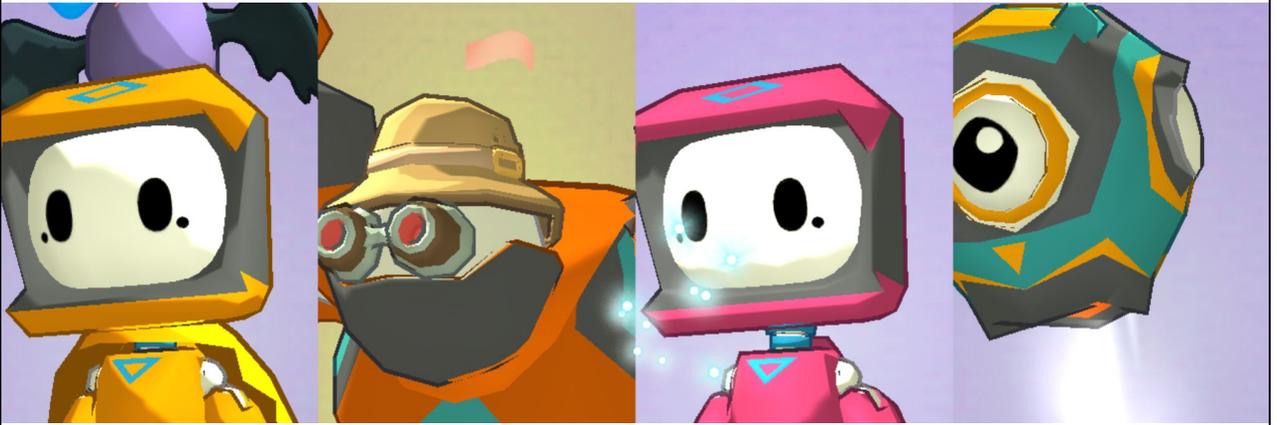


tg



SWITCH & GLITCH:  
GO TO Command

```
new 1 x
1 @ECHO OFF
2
3 :top
4 ECHO This is an endless loop
5 GOTO top
6
```

- *GOTO* or *GO TO* is a command that changes where the program is reading. In the image, the “GOTO top” takes the program to the “:top” section. This causes the program to run forever, in an **endless loop!**
- Easy to cause an endless loop by using this command.

**Tell** the students that programming is **giving commands to a computer**: the computer reads the code beginning from the top (the first row).

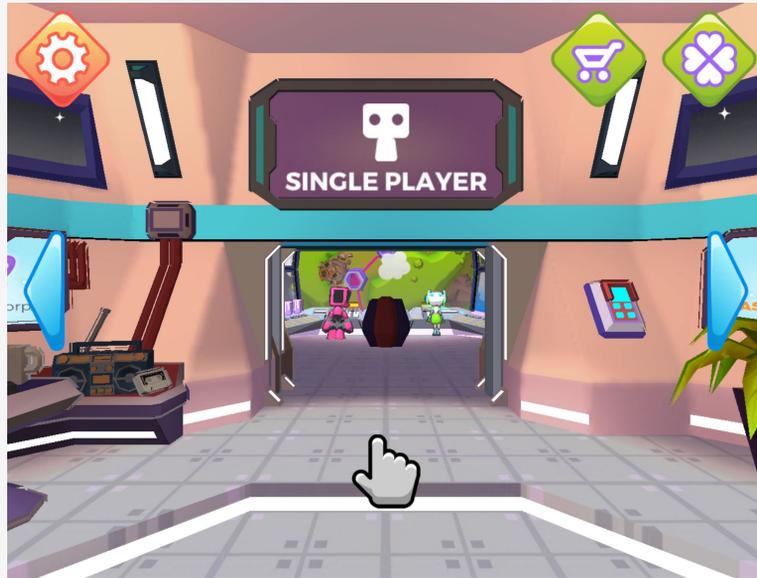
**Discuss** with your students for their opinion on:  
*“What problems can arise if a program is left in an endless loop?”*

**Most relevant observations for learning:**

- A computer executes commands beginning from the top (the first row)
- A program can be made the wrong way, causing it to be uninterruptible.
- Programming languages are evolving and not all programming commands remain necessary.



## NAVIGATE: SPACE HOPPER



**Tell the students:** *“Today we’re playing Switch & Glitch Jungle Level 5 and onward. If you didn’t get to level 5 yet, just continue where you left off.”*



## LESSON GOAL

Complete Switch & Glitch **single player levels 5-8.**

Think - how would you reach the goal faster by using the jumping mushrooms?

**You can let the students play the game until there is about 10 minutes left of the lesson. Then, move on to the next slide.**

The goal of this lesson is to play the jungle planet's levels 5-8.

**As the teacher, you can challenge the students to achieve the full 3 stars in some or all the levels.**



## SHARE & DISCUSS

- Did one of you earn three stars?
- How hard was it to use the jumping mushrooms?
- Did you consistently use 6 commands each turn?  
Or less?

Go through the yields of the lesson with the students. How did the students manage? Pay attention: **which things were most difficult** and does **a specific student need extra help?** It is likely that there will be extensive variation among the students, as some may have played similar commercial games, giving them an edge.

If you task the students with taking screenshots of the game, the Cretubbles service works excellently to build this kind of portfolio.



## TASKS AFTER PLAYING

- Your friend asks you 'what is the harm if computer sticks in endless loop'? How do you answer?

You can give the students time to consider these questions. They can answer them as pairs, as small groups, or individually, whichever way you prefer. These tasks function as control questions and help students demonstrate their learning which will help your assessment.

**DISCUSS:** You can continue to discuss why it is important to make prim and proper code. Students might have seen examples of the bad code so they could have stories about computers, cell phones, and machines not working.



## ADDITIONAL TASKS

### Teach the Teacher-Robot

Play as a programmer and a robot. The teacher has hidden treasure in the schoolyard or classroom. In this competition, the programmer needs to give orders for the robot to find the treasure and repeat the progress until the robot finds it. He can't just give the order 'find treasure'!

**Note:** If you have free time, you can let students play the robot. The idea is to learn to give explicit orders and notice things that can go wrong. Like "turn left" - robot would interpret this as her left and not programmers.