Lesson 18 – Dilemma of Learning through Sprites or Devices

In this lesson, we will evaluate Role of Devices
& see why we must also learn coding of Devices?

The Dilemma

Field of education, has forever had a dilemma whether learning through theory is better, or through Projects.

Coding has traditionally been the domain of graduation & post graduation.

With the advent of Scratch, it has not only come to the schools, but to the primary school.

Thus, a similar dilemma now exists on how to teach coding to children?

Should it be based on Sprites, or should it be based on devices, or should it be done using both. Let us examine.

Advantages of Coding through Sprites

Sprites are an amazing coding tool with following advantages:

- **Sprites enable young children to generate interest, & ultimately create a foundation for themselves in coding.**
- **Requires no technical skills.**
- **Stimulates a child's imagination & creativity the fun way.**
- **Children get to see the result of their creation immediately.**
- **Sharing with others is simple.**
- **The basic SW for their use is free, thus affordable by all.**

Shortcomings of Coding Through Sprites

Use of sprites comes with a few shortcomings. The key shortcomings are:

Shortcoming 1. Sprites are 2D images used to code. The result of coding is seen on a stage provided in the software.

In real world, Coding is used for machines, robots & sensors.
These do not exist on a stage, but in an environment outside the PC.

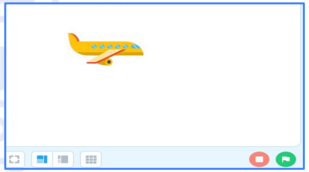
Sprites cannot perform in the real world outside.

Situation of sprites is synonymous to fishes in an aquarium. Cannot exist outside it.

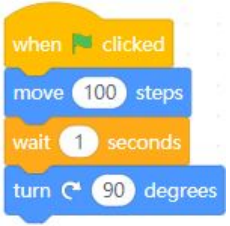
Shortcoming 2

Since sprites have only two dimensions, they fail when required to represent the third dimension.

Because of this, a child's mind, fails to get the right perspective.
To understand this, Select airplane 4.

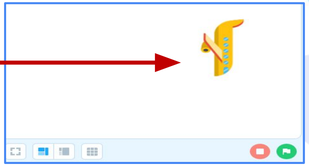


Now write this code, run & see the effect.



```
when 🏁 clicked
move 100 steps
wait 1 seconds
turn ↻ 90 degrees
```

Command turn 90 degree shows
The airplane falling downwards.

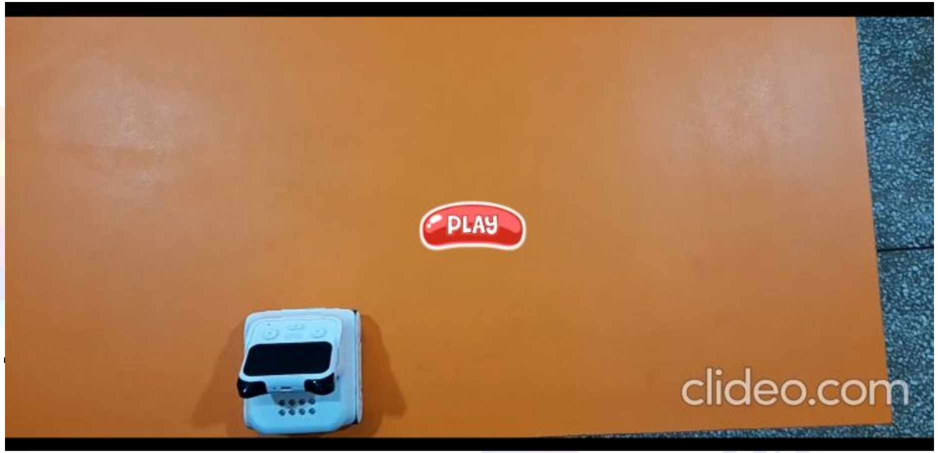


A child can get confused & take home a Wrong Perspective.

Now upload the same code on a device & see its video.

View is totally different from that of a sprite.

This is because in Real World Devices have 3 dimensions to which we all routinely relate.



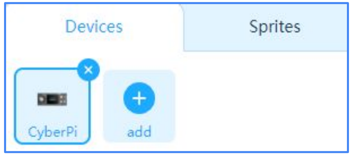
Shortcoming 3

In real world, a code must have the ability to sense the environment & take decisions. In the case of sprites this is done by sensing blocks.

Num of sensing blocks available is limited. This restricts teaching as well as learning.

Devices have no limitations on the use of sensors to sense the environment.

To get an idea, select devices.
See availability of blocks for default device Cyber Pi.



Also see its block extensions & the device library.
They offer a reservoir of blocks to learn from.

Advantages of Coding with Devices

Devices offer the best method to trigger a child’s imagination & his ability to design a solution.

Children see the results immediately & in a format that gives them the joy of creativity, & a sense of achievement.

These achievements add a feather in a child’s profile, build his confidence & open the gateway to competitive coding.

Develops a child’s skill sets & aptitude for entrepreneurship or make a child job ready by the time he leaves school.

Coding devices is as easy as coding sprites. Even a child of grade 3 can start & learn from.

Basic SW for learning Devices is also free.

Disadvantages of Coding with Devices

There is a cost involved in its purchase. However, keeping the amount a child can learn, it is insignificant.

Sharing the code with others requires them to have a similar device.

In the case of schools, these are not relevant.

Evaluation of Options

On evaluation the following deductions stand out:

- Sprites are a great tool to develop interest in coding.

- Once developed, children need to be initiated to devices for the vast & real life learning they provide.

- Sprites are found wanting when it comes to professional languages - Python & Arduino C. Devices enable all three.

- Having learnt Scratch, learning professional languages becomes easy.

Approach Adopted in our Teaching

Emphasis on Sprites for foundation laying.

Gradual introduction to the fascinating world of devices.

- Initially, go over the sprite codes using a device. Later learn to code directly.

- Recommended devices are:
  - ✔ Appropriate to child's learning level & requirements.
  - ✔ Have ability to migrate children to Python & Arduino C.
  - ✔ Offer multilevel learning using the same device.

Why this Option?

Reason 1. Coding is considered as The English of Tomorrow.

In another ten years
a child not knowing Coding will be at the same disadvantage in landing himself a decent job
as a child not knowing English has been over the last 100 years.

Do we want that to happen again?

Reason 2. Learning Coding is easier than learning English.

Devoid of all the grammar, learning Scratch Based Coding is easier than learning English, Hindi, Sanskrit, French, German or any other language.

Reason 3. In the near future, Coding will be a key Asset to land lucrative Jobs.

According to Bill Gates 'A revolution in machine learning has yet to come, & when it does, it will be worth ten Microsoft's'.

That is where these jobs will exist.

Takeaways…

- Sprites are a great Initiator of children to coding.
- Devices are the Finisher of choice.
- To be effective, learning:
  - Must use Both.
  - Be concurrent.
  - Be in three languages – Scratch, Python & Arduino C.
  - Children leaving school must be employable.

Time to Do

Go through the Device Library of mBlock 5, & before investing in a device, evaluate the learning potential for yourself.

End of Lesson 18

Code Karega India Badhega