



**Reviving hands-on educational play for
learning skills of tomorrow**
PROJECT N° 2019-1-UK01-KA201-061466

MÓDULO 2

Kits Eletronicos – Scratch 2.0

DESENVOLVIDO BY IDEC & PLATON

CIVIC

 **idec**


ΕΚΠΑΙΔΕΥΤΗΡΙΑ
ΠΛΑΤΩΝ

 **learn**
EUROPEAN DIGITAL LEARNING NETWORK


scholé

Emphasys
CENTRE

 **CCS**
Digital Education

DESCRIÇÃO DO MÓDULO

O Scratch é uma ferramenta de programação visual. Ao utilizar Scratch pode criar animações e jogos com uma interface de arrastar e largar. Torna fácil a criação de jogos de computador e histórias interativas sem ter de escrever código.

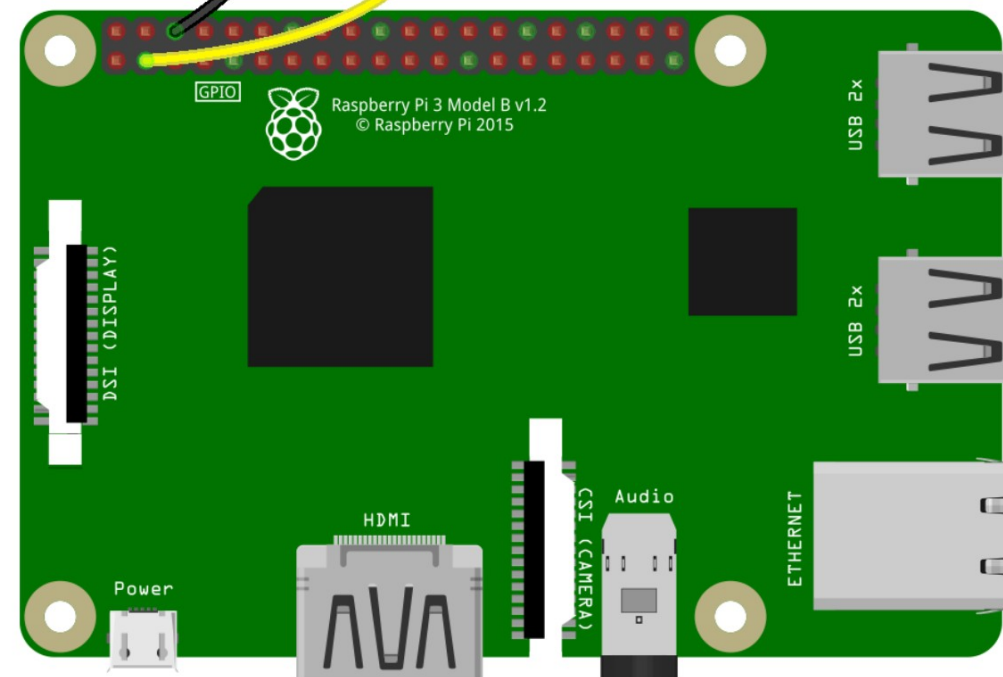
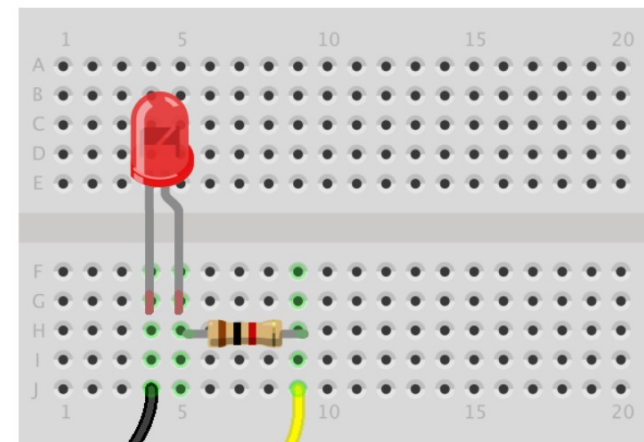
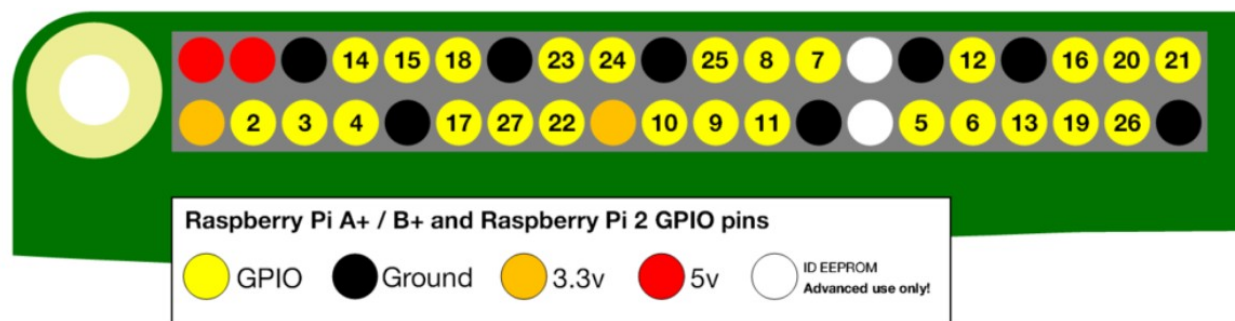
A versão do Scratch incluída no Raspberry Pi facilita a comunicação entre os pinos Raspberry e GPIO (General Purpose Input Output). Através destes pinos pode ligar o seu Raspberry Pi a sensores, luzes LED, botões e muito mais.

Vamos agora fazer uma pequena demonstração usando o Pi e alguns componentes. Nesta demonstração simples usaremos Scratch para programar o Pi para piscar um LED.

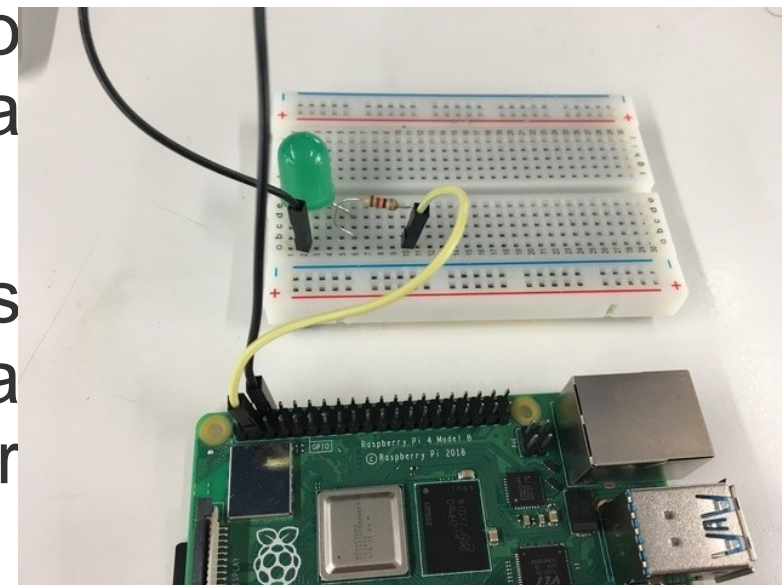
Primeiro, vamos ligar tudo.

Vais precisar de:

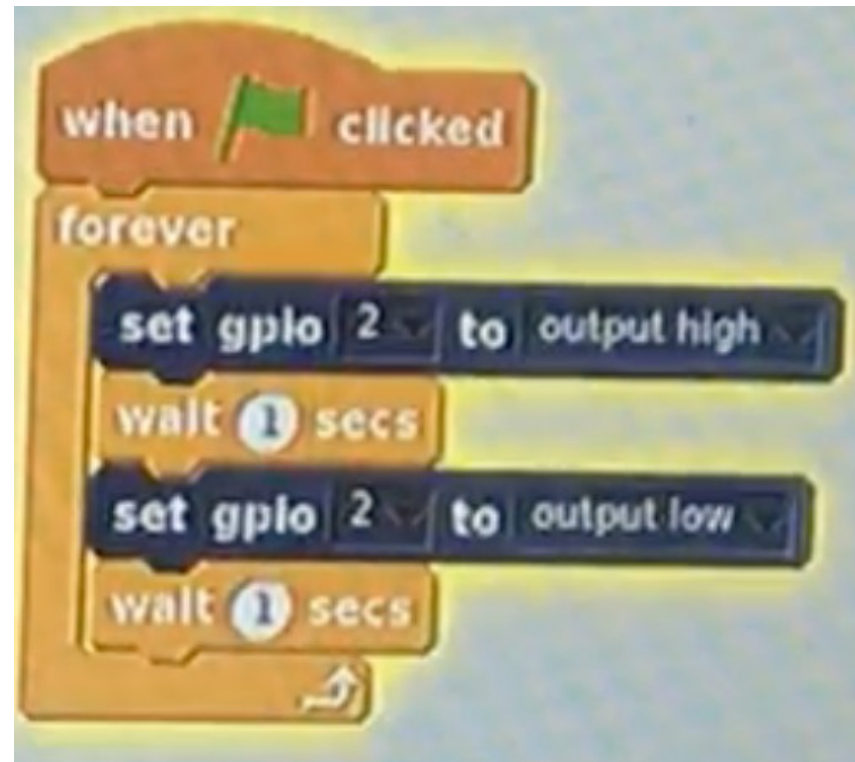
- um LED (de qualquer cor)
- uma resistência de 1k ou menos
- 2 cabos jumper fêmea - macho



- Liga um cabo jumper fêmea-macho do GPIO pino 2 do Pi a qualquer local da breadboard.
- Agora, liga a resistência (1k ou menos serve) do cabo jumper da GPIO2 a um outro local da breadboard
- Agora pega no LED e descobre a perna mais longa. Liga essa perna a uma ponta da resistência e a perna mais curta pode ligar em qualquer lugar da bread board.
- Por fim, liga o cabo jumper macho – fêmea do lado desconectado do LED ao pino terra (ground) do Pi.



Agora, vamos programar o Pi usando o Scratch, para que LED pisque.

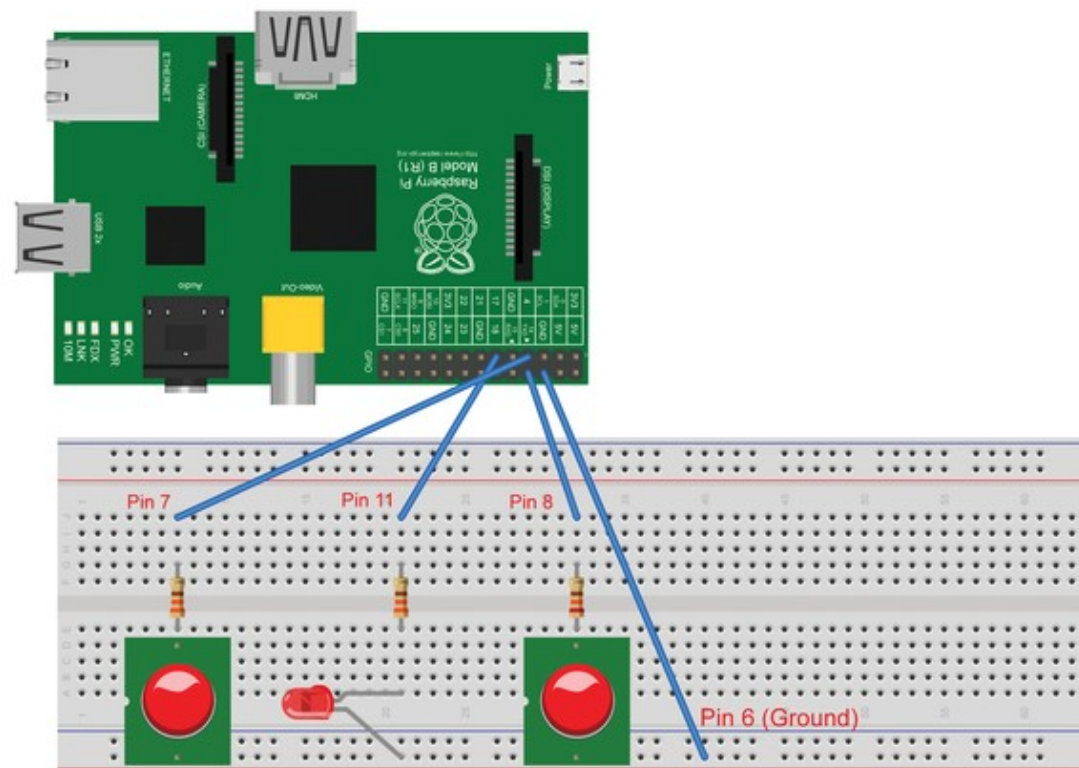


O teu LED está a **PISCAR**

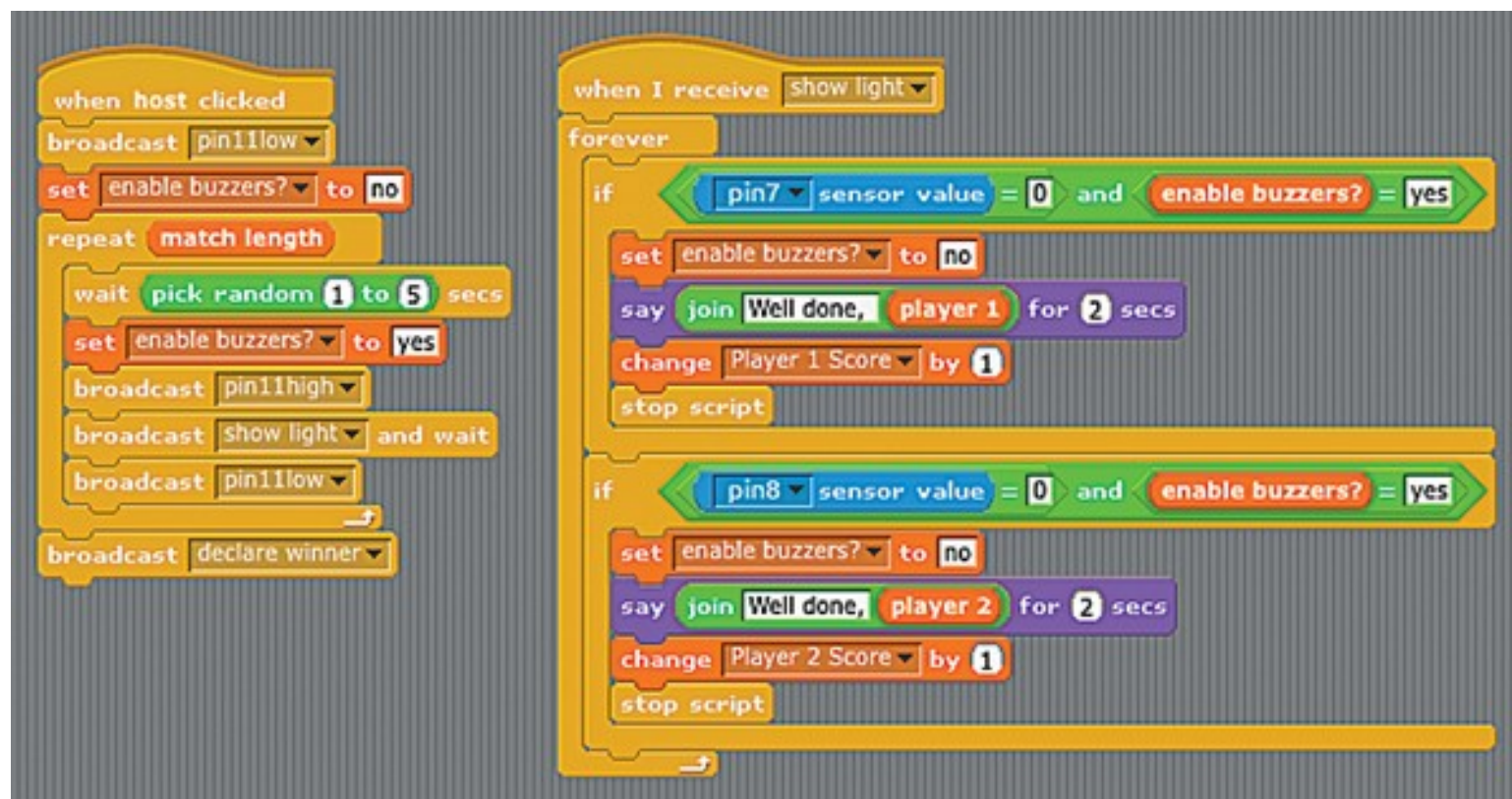
Vamos agora fazer uma pequena demonstração usando o Pi e alguns componentes. Nesta demonstração simples usaremos o Scratch para programar o Pi para criar um jogo multiplayer de reação rápida.

Para ver qual dos jogadores tem a reação mais rápida, é necessário ligar duas campainhas à breadboard. O programa Scratch controlará a permissão dos jogadores carregarem na campainha, detetando o primeiro sinal sonoro, e mantendo a pontuação.

Para a campainha, vou usar dois simples botões de pressão, um para cada um dos dois jogadores. A campainha do jogador um ligar-se-á ao pino 7, e a do jogador dois ligar-se-á ao pino 8. (Ver figura à direita para os circuitos da campainha). Para detetar os cliques da campainha, o guião do Scratch precisa de monitorizar a entrada nos pinos do Raspberry Pi. O ScratchGPIO trata os pinos 3, 5, 7, 8, 10, 19, 21, 22, 23, 24, e 26 como entrada.



Os dois guiões mostram a lógica principal do jogo. O guião que começa com o bloco **when host clicked** inicia o jogo. Acrescentei uma nova variável chamada **enable buzzers?** e atribuí-lhe o valor **no**, para evitar que os jogadores se carreguem demasiado cedo.



```
when host clicked
  broadcast pin1low
  set enable buzzers? to no
  repeat match length
    wait pick random 1 to 5 secs
    set enable buzzers? to yes
    broadcast pin1high
    broadcast show light and wait
    broadcast pin1low
  broadcast declare winner

when I receive show light
  forever
    if pin7 sensor value = 0 and enable buzzers? = yes
      set enable buzzers? to no
      say join Well done, player 1 for 2 secs
      change Player 1 Score by 1
      stop script
    if pin8 sensor value = 0 and enable buzzers? = yes
      set enable buzzers? to no
      say join Well done, player 2 for 2 secs
      change Player 2 Score by 1
      stop script
```

O loop **repeat()** utiliza uma variável chamada **match length** para controlar quantos jogos ou partidas cada jogo terá. Esta abordagem permite aos jogadores alcançar os dois melhores de três, por exemplo. Esta configuração pode ser controlada através de um controlo deslizante no stage.

Depois de esperar um tempo aleatório de **1 a 5** segundos, o guião ativa as campainhas e depois liga o LED, transmitindo uma mensagem de **pin11high**. Quando o LED acende, este é o sinal para os jogadores carregarem no botão. O bloco **broadcast(show light) and wait** coordena a programação necessária para detetar as campainhas dos jogadores.

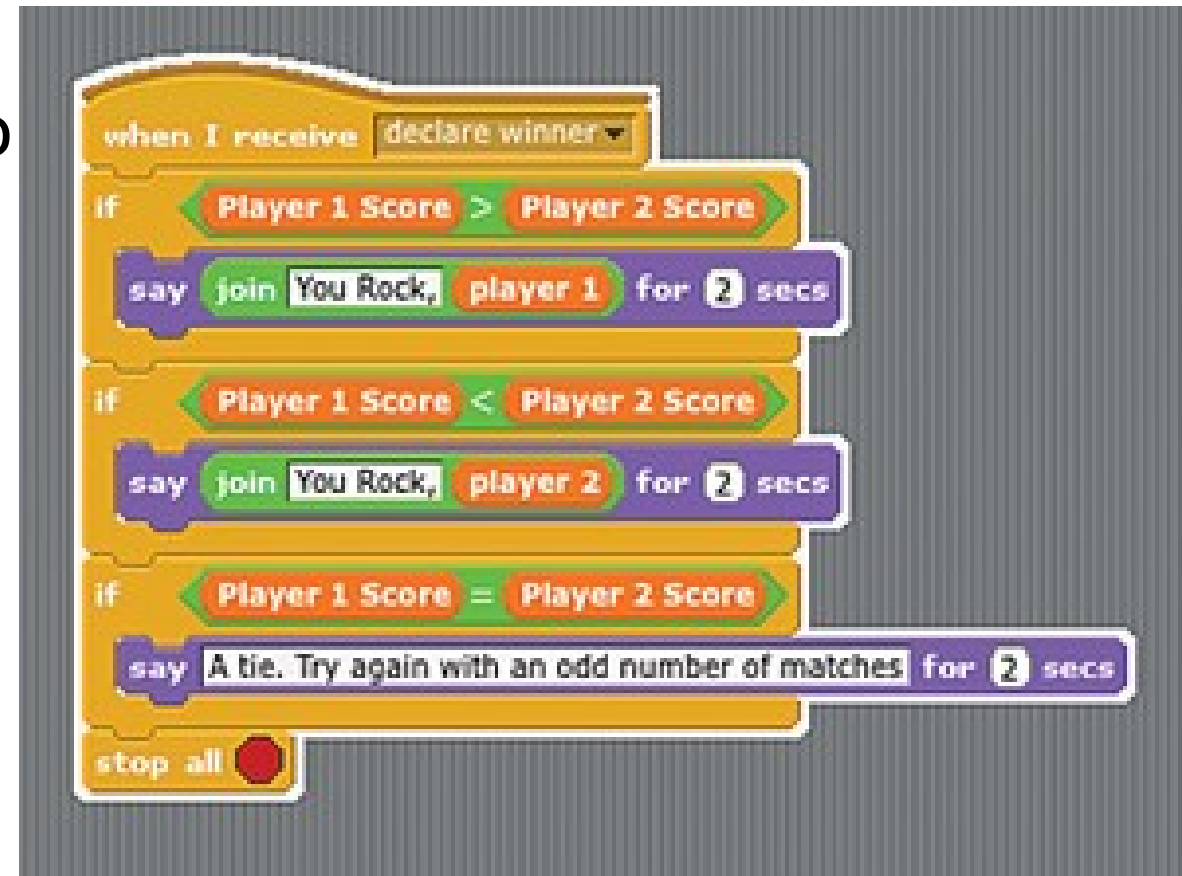
O guião **when I receive(show light)**, deteta a entrada do pino. O loop **forever** assegura que o guião continuará a correr até que um dos jogadores carregue na campanha. O bloco **()sensor** value tem opções descendentes para as entradas do ScratchGPIO.

Além disto, o primeiro bloco **if()** verifica se o pino 7 (jogador 1) está ligado e se a variável **enable buzzer?** é igual a **yes**. Se ambas as condições forem verdadeiras, o guião aumenta a pontuação do jogador, desativa o sinal sonoro, e para o guião atual. O segundo bloco **if()** executa a mesma verificação para o pino 8 (jogador 2).

Os pinos de entrada têm um valor por defeito de **1** na posição "off". Quando a entrada é detetada, por exemplos clicando na campanha, o valor do pino torna-se **0**. Isto pode ser visto ao ativar o valor do sensor do pino7 no stage monitor.

A desativação da campanha impede o outro jogador de carregar e obter um ponto. Parar o guião quebra o ciclo **forever** e sinaliza o bloco **when host clicked** do guião para recomeçar no ponto onde parou, que será para executar o bloco **broadcast(pin11low)** para desligar o LED. A vantagem de usar o bloco **broadcast(show light) and wait** é que o guião irá pausar até que todos os guiões **when I receive(show light)** recebidos estejam completos.

Então, o guião continua a acender aleatoriamente o LED para o número especificado de correspondências. No final do jogo, o programa declara o vencedor com o bloco **broadcast(declare winner)** no final do guião **when host clicked script**. O guião **when I receive(declare winner)** determina que jogador teve a pontuação mais alta e anuncia o vencedor.



VAMOS JOGAR!

Por favor não partam os botões...

Referências

<https://learn.adafruit.com/programming-with-scratch-on-raspberry-pi/demo>

[https://www.raspberry-pi-geek.com/Archive/2014/07/Creating-a-multiplayer-quick-reaction-game/\(offset\)/2](https://www.raspberry-pi-geek.com/Archive/2014/07/Creating-a-multiplayer-quick-reaction-game/(offset)/2)

<https://www.instructables.com/id/Physical-Computing-Scratch-20-for-Raspberry-Pi/>



**Reviving hands-on educational play for
learning skills of tomorrow**
PROJECT N° 2019-1-UK01-KA201-061466

CIVIC

 **idec**


ΕΚΠΑΙΔΕΥΤΗΡΙΑ
ΠΛΑΤΩΝ

 **elearn**
EUROPEAN DIGITAL LEARNING NETWORK


scholé

Emphasys
CENTRE

 **CCS**
Digital Education