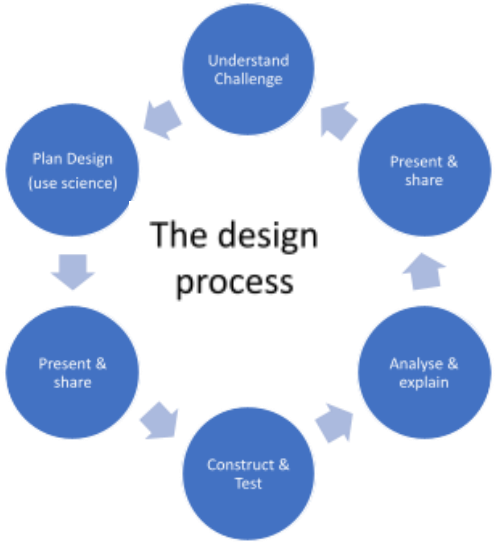
iSTEAM

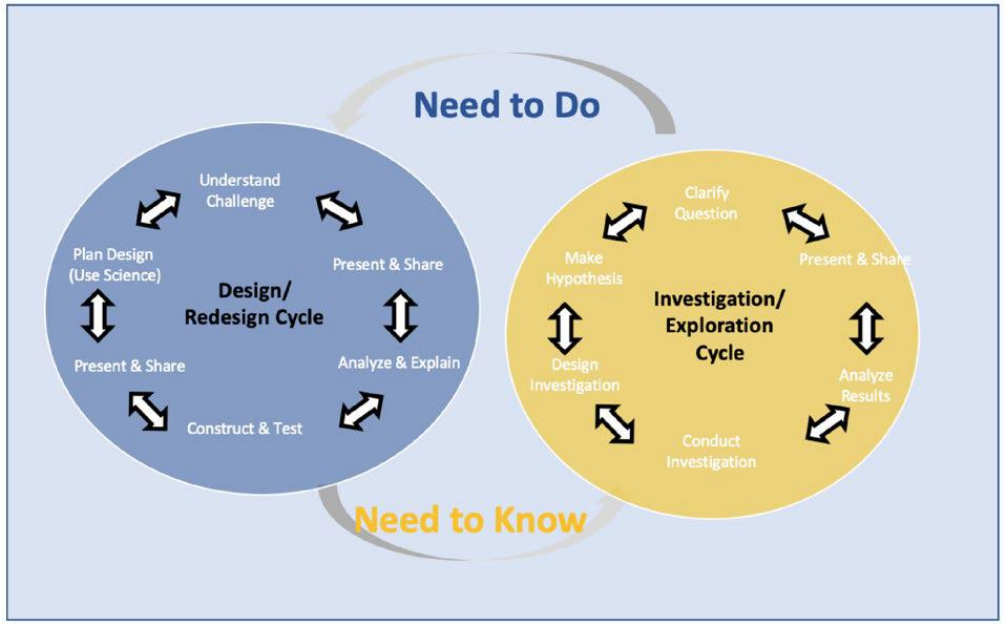
Integrated STEAM



Computational thinking

Data collection	
Data analysis	Abstraction
Data representation	Algorithms & procedures
Problem decomposition	Automation
Pattern recognition	Simulation
Debugging	Parallelization
	Generalization

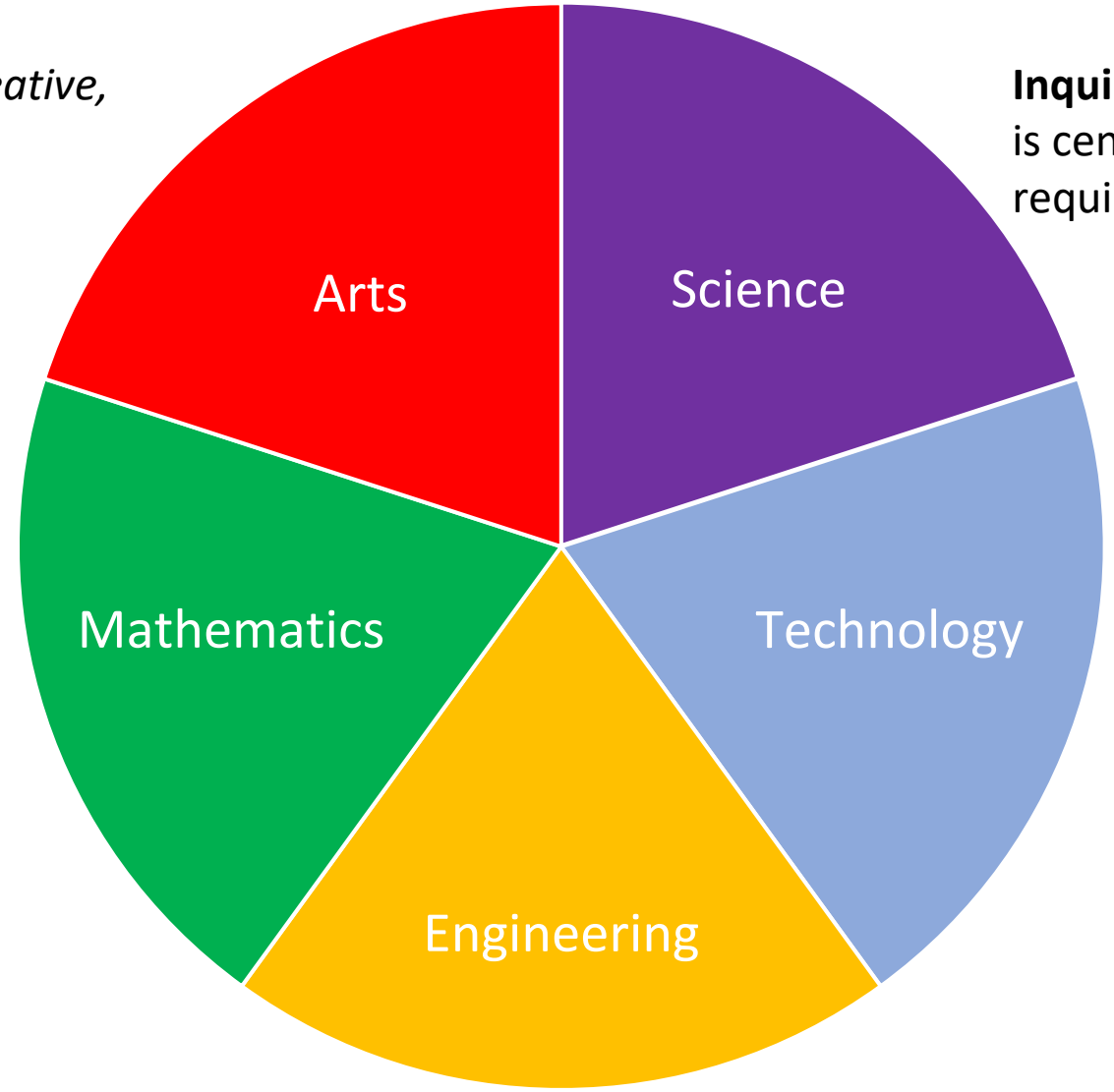




Underlying principles, models: iSTEAM

The **application** of *creative, critical thinking and analytical skills*

Inquiry of *scientific concepts* is central. Inquiry skills are required for this.

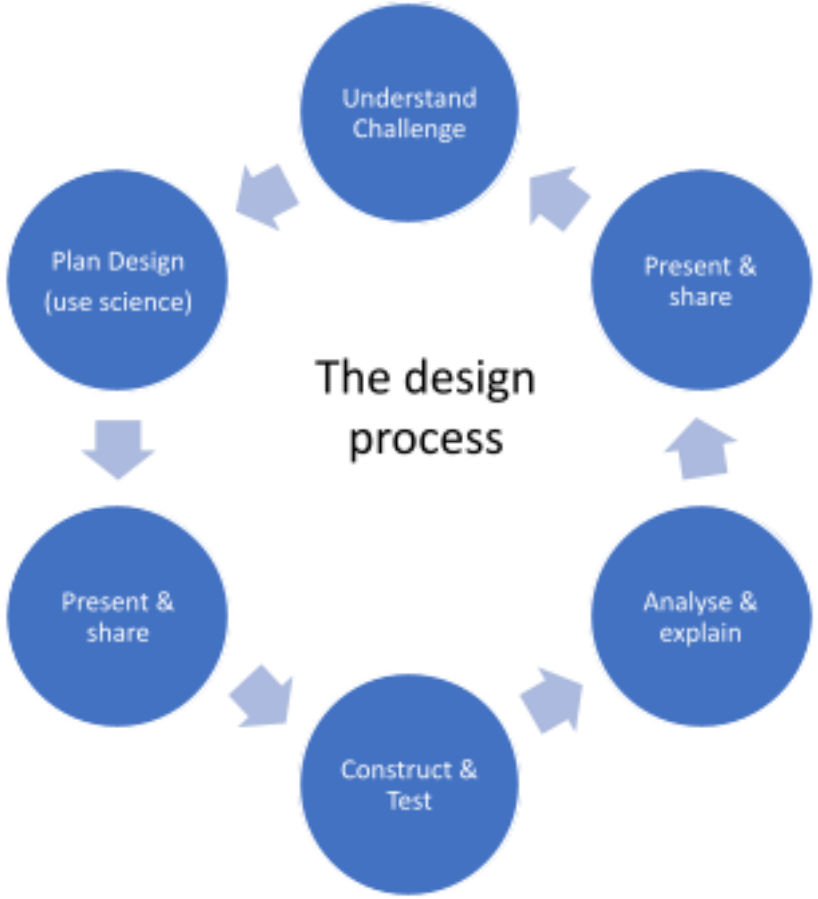


The **application** of *mathematical concepts* is central. Ex. relationships, analysis, volume, scale calculation, area calculation, Collecting and representing data, graphs,…

The **design** and **making** of *technical systems* is central.

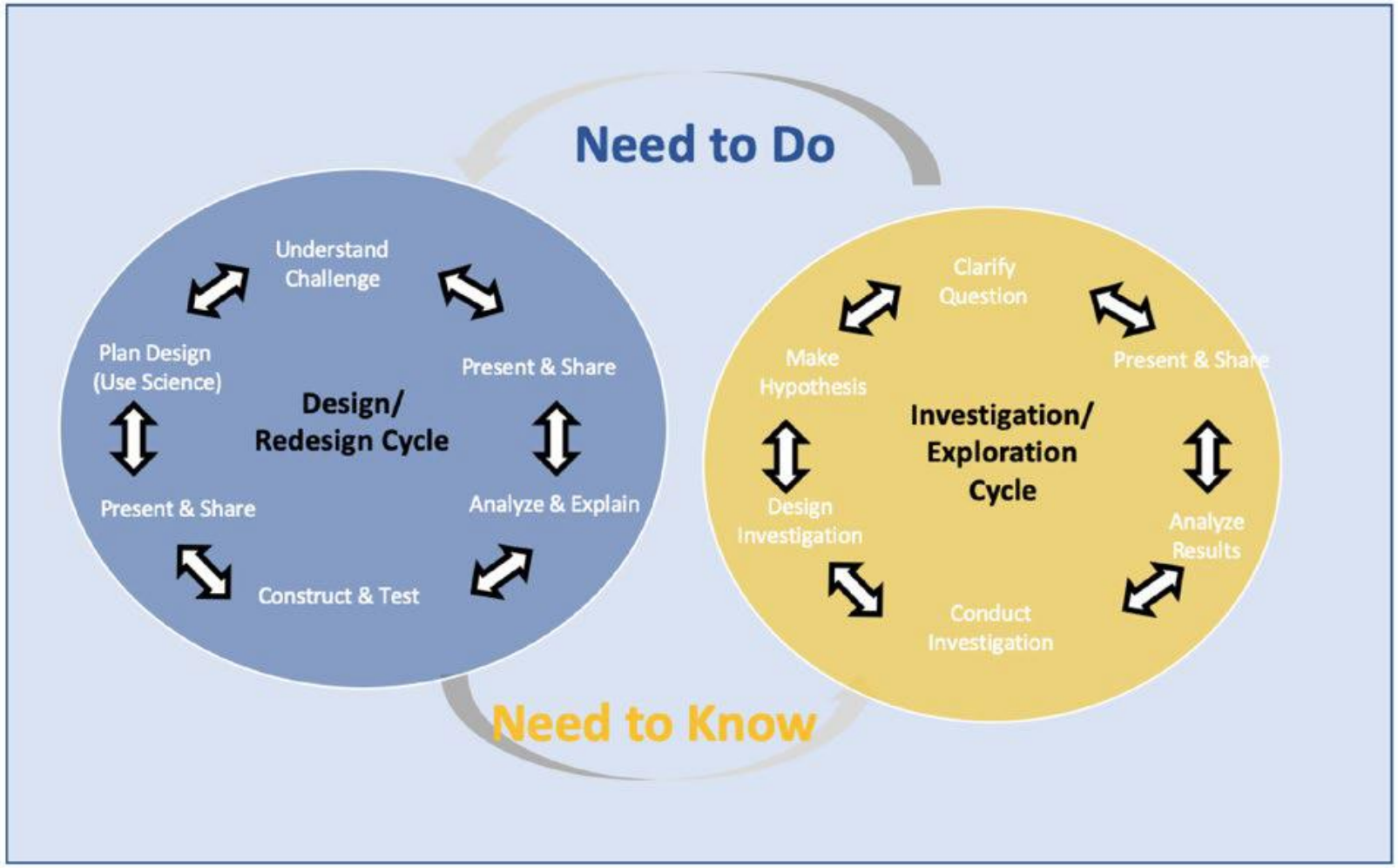
Optimizing design and process is central. Design and research go hand in hand

Underlying principles, models: iSTEAM



Van Graft & Kemmers, 2007; Kolodner et al., 2003;

Underlying principles, models: *iSTEAM*



Kolodner et al. (2003); Vossen (2019)

Underlying principles, models: *Computational thinking*

Data collection	Collect relevant data through experiments or investigation to solve the problem.
Data analysis	Analyse and understand data to find patterns and draw conclusions.
Data representation	Transform the analysed data to be readable or interpretative through graph, chart, writing or drawing.
Problem decomposition	Divide problems into manageable parts/factors.
Pattern recognition	Recognize certain patterns in a sequence of an algorithm or procedure.
Debugging	Find errors in algorithms & procedures and also to remove them.
Abstraction	Identify key elements for troubleshooting and simplify complexity by weighing those elements differently (reducing complexity).
Algorithms & procedures	Explore a problem and translate the problem into sequential steps so that a computer system can perform. Often also seen as “extraction”: considering common features of several examples and building a structure or category which has all of these features.
Automation	Execute repetitive tasks quickly in a form that a computing system can perform.
Simulation	Run a sequential program you created to solve the problem and get the results (automation).
Parallelization	Perform operations simultaneously with different parameters to achieve a goal.
Generalization	Apply the product/the solution in various types of context if it is working or not.

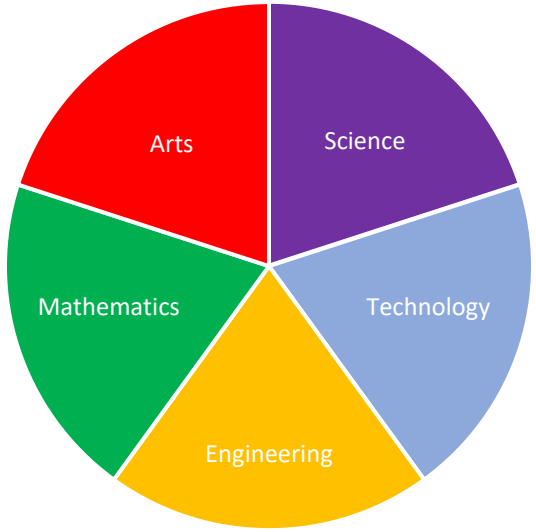
Reflection

First, discuss the following questions:

1. What have you learnt during the activity?
2. Did you use mathematics? When? Examples?
3. Did you recognise elements of computational thinking?
4. What did you do very well? Why?
5. What went wrong? Why?
6. What will you do differently next time?

Reflection

- Did you recognize elements of the different STEAM disciplines? Where did you use elements of ‘science’, ‘technology’, ‘engineering’, ‘mathematics’, ‘arts’?
  - Use the *iSTEAM model* to formulate your answer and use terms such as ‘inquiry’, ‘design’, ‘make’, ‘optimize’, ’apply’, ‘create’ …



Reflection

- Did you apply elements of the inquiry cycle and design cycle? Use the model below to explain