UNIVERSITÀ DEL PIEMONTE ORIENTALE

Programmazione con Snap4Arduino

Paola Giannini

Informazioni preliminari

Accedete a

https://orienta.dir.uniupo.it/course/view.php?id=43

Con le vostre credenziali Facebook o gmail

Trovate questi lucidi e un contenitore in cui salvare i vostri programmi.

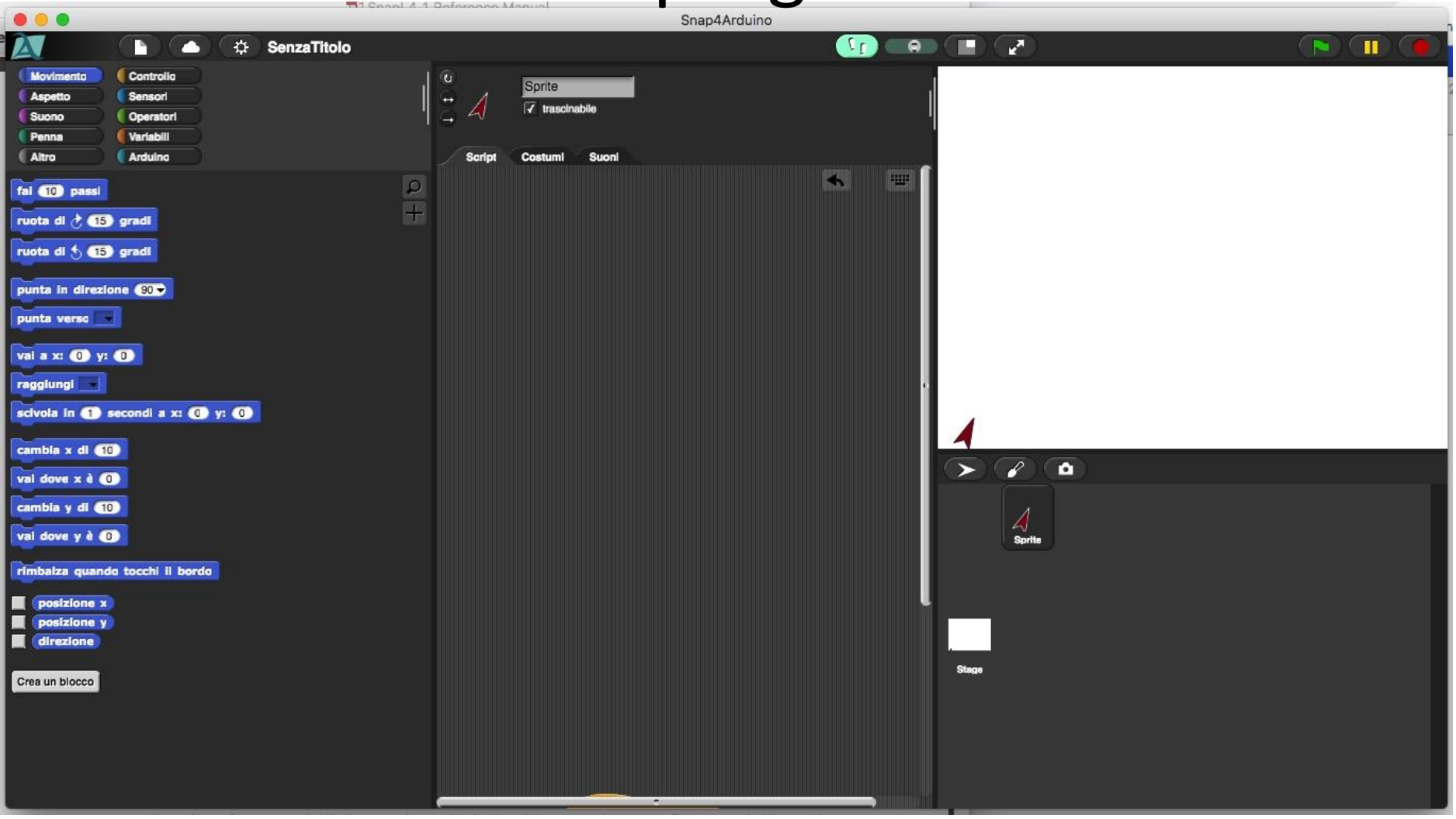
Snap

- Ambiente di programmazione, con insieme di strumenti che aiutano nell'attività di programmazione:
  - Editor (in questo caso visuale, anche la sintassi è visuale)
  - Accesso alle librerie (vedrete l'estensione per Arduino)
  - Possibilità di eseguire parti di codice (script)
- Organizzazione del codice:
  - Script associati con Sprites e Sfondi
  - Multithreading (più script in esecuzione allo stesso tempo)
  - Attivazione da eventi (tastiera, sprite entra in contatto con un colore di sfondo, bandiera, ec…..). Possibilità di generare eventi (broadcast o mirati ad uno Sprite).
  - Interazione simile a una Graphical User Interface (GUI)

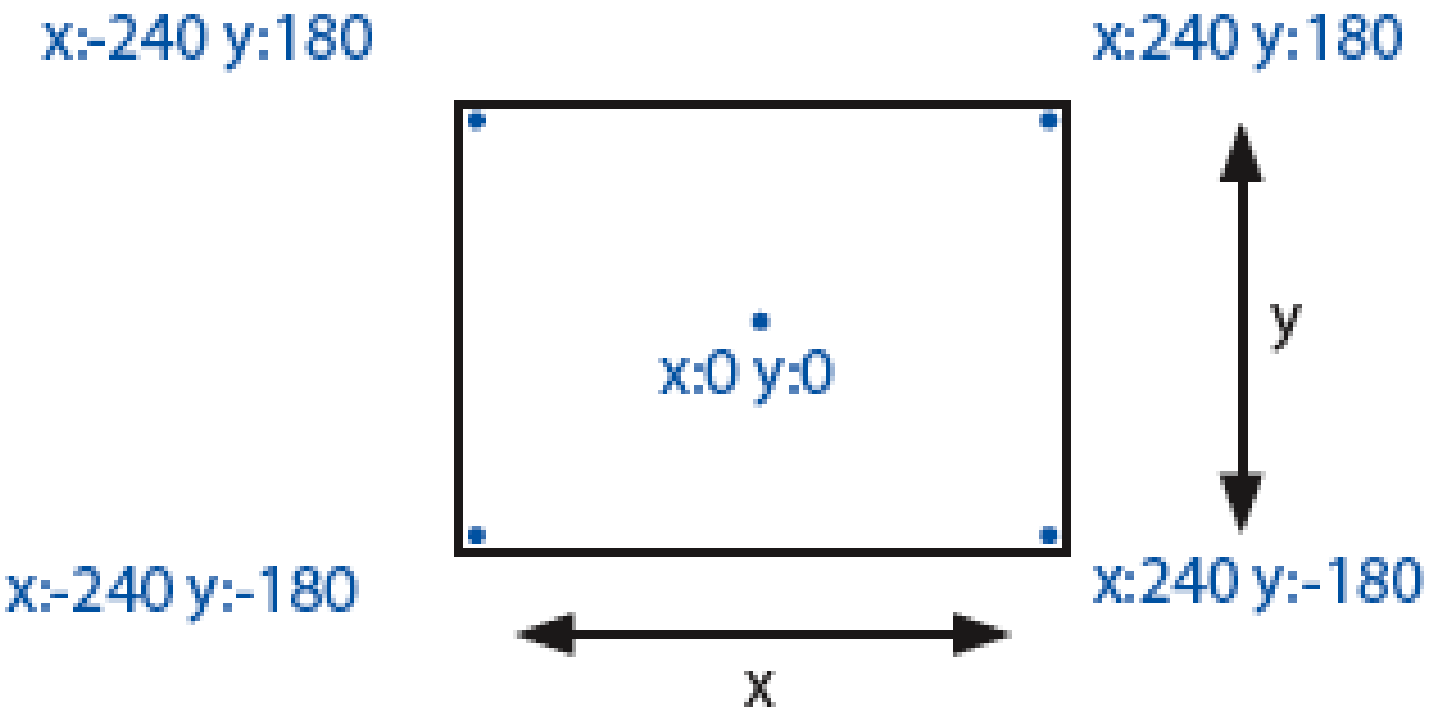
Per fare programmi in Snap potete anche accedere al sito

https://snap.berkeley.edu/snap/snap.html

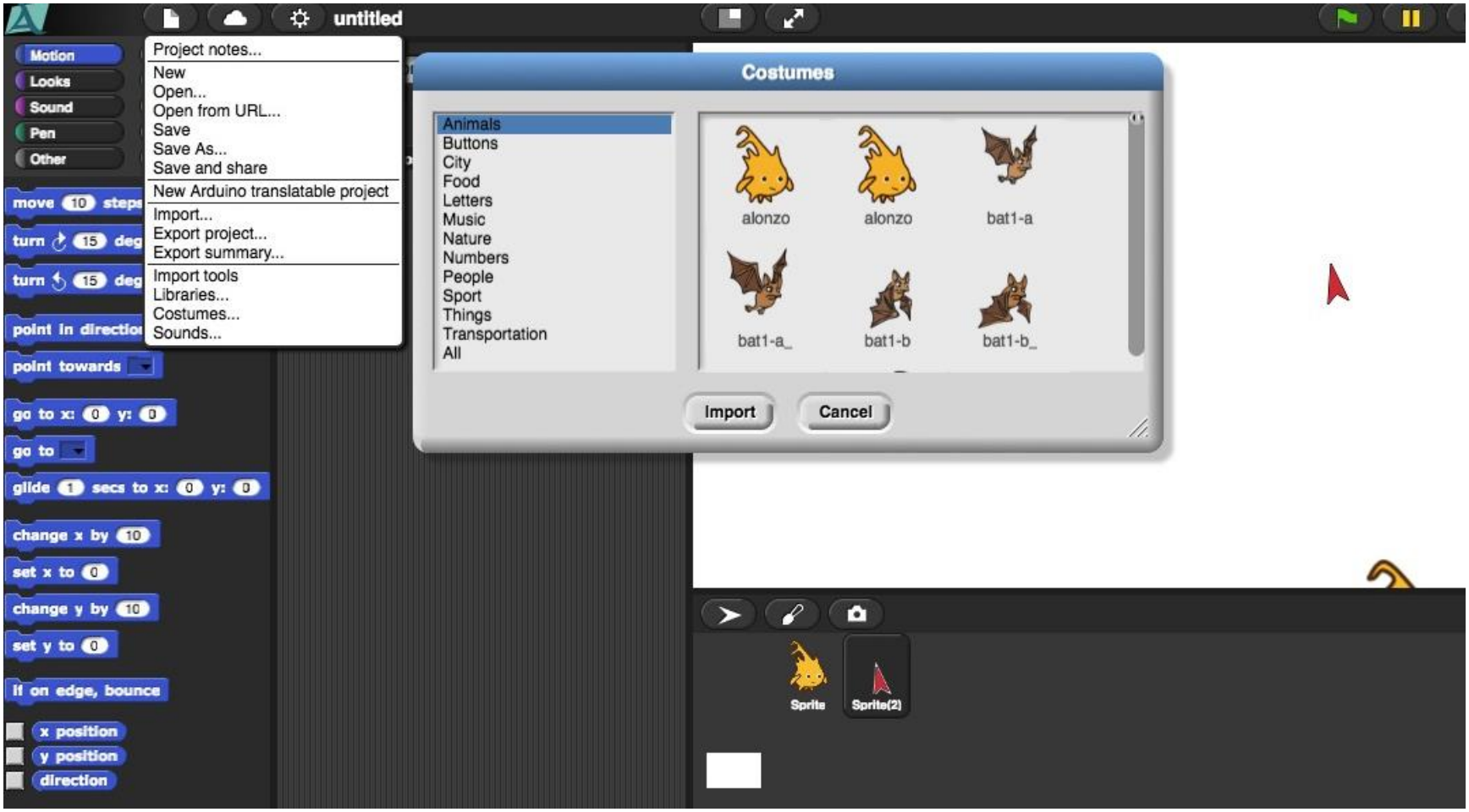
L’ambiente di programmazione



Lo stage



Area Sprite/Stage (può essere programmato anche lo stage)



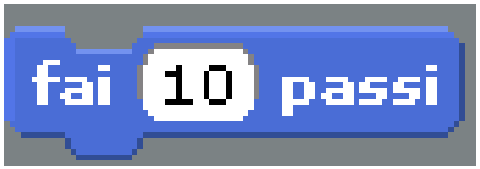
Associati con Sprite/Stage ci sono le aree: Script, Costumi, Suoni

Elementi fondamentali di un linguaggio di programmazione

- Snap e' un linguaggio "visuale" imperativo, la computazione e' fatta eseguendo azioni su uno stato). I compomenti di base sono:
  - Le istruzioni (azioni possibili)
  - Le espressioni (calcoli necessari per ottenere valori)
  - Le variabili (i contenitori di valori che rappresentano lo stato della computazione e che possono essere modificati)

Istruzioni elementari (fanno qualcosa che modifica lo stato del programma)

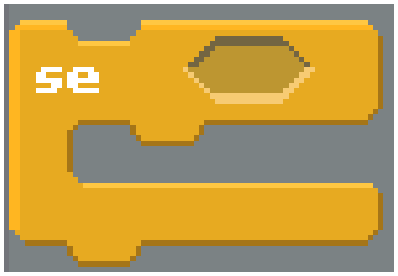
- In Scratch le Istruzioni si chiamano Blocchi in aggiunta a leggere/scrivere, definire e/o modificare variabili abbiamo istruzioni che agiscono sugli sprite:



- Forma caratteristica delle istruzioni, che possono essere messe in sequenza (il ; del C, C++, Java)

Istruzioni condizionali

- Condizionali: if then e if then else



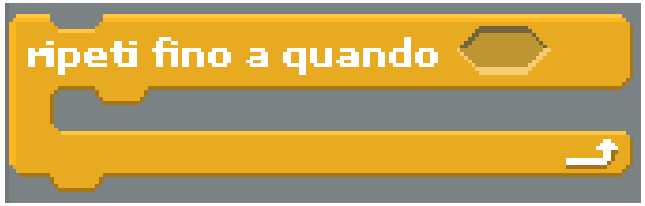
- Attivazione basata su eventi

Iterazione

- Ripetizione (infinita):

- For:

- Ripetizione condizionata: se la condizione è vera abbandona la ripetizione altrimenti esegue le istruzioni del ciclo:



- Attesa condizionale e ripetizione attesa:

Espressioni (calcolare un valore)

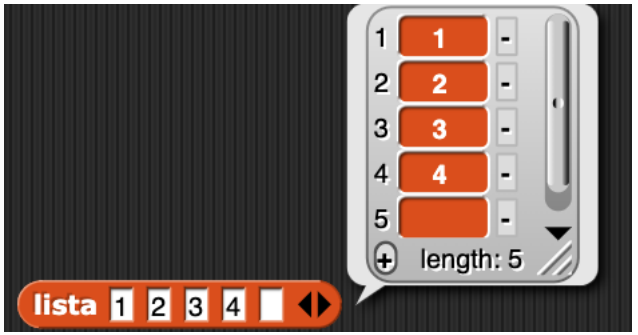
- Le espressioni ritornano un valore, ad esempio nel blocco Fai 10 passi o Attendi fino a quando si possono inserire espressioni:

ritornano vero o falso (booleane)

ritornano un numero

Variabili

- Locali allo Sprite o globali a tutti gli Sprite
- Possono denotare valori numerici, stringhe (operazione di concatenazione, ecc..) e/o
- Liste (aggiungere, rimuovere, cercare)



Multi-Threading

- Thread: gruppo di istruzioni che può essere eseguito in parallelo ad altri thread.
- In C, C++ si programmano usando le primitive del sottostante sistema operativo, altri linguaggi quali Java, C# ecc.. sono implementati (Jvm o .Net) attraverso macchine virtuali multithreaded

Multi-threading in Snap

- In Scratch i Thread si chiamano Script iniziano con un blocco arrotondato, ad esempio

si attiva quando si preme la bandiera verde (in genere inizio del programma).

Se ci sono più Script (Thread) con questo blocco di inizio, vengono attivati tutti quando si clicca sulla bandiera verde.

Eventi

- Gli Script comunicano fra loro:
  - generando eventi e
  - rispondendo ad eventi (handling)
- Comunicazione inter/intra Sprite:

Esercizio 1: Realizziamo un gioco

Gioco: Dobbiamo indovinare il numero che uno sprite ha pensato!

Come realizziamo questo gioco?

Algoritmo:

1) Lo sprite "pensa" un numero (diciamo fra 0 e 30)
2) Lo sprite ci "dice" di dirgli quale numero crediamo abbia pensato e noi gli "rispondiamo" un numero
3) Lui ci "dice" se il nostro numero e'
   - maggiore,
   - minore o
   - uguale

   al numero che ha pensato
4) Se e' uguale abbiamo vinto altrimenti ritorniamo al passo 2)

(Un po' più complicato: Prima di pensare il numerolo sprite ci chiede un valore minimo e un massimo e pensa un numero in quell'intervallo)

Implementiamo l'algoritmo in snap

- “pensa” =>
  - genera **un numero a caso** fra 0 e 30 e lo assegna ad un variabile **numero** (per ricordarselo!).
    Le variabili sono la “memoria” del programma! Ci sono istruzioni **porta**, **cambia**,.. per modifcare le variabili
- “chiede” =>
  - c'e' un'istruzione, **chiedi…**, che permette allo sprite di scrivere sullo stage un messaggio e crea anche un campo nel quale noi possiamo scrivere. Quando digitiamo un “a capo” quello che abbiamo scritto viene letto e assegnato alla variabile **risposta** (questa e’ definite da Snap e la troviamo nei sensori)
- “dice” =>
  - c'e' anche un'istruzione, **dire…**, che permette allo sprite di scrivere sullo stage un messaggio

Ripetizione e selezione

Le istruzioni 2, 3, 4 vengono ripetute.

Ci sono vari tipi di ripetizione.

Noi useremo **per sempre** perche' se voi continuate a dare la stessa risposta e questo non e' il numero il ciclo potrebbe andare avanti all'infinito "e oltre…."

Pero' se indovinate volete bloccare tutto. Per questo c'e' **ferma tutto**

In dipendenza del fatto che la risposta sia maggiore o minore o uguale dovete eseguire istruzioni diverse, per questo ci sono l condizionali, **se** e se **altrimenti**.

Esercizio 1: l'algoritmo

Creiamo 2 variabili
- **numero** (il numero da indovinare)
- **risposta** (il numero che voi avete proposto)

Porta **numero** a numero casuale da 0 a 30
Ripeti per sempre:
- Chiedi un numero
- Se **risposta** = **numero**,
  allora Dire INDOVINATO e blocca tutto
- Se **risposta** > **numero**,
  allora Dire TROPPO ALTO
- Se **risposta** < **numero**,
  allora Dire TROPPO BASSO

Osservare e bloccare l'esecuzione degli Script

- Potete mettere in pausa l'esecuzione di uno script con il bottone centrale di

- Oppure controllare l'esecuzione delle istruzioni cliccando sui passi

- Viene "illuminata" l'istruzione eseguita

Esercizio 2: uno sprite che indovina il nostro numero

- Ci dice di pensare un numero fra 0 e 30 e poi inizia a
- dirci dei numeri, e ora siamo noi a dover dare l'informazione se quello che lui ci ha detto
  - È troppo alto
  - È troppo basso
  - È giusto
- Lo fa chiedendoci di inserire
  - 0 se il numero che ci ha detto è più piccolo di quello che abbiamo pensato
  - 1 se è uguale
  - 2 se è più grande
- Nel secondo caso dopo essersi congratulato con se stesso finisce il programma.
- Negli altri casi ritorna a proporci un numero
- (Potremo anche far si' che lo sprite capisca e ci dica se abbiamo BARATO, cioè ad un certo punto abbiamo dato una risposta sbagliata?)

Esercizio 2: l'algoritmo

Creiamo 3 variabili

min e **max** (il numero da indovinare sta fra queste due variabili)

media (il numero che viene proposto dal computer, media fra **min** e **max**)

Chiedi di pensare un numero

Ripeti per sempre:

Porta **media** a arrotonda((**min**+**max**)/2)

Dire un numero

Chiedi se e' maggiore (2) minore (0) o uguale (1) a quello pensato

Se **risposta** = **1**,

allora Dire INDOVINATO e blocca tutto

Se **risposta = 0** ,

allora Porta **max** a **media**-1

Se **risposta = 2**,

allora Porta **min** a **media**+1

Esercizio 2: (il computer si accorge che abbiamo barato

Creiamo 3 variabili

min e **max** (il numero da indovinare sta fra queste due variabili)

media (il numero che viene proposto dal computer, media fra **min** e **max**)

Chiedi di pensare un numero

Ripeti per sempre:

Se **min** > **max**,
    allora Dire HAI BARATO! e blocca tutto

Porta **media** a arrotonda((**min**+**max**)/2)

Dire un numero

Chiedi se e' maggiore (2) minore (0) o uguale (1) a quello pensato

Se **risposta** = **1**,
    allora Dire INDOVINATO e blocca tutto

Se **risposta = 0** ,
    allora Porta **max** a **media**-1

Se **risposta = 2**,
    allora Porta **min** a **media**+1

Esercizio 3: versione ancora piu' complicata

- Il computer conosce solamante che deve indovinare un numero positivio, ma non sa quale e' il massimo possible!
- Come fare?
  - Prima cercare il massimo e minimo (all'inizio lo sapete) e poi fare l'algoritmo precedente
- Cerca il massimo e minimo
- minimo=0
- limite =100

  porta possibileMax a numero a caso fra minimo e limite
  chiedi se il numero pensato e' minore o uguale a possibileMax
  se si' allora massimo= possibileMax
  altrimenti minimo=possibleMax limite=limite+100

-