RAPPRESENTAZIONE DATI GEOMETRICI SU PANNELLO GRAFICO

Premessa

In questo esempio viene approntato un modello che, nell'ambito di un programma di calcolo, può servire a creare una pagina web (page) da utilizzare per l'impostazione numerica inerente applicazioni tecniche rappresentabili geometricamente in ambiente grafico.Per attuare tale finalità si può strutturare una page come quella che si propone in questo contesto. Essa contiene:

- una porzione della page, (0) localizzata nella parte alta, in cui inserire il titolo dell'applicazione;
- un ambito (1) per l'impostazione dei dati necessari a risolvere la problematica per
- la quale si intende creare il "software";
- un pannello (3) in cui vengono rappresentati graficamente, per verificare la congruità geometrica, i dati da acquisire;
- altri pannelli (2), come quello in cui è riportata questa descrizione, per fornire informazioni e, eventualmente, guidare l'utilizzatore nell'uso del programma;
- un pannello (4) in cui sono allocati i comandi per avviare le elaborazioni.

Tutto ciò può concretizzarsi con l'approntamento del codice di elaborazione, nelle fasi che seguono.

Fase 1. Struttura pagina

Inizialmente si definisce l'ambiente <html></html> all'interno del quale vengono inseriti i codici inerenti l'applicazione di cui al titolo inserito nel tag <title></title>	<html> <head> <title>TITOLO Pannello impostazione grafia</title> </head></html>
Nella sezione < <i>style></i> <i style> si definiscono le dimensioni dei pulsanti e delle caselle Text create nella sezione < <i>body></i> <i body>. In questa vengono utilizzati, altresì, i tag < <i>table></i> e < <i>div></i> per definire i pannelli (o porzioni di pagina) che servono a realizzare l'applicazione di cui alla descrizione precedente.	<pre><style type="text/css"> #Text2 {height: 20px; width: 70px;} #Button1 {width: 140px; height: 20px;} #Button2 {width: 140px; height: 28px;} #Text1 {height: 20px; width: 70px;} #Text3 {height: 20px; width: 70px;} #Text4 {height: 20px; width: 70px;} #Button2 {width: 126px; height: 20px;} </style></pre>
Nella sezione < <i>script></i> <i script> si definiscono le <i>"function"</i> legate ai pulsanti <i>Button1</i> e <i>Button2</i>	<script> function Button1_onclick() { } function Button2_onclick() { } </script>
Rif1 Nella sezione <i><body> <!--<body-->></body></i> Con il tag <i></i> , viene creata la tabella in cui sono inserite le caselle text e i pulsanti <i>button</i> precedentemente dimensionati	> > Titolo1 > > Pannello (0) > > Informazioni utilizzo programma(0) <ti>nput id="Text1" type="text" /> <input id="Text2" type="text"/> <input id="Text3" type="text"/> <input id="Text4" type="text"/> > > <ti>> > > > > > > > > </ti></ti>

In questo modo la porzione di schermo (di pagina), conseguente all'esecuzione del codice su esposto, attuabile incollando il relativo testo in un qualunque documento con ..HTML. appare come di seguito.

Titolo1	Pannello (0)	
Informazioni utilizzo programma(0)	Disegna linea	Pulisci video

Rif 2 <br

Procedendo nella stesura del codice, con un altro tag da inserire fra il testo già approntato e la chiusura della sezione *</body>*, viene definita un'altra tabella da utilizzare successivamente per l'impostazione dei dati.

	1) Pannello 1	
	<pre>Impostazione dati 2</pre>	
	<pre>Impostazione dati 3</pre>	
	<pre>Impostazione dati 4</pre>	
	· ·	

Così operando la porzione di schermo, conseguente all'esecuzione del codice integrato come precedentemente esposto, appare come di seguito.

Titolo1 Informazioni utilizzo programma(0)	Pannello (0) Disegna linea	Pannello (0) Disegna linea Pulisci video				
1) Pannello 1						
Impostazione dati 2 Impostazione dati 3						
Impostazione dati 4						
Rif3 → <div></div>						

con il tag *<div>* riportato a fianco, inserito prima della chiusura della sezione *</body>*, si crea il pannello per inserire il testo che accompagna la descrizione che si vuole fornire in fase di esecuzione.

Si inserisce infine il tag ...div... relativo al pannello da predisporre per l'inserimento del disegno. Rif4 →

<div></div>
Pannello 3)
<pre> </br></pre>

Complessivamente, con l'aggiunta di questo ultimo codice, si ottiene la seguente videata, nella quale sono evidenziati i riquadri contenenti gli elementi, che si è deciso di creare.

Titolo1 Informazioni utilizzo programma(0)	Pannello (0) Disegna linea Pulisci video	
1) Pannello 1 Impostazione dati 2 Impostazione dati 3 Impostazione dati 4	Tag	
Pannello2 IMPOSTAZIONE DATI GEOMETRICI In questa sezione viene approntato un modello che, nell'ambito di un prog geometricamente un ambiente grafico. Per attuare tale finalità si può strutturare una page come quella che si prop -una porzione della page,(0) localizzata nella parte alta, in cui inserire i ti -un ambito (1) per l'impostazione dei dati necessari a risolvere la problemu -un pannello (3) in cui vengono rappresentati graficamente, per verificare -lattri pannelli (2), come quello in cui e irportate questa illustrazione, per fi -un pannello (4) in cui sono allocati i comandi per avviare le elaborazioni.	gramma di calcolo, può servire a creare una pagina oonein questo contesto. Essa contiene: tolo dell'applicazione; atica per la quale si intende creare il "software"; la congruità geometrica, i dati da acquisire; ornire informazioni ed, eventualmente, guidare l'ut	web (page) da utilizzare per l'impostazione numerica inerente applicazioni tecniche rappresentabi lizzatore nell'uso del programma;
Pannello 3) 		Tag <div> </div>
Edita commento codiceLink	/	

In questa si può notare pure un link "Edita commento codice" creato con l'istruzione che segue:

Edita commento codice </body> </html>

Fase 2: Denominazione Pannelli "id" e la proprietà colore sfondo

Avvia codice Fase 2 →

2.1 Denominazione dei vari pannelli "id "

Con i codici riportati nel riquadro a fianco, i pannelli di cui sopra(Rif1, Rif2, Rif3, Rif4) si denominano nella sezione <body></body> rispettivamente, con la proprietà *id*, come "Pannello0", "Pannello1", "rrr" e " SecondoDiv"

<!Rif1>
<!Rif2>
<!Rif3><div id="rrr">
<!Rif3><div id="SecondoDiv">

2.2 Definizione del colore dello sfondo "background color "

Per meglio individuarli, i \ sopra denominati è opportuno evidenziarli, con il "colore di sfondo", o meglio con la proprietà "background color". Ciò si può attuare nei tag "tab" e "div", per i vari pannelli utilizzando l'attributo style="..."

<pre><!--Rif2--></pre>
<pre><!--Rif2--> </pre>
TITOLO 1 TITOLO 2 Informazioni utilizzo programma Avvio elaborazione Desegna linea Pullsci video
IMPOSTAZIONE DATI GEOMETRICI In questa secione view approvatato un modello che, nell'ambito di un programma di calcolo, può servire a creare una pagina web (page) da utilizzare per l'impostazione numerica inerente applicazioni tecniche rappresentabili geometricano della page) (Dolizzata atella prisori di titolo dell'applicazione;
<pre><!--Rif3--> <div id="rrr" style="background-color:</pre></pre>
<pre><!--Rif4--><div id="SecondoDiv" style="background-color:</pre></pre>

Fase 3: Tipo di caratteri...Organizzazione della pagina

Avvia codice Fase 3 ->

3.1 Il tipo di caratteri

Per modificare i caratteri dei testi da utilizzare per la costruzione della pagina è opportuno utilizzare la relativa proprietà nel tag <body ...> con l'attributostyle= ... e la proprietà ...font-family.... a cui associare il carattere ...Arial...

<body style="font-family: Arial" >

3.2 Organizzazione della pagina

È opportuno a questo punto, organizzare la pagina come nello schema di seguito rappresentato. I pannelli descritti vengono rispettivamente così allocati:

- il ..table... pannello(0) nella parte alta della pagina;
- il ...table .. "pannello (1)" e il ..div .. "rrr", opportunamente dimensionati in altezza e in larghezza, nella • parte destra;

ed. infine.

il pannello da utilizzare per la grafica ..div.. "SecondoDiv" nella parte centrale.

Pannello (0) Titolo1 Informazioni utilizzo programma(0) x=865	Papello (0)Titolo2	Top=75
1) Pannello 1 Ir table pannello(1) Impostazione dati 4 Pannello (2) Impostazione patri ceometrici		•
In questa sezione viene approntato un modello che, nell'ambito di un programma di calcolo, può servire a creare una pagina (page) da utilizzare per l'impostazione numerica inerente applicazioni teoniche rappresentabili geometricamente in ambiente grafico. Per attuare tale finalità si può strutturare una page come quella che	Pannello grafico	width = 1000
si propone in questo contesto. Essa	rainello granco	1000
 Juna pCIV 'rrrr'' localizzata nella parte alta, in cui inserire il titolo dell'applicazione; -un ambito (1) per l'impostazione dei dati necessari a risolvere la problematica per la quale si intende creare il "software"; -un pannello (3) in cui vengono rappresentati graficamente, per verificare la congruità geometrica, i dati da acquisire; -attri pannelli (2), come quello in cui è riportate questa descrizione, per fornire informazioni ed, eventualmente, guidare l'utilizzatore nell'uso del programma; -un pannello (4) in cui sono allocati i 	div" SecondoDiv	
•width = 295•	• width = 1000	+
← Left=325•	+-	

Per la sua peculiarità finalizzata a rappresentare in modo ottimale la geometria di ciò che si vuole impostare, si ritiene, inizialmente, di operare il dimensionamento del pannello grafico "SecondoDiv", assegnando a questo dimensioni, in altezza e in larghezza, pari a 1000 px

(width: 1000px; height: 1000px).

E' opportuno inoltre assegnare al tag div la proprietà position: absolute in questo modo il pannello, lateralmente può essere affiancato al pannello(1) e al div.."rrr" assegnando il valore (left:325px). Verticalmente, il pannello può essere posizionato sotto il pannello(0) assegnando il valore (top: 75px)

<div id="SecondoDiv" style="background-color: #F0F0FF; width: 1000px; height: 1000px;</pre> position: absolute; top: 75px; left:325px;"> </div>

In sintonia con le dimensioni precedentemente espresse si assegna la larghezza dei due pannelli laterali pari a width: 295px.

<div id="rrr" style="background-color: #FEF8EF; width: 295px; height: 384px; ">
Relativamente a questo ultimo pannello, atto ad accogliere dei testi alfa-umerici è opportuno pure
giustificare il testo inserendo la proprietà text-align: justify;

Inoltre è opportuno inserire in questo <div> una barra laterale che possa far scorrere il testo all'interno della finestra definita, funzionalmente, indipendentemente dalla lunghezza da questo, in proporzione all'altezza della pagina testé definita.

Ciò si attua aggiungendo l'istruzione overflow:auto
<div id="rrr" style="background-color: #FEF8EF; width: 295px; height: 384px;
overflow:auto; text-align: justify;">



Fase 4: disegno figure geometriche: LINEA

Avvia codice Fase 4 →

Il disegno di una linea, utilizzando il TAG <svg> </svg> , si può attuare nel modo seguente.

1) Inizialmente si definiscono le variabili, necessarie in tutti i codici del programma, che con il significato dei simboli si riportano a fianco.

//Dichiarazi	
Larghezza e Altezza finestra grafica	var AltezzaFinestra = 800
	Var LargnezzaFinestra = 800
Coordinata dai nunti aka dafiningana la linga	Var Puntox = parseFloat(0)
Coordinate dei punti che definiscono la linea	van PuntOF = parserioal(0)
	van Punt1X = parseFloat(0)
Variabila atta a compare il tasta aba costituisco il	var full($f = parseridat(0)$
"TAO down down do the family nells norming di	Val cescobisegno - Scillig
TAG <svg></svg> da trasferire nella porzione di	
pagina (pannello), definita nella sezione	
<div></div> , creata per rappresentare gli oggetti da	
disegnare.	
Sottostringa contenente le istruzioni per disegnare	var lineaNn = String
una linea.	
Attributi da inserire nell'elemento geometrico da	<pre>var lineColoreLinea = String</pre>
disegnare (linea, cerchio, rettangolo, poligono	var spessoLin = parseFloat(0)
acc): in questo caso dell'elemento che definisce	
Attribute aba definizza la sfanda del nonnello	van Sfondcol - Stning
granco su cui disegnare	von fatterala - 50
Variabile che definisce il fattore (in pixel) legato	Var fattscala = 50
all'unità di misura del sistema di riferimento	
cartesiano. Ciò è utile per definire la scala del	
disegno: in questo caso, si assume come singola	
unità da rappresentare un valore di 50 pixel.	
	<pre>function Button1_onclick(){</pre>
2) Nella function Button1_onclick() (Disegna	//INIZIO Disegna linea
linea su sfondo) si assegnano, alle variabili che	testoDisegno = ""
definiscono la linea, i valori delle coordinate dei punti	<pre>var estrem0x = parseFloat(0.00):</pre>
estremi (0,0) (9.00,10.00).	<pre>var estrem0y = parseFloat(0.00)</pre>
	<pre>var estrem1x = parseFloat(9.00)</pre>
	<pre>var estrem1y = parseFloat(10.00)</pre>
3) Si trasformano queste coordinate, definite rispetto	
ad un sistema di riferimento assegnato, in pixel. Nel	Puntox = estremox * fattscala
caso in esame una unità nel sistema di riferimento	PuntOY = estremØy * fattscala
dei punti appena definiti corrisponde a 50 pixel. Tutto	Puntix = estremix * fattscala
ciò in relazione al fattore di trasformazione (o fattore	Puntly = estremix * fattscala
di scala), inizialmente definito: (fattscala=50).	
1) Si definisce la spessore della linea pari a 1 e il	///IBlue transparent Agua Euchsia Red Black
relative colore coordiondole ad ecompio fre clouri	White Vellow none Green Margon Silvers
di sussifi	Colorelines = "hlue": spessol in = 1
5) Si compone la stringa "testoDisegno" che	testoDisegno = "";
dotinicco il codico da incoriro noll'olomonto divo	
	<pre>testoDisegno = testoDisegno + "<svg 2mm="" abaalutaa="" atula="" lafta<="" lagaitiana="" pre="" tana=""></svg></pre>
denominato "SecondoDiv" nel cui ambito,	<pre>testoDisegno = testoDisegno + "<svg "="" "prv"<="" "px;="" +="" -="" 0px;="" absolute;="" alterrafinestra="" boight:="" larghezzafinestra="" left:="" position:="" pre="" style='position: absolute; top: 0px; left: Opv: uidth:", langhages="income of "provide the statement of the statement</td></tr><tr><td> </</td><td><pre>testoDisegno = testoDisegno + "<svg style=' top:="" width:"+=""></svg></pre>
 <!--</td--><td><pre>testoDisegno = testoDisegno + "<svg <="" style='position: absolute; top: 0px; left: 0px; width:"+ LarghezzaFinestra + "px; height: " + AltezzaFinestra + "px;" + ".' td=""></svg></pre></td>	<pre>testoDisegno = testoDisegno + "<svg <="" style='position: absolute; top: 0px; left: 0px; width:"+ LarghezzaFinestra + "px; height: " + AltezzaFinestra + "px;" + ".' td=""></svg></pre>
denominato "SecondoDiv" nel cui ambito,assumendo validità, serve a disegnare la linea.Questa verrà disegnata nella finestra grafica definitadai valori LarghezzaFinestra, AltezzaFinestra	<pre>testoDisegno = testoDisegno + "<svg style='position: absolute; top: 0px; left: 0px; width:"+ LarghezzaFinestra + "px; height: " + AltezzaFinestra + "px;" + ";'><defs></defs>" disegnolinea() //(PifE)acquiciesi > lineaWa</svg></pre>
denominato "SecondoDiv" nel cui ambito,assumendo validità, serve a disegnare la linea.Questa verrà disegnata nella finestra grafica definitadai valori LarghezzaFinestra, AltezzaFinestrapari a 800x800 impostati nella sezione//	<pre>testoDisegno = testoDisegno + "<svg style='position: absolute; top: 0px; left: 0px; width:"+ LarghezzaFinestra + "px; height: " + AltezzaFinestra + "px;" + ";'><defs></defs>" disegnolinea() //(Rif5)acquisisci>lineaNn (/ (Pif 6)</svg></pre>
denominato "SecondoDiv" nel cui ambito, assumendo validità, serve a disegnare la linea. Questa verrà disegnata nella finestra grafica definita dai valori LarghezzaFinestra, AltezzaFinestra pari a 800x800 impostati nella sezione// dichiarazione variabili	<pre>testoDisegno = testoDisegno + "<svg style='position: absolute; top: 0px; left: 0px; width:"+ LarghezzaFinestra + "px; height: " + AltezzaFinestra + "px;" + ";'><defs></defs>" disegnolinea() //(Rif5)acquisisci>lineaNn // (Rif.6) testoDisegno = testoDisegno + lineaNn</svg </pre>
denominato "SecondoDiv" nel cui ambito, assumendo validità, serve a disegnare la linea. Questa verrà disegnata nella finestra grafica definita dai valori LarghezzaFinestra, AltezzaFinestra pari a 800x800 impostati nella sezione// dichiarazione variabili	<pre>testoDisegno = testoDisegno + "<svg style='position: absolute; top: 0px; left: 0px; width:"+ LarghezzaFinestra + "px; height: " + AltezzaFinestra + "px;" + ";'><defs></defs>" disegnolinea() //(Rif5)acquisisci>lineaNn // (Rif.6) testoDisegno = testoDisegno + lineaNn testoDisegno = testoDisegno + "</svg </pre>
denominato "SecondoDiv" nel cui ambito, assumendo validità, serve a disegnare la linea. Questa verrà disegnata nella finestra grafica definita dai valori LarghezzaFinestra, AltezzaFinestra pari a 800x800 impostati nella sezione// dichiarazione variabili	<pre>testoDisegno = testoDisegno + "<svg style='position: absolute; top: 0px; left: 0px; width:"+ LarghezzaFinestra + "px; height: " + AltezzaFinestra + "px;" + ";'><defs></defs>" disegnolinea() //(Rif5)acquisisci>lineaNn // (Rif.6) testoDisegno = testoDisegno + lineaNn testoDisegno = testoDisegno + "" SecondoDiv_innerHTML = testoDisegno</svg </pre>
denominato "SecondoDiv" nel cui ambito, assumendo validità, serve a disegnare la linea. Questa verrà disegnata nella finestra grafica definita dai valori LarghezzaFinestra, AltezzaFinestra pari a 800x800 impostati nella sezione// dichiarazione variabili	<pre>testoDisegno = testoDisegno + "<svg style='position: absolute; top: 0px; left: 0px; width:"+ LarghezzaFinestra + "px; height: " + AltezzaFinestra + "px;" + ";'><defs></defs>" disegnolinea() //(Rif5)acquisisci>lineaNn // (Rif.6) testoDisegno = testoDisegno + lineaNn testoDisegno = testoDisegno + "" SecondoDiv.innerHTML = testoDisegno //case_FINE_Disegna_linea=</svg </pre>
denominato "SecondoDiv" nel cui ambito, assumendo validità, serve a disegnare la linea. Questa verrà disegnata nella finestra grafica definita dai valori LarghezzaFinestra, AltezzaFinestra pari a 800x800 impostati nella sezione// dichiarazione variabili Per attuare lo scopo suddetto si utilizza il metodo "inpert/TMU"	<pre>testoDisegno = testoDisegno + "<svg style='position: absolute; top: 0px; left: 0px; width:"+ LarghezzaFinestra + "px; height: " + AltezzaFinestra + "px;" + ";'><defs></defs>" disegnolinea() //(Rif5)acquisisci>lineaNn // (Rif.6) testoDisegno = testoDisegno + lineaNn testoDisegno = testoDisegno + "" SecondoDiv.innerHTML = testoDisegno //FINE Disegna linea-</svg </pre>

Nella composizione della stringa *"testoDisegno",* come si nota nel codice precedentemente esposto (Rif.5) viene effettuata la ...chiamata.. alla **function** *"disegnolinea()"*

In tale ambito viene approntata la parte di codice, compresa in una sottostringa denominata lineaNn che, con l'utilizzo delle variabili PuntOX, PuntOY, Punt1X, Punt1Y, ColoreLinea, spessoLin, consente di definire l'elemento <line> </line> da associare a detta variabile lineaNn

```
function disegnolinea() {
    lineaNn = "<line x1='" + PuntOX + "' y1='" +
    PuntOY + "' x2='" + Punt1X + "' y2='" +
    Punt1Y
    lineaNn = lineaNn + "'style='stroke:" +
    ColoreLinea + "' stroke-width='" + spessoLin
    + "'/>"
}
```

Detto codice, aggiunto alla stringa testoDisegno, con il metodo *innerHTML* (Rif.6 testoDisegno = SecondoDiv.innerHTML = testoDisegno testoDisegno + lineaN) genera, nel pannello appositamente predisposto, il disegno della linea

E' opportuno evidenziare, osservando il disegno che scaturisce dall'esecuzione del programma, che il sistema di riferimento di default nel metodo <svg>, diversamente di quello più usuale, ha origine nel vertice in alto a sinistra del pannello con coordinate x e y crescenti, rispettivamente, verso sinistra e verso il basso.



Si vedrà, nella Fase successiva, come riportare tale riferimento a quello più usuale, con origine nel vertice in basso a sinistra e direzione positiva degli assi diretti rispettivamente verso l'alto e verso sinistra.

<u>Avvia codice Fase 5 →</u>

Dopo avere definito il codice per disegnare una linea, può essere utile disegnare sul pannello grafico uno sfondo, costituito da griglia a passo costante, sulla quale rappresentare graficamente dati geometrici o risultanze frutto di elaborazioni.

Si appronta, dunque, allo il pulsante Button3 (Disegno sfondo), appositamente creato nella sezione <body> </body>

nella cui relativa *function* è inserito il codice descritto dai commenti appositi che, unitamente alle varie istruzioni, si riporta di seguito.

	<pre>function Button3_onclick() {</pre>
// Si cancella il contenuto del div "SecondoDiv" e	SecondoDiv.innerHTML = ""
della stringa "testoSfondo"	testoSfondo = ""
// Si definisce con il colore giallo lo sfondo del	Sfondcol = "Yellow"
pannello grafico	
// Si definisce il tag <svg> </svg>	testoSfondo =testoSfondo + " <svg< td=""></svg<>
	tyle= position: absolute; top: 0px; lett: 0px: width:" + LarghezzaEinestra + "px:"
// Si compone il testo "testoSfondo" da trasferire nel	testoSfondo = testoSfondo + "height: " +
tag <div> con il metodo innerHTML</div>	AltezzaFinestra + "px; background-color:" +
5	<pre>Sfondcol + ";'><defs></defs>"</pre>
// Si effettua la chiamata alla "function	<pre>DisegnoSfondo() // acquisisci</pre>
DisegnoSfondo()" nel cui ambito viene composta la	>SfondoNn
parte di stringa "SfondoNn" (sottostringa) da	testoStondo = testoStondo + StondoNn
aggiungere a quella in corso di definizione	+ \/ SVg/
"TestoStondo"	SecondoDiv innonHIML - tostoSfondo
() Si trasferisce nel lay <ui>la stilliga testosionuo</ui>	
cosi come precedentemente composta.	
	<pre> function DisegnoSfondo() { </pre>
// Si cancella il contenuto della sottostringa	SfondoNn =""
// Si fissano, oltre allo spessore e al colore, le	ColoreLinea = "red" ; spessoLin = .3
coordinate iniziali a partire dalle quali disegnare le	PuntOX = -fattscala; PuntOY = -fattscala
linee orizzontali atte a comporre la griglia a maglie	Punt1X = 100*fattscala; Punt1Y = -fattscala
quadrate.	
//Operando con il ciclo for si disegnano dette linee	<pre>for (i = 1; i <= 300; ++i) {</pre>
parallele incrementando, sequenzialmente di un	PuntOY = PuntOY + fattscala
passo costante pari a, "fattscala", il valore delle	Puncty = Puncoy
ordinate Y	dicognalines()
//Si effettua la chiamata all'iunction disegnolinea()	sfondoNn = SfondoNn + lineaNn
disegnare la generica linea "lineaNn" orizzontale	}
	PuntOX = -fattscala
//Si effettua, con la stesa seguenza, il disegno delle	PuntOY = -fattscala
linee verticali atte a completare la griglia a maglie	Punt1X = -fattscala
quadrate (fattscala x fattscala)	Punt1Y = 100* fattscala
	tor $(1 = 1; 1 \le 300; ++1) \{$ PuntOX = PuntOX + fattecala:
	Punt1X = Punt0X
	disegnolinea()
	SfondoNn = SfondoNn + lineaNn
	}
	}

In questa fase si è provveduto altresì, come riportato a fianco di questa colonna, nella porzione di codice predisposto allo scopo //---- Dichiarazione variabili, a ridefinire, in funzione dell'altezza dello schermo screen.height, le variabili AltezzaFinestra e larghezzaFinestra, entrambe inizialmente definite pari a 800.

<scri< th=""><th>pt></th><th></th><th></th></scri<>	pt>		
//	- Dichiarazione var	riabili -	
var	AltezzaFinestra =	screen.he	eight
var	LarghezzaFinestra	=1.5 *	screen.height
	•		
<scri< th=""><th>pt></th><th></th><th></th></scri<>	pt>		

Titolo1.		Titolo2				
Informazioni utilizzo programma		Avvio elaborazion	Disegna linea	Pulisci video	Disegno sfondo	
1) Pannello 1 Impostazione dati 2 Impostazione dati 3 Impostazione dati 4						
In guesta sezione viene approntato un modello che, nell'ambito di un programma di calcolo, può servire a creare una pagina web (page) da utilizzare per l'impostazione numerica inerente applicazioni tecniche rappresentabili geometricamente in ambiente artifica.						
ambene granco. Per attuare tale finalità si può strutturare una page come quella che si propone in questo contesto. Essa contiene: -una porzione della page,(0) localizzata nella parte alta, in cui						
inserire il titolo dell'applicazione; -un ambito (1) per l'impostazione dei dati necessari a risolvere la problematica per la quale si intende creare il "software"; -un pannello (3) in cui vengono *						
Edita commento codice						

Fase 6: Disegno di una LINE su sfondo con griglia

<u>Avvia codice Fase 6 →</u>

Si può procedere, a questo punto, a rappresentare, sullo sfondo così predisposto, la linea precedentemente disegnata nella Fase 4. Ciò si può attuare aggiungendo il codice già predisposto in tale fase nella function Button3_onclick attivabile con il pulsante Button3 che, ora opportunamente, si può definire, nella sezione <body></body> con la denominazione (Disegna linea su sfondo).

II La parte di codice relativa al disegna dello sfondo, si può completare con il disegno della linea precedente operando:	<pre>function Button3_onclick() { SecondoDiv.innerHTML = "" Sfondcol = "Yellow" testoDisegno = "" testoDisegno = testoDisegno + "<svg style='position: absolute; top: 0px; left: 0px;width:" + LarghezzaFinestra + "px;" testoDisegno = testoDisegno + "height: " + AltezzaFinestra + "px;background-color:"+ Sfondcol + ";'><defs></defs>" DisegnoSfondo() // acquisisci>SfondoNn</svg></pre>
// -l'assunzione delle coordinate estreme della linea	<pre>var estrem0x = parseFloat(0.00) var estrem0y = parseFloat(0.00) var estrem1x = parseFloat(9.00) var estrem1y = parseFloat(9.00)</pre>
// -la trasformazione delle coordinate in px	PuntOX = estrem0x * fattscala PuntOY = estrem0y * fattscala Punt1X = estrem1x * fattscala Punt1Y = estrem1y * fattscala
//-l'acquisizione della sottostringa "lineaNn" ottenuta, dopo avere fissato il colore e lo spessore della linea, con lafunction disegnolinea()	ColoreLinea = "blue"; spessoLin = 1 disegnolinea() // acquisisci>lineaNn
// -il disegno della linea su sfondo inserendo nella div SecondoDiv le sottostringhe "SfondoNn" e "lineaNn"	<pre>testoDisegno = testoDisegno + SfondoNn + lineaNn+"" SecondoDiv.innerHTML = testoDisegno }</pre>

function disegnolinea() {
 lineaNn = "";
 lineaNn = "<line x1='" + PuntOX + "' y1='" + PuntOY + "' x2='" + Punt1X + "' y2='" + Punt1Y
 lineaNn = lineaNn+ "'style='stroke:" + ColoreLinea + "' stroke-width='" + spessoLin + "'/>"



Fase 7: Trasformazione degli assi

Avvia codice Fase 7 →

Per ricondurre la rappresentazione grafica precedentemente illustrata all'usuale riferimento cartesiano, con origine nella parte in basso a sinistra e il verso positivo degli assi x-y rispettivamente crescente da sinistra verso destra e dal basso verso l'alto (assi colorati in rosso), effettuando qualche riflessione, si può operare con la trasformazione della function disegnolinea().

Questa si può attuare con l'assunzione delle coordinate dei punti (TrasfPuntOX, TrasfPuntOY, TrasfPunt1X, TrasfPunt1Y) da utilizzare nel codice che consente di disegnare, nella tag <svg></svg>, la linea, o di disegnare, con differenti codici, altre figure geometriche.





Fase 8 Inserimento scala graduata

Avvia codice Fase 8 →

È opportuno ora, conoscendo il codice per disegnare le linee, inserire, nel nuovo sistema di riferimento, la scala graduata, che consente di visualizzare, le coordinate (x,y) nel sistema di riferimento, convenzionalmente, assunto come in precedenza.



Ciò si può realizzare con le istruzioni seguenti, da inserire nella function DisegnoSfondo(), attuando sequenzialmente "chiamate" alla function **disegnolinea()** che, in relazione alle coordinate dei segmenti (delle lineette) da disegnare trasversalmente agli assi x e y, consentono di definire, ciclicamente, le sottostringhe **lineaNn**, contenenti i codici inerenti le lineette da disegnare, da inserire nella stringa principale **SfondoNn**. Per chiarezza di esposizione si può far riferimento ai commenti inseriti in dette istruzioni che prevedono:

function DisegnoSfondo()

```
var VerPuntOY = 0
```

//l'impostazione delle coordinate (x,y) iniziali delle "lineette" (con ascisse invariabili per quelle da disegnare trasversalmente all'asse y)

PuntOX = 0

```
PuntOY = 0
Punt1X = -.08 * fattscala
Punt1Y = PuntOY
```

// la definizione ciclica delle ascisse (x) delle "lineette" da disegnare a passo costante ortogonalmente all'asse y

```
for (i = 0; i <= 3000; ++i) {
    PuntOY = PuntOY + fattscala / 10
    Punt1Y = PuntOY
    ColoreLinea = "black"
    spessoLin = .5</pre>
```

// la chiamata alla function disegnolinea() con l'acquisizione della sottostringa lineaNn

```
disegnolinea() >> acquisizione della sotto stringa lineaNn
SfondoNn = SfondoNn + lineaNn
```

} var VerPuntOY = 0

//l'impostazione delle coordinate (x,y) iniziali delle "lineette" (con ordinate invariabili per quelle da disegnare trasversalmente all'asse x)

```
PuntOX = 0
PuntOY = 0
Punt1X = -.16 * fattscala
Punt1Y = PuntOY
```

// la definizione ciclica delle ordinate (y) delle "lineette" da disegnare a passo costante ortogonalmente all'asse x

```
for (i = 0; i <= 3000; ++i) {
    PuntOY = PuntOY + fattscala / 2
    Punt1Y = PuntOY
    ColoreLinea = "black"
    spessoLin = 2
    disegnolinea()
    SfondoNn = SfondoNn + lineaNn
}
</pre>
```

......

Avvia codice Fase 9 →

Avendo già definito i codici per rappresentare la scala graduata, si vuole con le operazioni che seguono, inserire per ogni relativo intervallo il testo che la caratterizza.

Ciò si concretizza con l'elemento <text></text> con il quale è possibile rappresentare,

il testo, definito inizialmente con la variabile numerica **testoXY**, comprendente i valori delle ascisse e delle ordinate riferite agli assi X e Y. Questo verrà inserito in un determinato punto di coordinate

(TrasfPuntOX=100+PuntOX , TrasfPuntOY=0.7*AltezzaFinestra-PuntOY)

desunte dalla trasformazione di quelle (PuntOX, PuntOY) riferite al sistema di riferimento originario di default-

//---- Dichiarazione variabili -----

var testoXY = parseFloat(0)
var testoNn = String
//---- Fine dichiarazione variabili linea-----

Allo scopo viene creata la "function disegnaTesto()" atta a creare una sottostringa, **testoNn** che, similmente a come operato per la definizione delle variabili **SfondoNn**, **lineNn**, consente, con il metodo "*innerHTML*", di concretizzare il suddetto intento.

```
function disegnaTesto() {
  TrasfPuntOX = 100 + PuntOX
  TrasfPuntOY = 0.7 * AltezzaFinestra - PuntOY
  testoNn = ""
  testoNn = "<text x='" + TrasfPuntOX + "' y='" + TrasfPuntOY + "' font-
  size='12' font-family='Arial'>" + testoXY + "</text>"
}
```

Ciò si attua aggiungendo alla **function DisegnoSfondo()**, le istruzioni seguenti **function DisegnoSfondo()**



for (i = 0; i <= 300; ++i) {</pre>

2) All'interno del ciclo **for** (...){ ...} si definisce per ogni iterazione l'ascissa **PuntOX** ottenuta da quella inerente il ciclo moltiplicato per il **"fattscala"** (oppure attenuta incrementando del valore **"fattscala"** quella precedente)

PuntOX = i * fattscala - 10

3) All'interno del ciclo **for (...)** si definisce, per ogni iterazione, il **testoXY** corrispondente alla i-esima iterazione

PuntOX = -2 / 3 * fattscala



for (i = 0; i <= 300; ++i) {</pre> 2) All'interno del ciclo for (...) { ...} si definisce per ogni iterazione l'ascissa PuntOY ottenuta da quella inerente il ciclo moltiplicato per il valore fattscala PuntOY = i * fattscala 3) All'interno del ciclo for (...) { ...} si definisce per ogni iterazione il testoXY corrispondente alla i-esima iterazione testoXY = i.toFixed(2) // <--Riferimento asse y originario</pre> 4) All'interno del ciclo for (....) { ...}, si acquisisce la sottostringa testoNn effettuando la chiamata alla function disegnaTesto() disegnaTesto() //----> testoNino 5) All'interno del ciclo for (....) { ...} si compone per incrementi successivi la stringa SfondoNn SfondoNn = SfondoNn + testoNn // } In definitiva, dopo avere attuato, nella function DisegnoSfondo(), le operazioni connesse al metodo innerHTML di seguito riportato testoDisegno = testoDisegno + SfondoNn testoDisegno = testoDisegno + lineaNn + "</svg>" SecondoDiv.innerHTML = testoDisegno si ottiene la seguente videata



Fase 10: Imposta coordinate linea

Avvia codice Fase 10 →

Dopo avere composto la page per effettuare il disegno di una linea, aventi coordinate predefinite, in questa fase, si vuole modificare il codice scritto nei paragrafi precedenti, assumendo dall'esterno le coordinate dei punti estremi che la definiscono.

```
Allo scopo bisogna agire nella "function" legata al
                                                  function Button3_onclick() {
                                                                                     -----
pulsante "Disegna linea" Button3 onclick(), in
                                                     estrem0x = parseFloat(0.00); var
                                                var
questa basta modificare le seguenti istruzioni, in cui
                                                estrem0y = parseFloat(0.00); var estrem1x
a mo' d'esempio erano sati assunti i valori di default
                                                    parseFloat(9.00);
                                                                          var
                                                                                 estrem1y
                                                =
(0.00, 0.00)-(9.00, 9.00).
                                                parseFloat(9.00); //----- }
```

In pratica bisogna ridefinire	
il pannello1, con	Imposta coordinate "linea"[mt]
l'insorimonto dollo casollo	Punto 1
	Xp1 ripetere 23 volte per posizionare Text5;
con le quali acquisire le	<input id="Text5" type="text"/>
coordinate dei punti estremi	Yp1 ripetere 23 volte per posizionare Text6. ;
(Xp1 Yp1) - (Xp2 Yp2) che	<input id="Text6" type="text"/>
definiscono la linea da	Punto 2
	Xp2 ripetere 23 volte per posizionare Text7;
disegnare.	<input id="Text7" type="text"/>
Si tratta di inserire fra il tag	Yp2 ripetere 23 volte per posizionare Text8;
< <table></table>	<input id="Text8" type="text"/>
istruzioni inoronti allo	
caselle testo l'ext5, l'ext6,	
Text7 eText8	

e trasformare la "table" pannello1 nel modo seguente:

		Imposta coordinate "line	ea" [mt]
1) Pannello 1		Punto 1	
Impostazione dati 2	\sim	Xp1	
Impostazione dati 3		Yp1	
Impostazione dati 4		Punto 2	
		Xp2	
		Yp2	

È opportuno, per facilitare la fase di "input", inserire dei controlli che guidino nella fase di input, avvertendo su eventuali dimenticanze nell'impostazione dei dati

Bisogna controllare, infatti, con le istruzioni seguenti, inserite fra quelle già esposte in precedenza ed evidenziate su sfondo colorato, che detti valori siano impostati, nelle caselle testo previste allo scopo. var estrem0x = parseFloat(Text5.value)

```
Text5.value = estrem0x.toFixed(2)
if (Text5.value == "NaN") {
      alert("Non è stata impostata l'ascissa del punto 1! Bisogna
      impostarla."); Text5.value = ""; Text5.focus()
return }
var estrem0y = parseFloat(Text6.value)
Text6.value = estrem0y.toFixed(2)
if (Text6.value == "NaN") {
     alert("Non è stata impostata l'ordinata del punto 1!
     Bisogna impostarla."); Text6.value = ""; Text6.focus();
return }
var estrem1x = parseFloat(Text7.value)// parseFloat(9.00)
Text7.value = estrem1x.toFixed(2)
if (Text7.value == "NaN") {
     alert("Non è stata impostata l'ascissa del punto 2! Bisogna
     impostarla."); Text7.value = ""; Text7.focus()
     return}
```

var estrem1y = parseFloat(Text8.value)//parseFloat(9.00)

```
Text8.value = estrem1y.toFixed(2)
if (Text8.value == "NaN") {
    alert("Non è stata impostata l'ordinata del punto 2! Bisogna
    impostarla."); Text8.value = "";Text8.focus()
(return }
```

È utile, inoltre, inserire un controllo che impedisca, nell'acquisizione dei dati, di impostare caratteri diversi da quelli attesi. Ciò si può attuare, con l'evento "onkeypress", collegato ad una determinata "function", abilitando solamente i tasti necessari a comporre i valori da impostare, costituiti dalle dieci cifre, variabili da 0 a 9, e dal punto decimale.

Allo scopo si può fare riferimento ai valori **"event.keyCode"** collegati all'uso della tastiera. In pratica, bisogna disabilitare, quelli indesiderati, non compresi nel seguente intervallo in cui, però, risulta inserito pure il codice 47, corrispondente al carattere "/", non utile allo scopo che si intende perseguire.

Tasto	•	/	0	1	2	3	4	5	6	7	8	9
codice	46	47	48	49	50	51	52	53	54	55	56	57

Bisogna verificare i codici generati dalla pressione dei vari tasti, con un controllo **if(..) {..}** che, con l'istruzione "event.returnValue= false", neutralizzi i tasti i cui codici event.keyCode soddisfano le seguenti relazioni :

```
event.keyCode<45
event.keyCode>57
event.keyCode= 47
```

o meglio bisogna prevedere il controllo seguente

```
if (event.keyCode < 45 || event.keyCode > 57 || event.keyCode == 47) {
        event.returnValue = false;
}
```

}

da inserire in delle function, collegate alle caselle Text5, Text6, Text7, Text8, con l'evento "onkeypress" di seguito opportunamente evidenziate

```
<input id="Text5" type="text" onkeypress="ParteNum5()"/>
<input id="Text6" type="text" onkeypress="ParteNum6()"/>
<input id="Text7" type="text" onkeypress="ParteNum7()"/>
<input id="Text8" type="text" onkeypress="ParteNum8()"/>
```

Si può notare che tutte le **function ParteNum5()** ... ParteNum6() ... ParteNum7() ... ParteNum8(), di seguito riportate, oltre a contenere il controllo **if(..){..}** necessario al suddetto scopo, ne contengono un altro atto a controllare la pressione del tasto "return"

if (event.keyCode == 13) {focus()}

con il quale, confermando i dati impostati nelle varie caselle, trasferisce, con l'istruzione Text...focus(), il puntatore a quelle che sequenzialmente bisogna riempire.

```
Text5(inputXp1) → Text6(inputyXp1) → Text7(inputXp2) →Text8(inputYp2)
function ParteNum5() {
    if (event.keyCode < 45 || event.keyCode > 57 || event.keyCode == 47) {
        event.returnValue = false;
}
if (event.keyCode ==13) { Text6.focus() }
```

```
}
```

}

```
function ParteNum6() {
    if (event.keyCode < 45 || event.keyCode > 57 || event.keyCode == 47) {
        event.returnValue = false;
}
```

```
if (event.keyCode == 13) { Text7.focus() }
}
```

```
function ParteNum7() {
    if (event.keyCode < 45 || event.keyCode > 57 || event.keyCode == 47) {
        event.returnValue = false;
}
if (event.keyCode == 13) {Text8.focus() }
```

function ParteNum8() {

```
if (event.keyCode < 45 || event.keyCode > 57 || event.keyCode == 47) {
    event.returnValue = false;
```

```
if (event.keyCode == 13) { Button3.focus()}
}
```

}

Completato l'inserimento di tutti i dati, il controllo viene trasferito al pulsante atto ad avviare la fase di elaborazione. Ciò si concretizza con l'inserimento nell'ultima "function ParteNum8()", connessa all'ultima casella testo (Text8), dell'istruzione Button3.focus().

Text5 \rightarrow Text6 \rightarrow Text7 \rightarrow Text8 \rightarrow Button3(avvio Elaborazione)

È opportuno inserire, infine, nella "function Button3_onclick()",le istruzioni (evidenziate con grassetto) con le quali, attivando la **function disegnaTesto()**, inserire gli indici e le coordinate che caratterizzano la linea da disegnare.

```
function Button3 onclick() {
                                            Si stabiliscono le coordina (PuntOX, PuntOY)
                                            PuntOX = estrem0x * fattscala
del punto a partire del quale si intende
                                            PuntOY = estrem0y * fattscala
riportare il testo relativo al primo indice
Dopo aver definito la variabile testoXY, si
                                            testoXY = "P1(X1=" + estrem0x.toFixed(2) + ","
                                            + "Y1=" + estrem0y.toFixed(2) + ")"
attivata la funzione disegnaTesto()
nell'ambito della quale viene composta la
                                            disegnaTesto() //→
parte di testo testoNn rappresentante la parte
di codice per disegnare ciò che è compreso
nella variabile
           testoXY =P1(x1,y1)
La stessa viene aggiunta alla stringa
                                            testoDisegno = testoDisegno + testoNn
(principale) testoDisegno contenente già il
codice per disegnare all'interno del Tag
</svg></svg> quanto si è preposto
                                            PuntOX = estrem1x * fattscala
                                            PuntOY = estrem1y * fattscala
Analogamente, per il secondo indice (P2)
                                            testoXY = "P2(X2="+estrem1x.toFixed(2)+","+
ripetendo la stessa procedure vista in
                                                    "Y2=" + estrem1y.toFixed(2) + ")"
precedenza, viene definita, con la "chiusura"
                                            disegnaTesto()
del Tag "</svg>, la stringa testoDisegno
                                            testoDisegno = testoDisegno + testoNn
che produrrà il disegno di cui alla figura
                                            testoDisegno = testoDisegno + "</svg>"
seguente.
```



Imposta coordinate "lin Punto 1	ea"[mt]
Xp1	1.00
Yp1	1.00
Punto 2	
Xp2	7.00
Yp2	3.00

IMPOSTAZIONE DATI GEOMETRICI In questa sezione viene approntato un modello che, nell'amabilo di un programma di calcolo, può servire a creare una pagina web (page) da utilizzare per l'impostazione numerica inerente applicazioni tecniche rappresentabili geometricamente in ambiente grafico. Per attuare tale finalità si può strutturare una page come quella che si propone in questo contesto. Essa contiene: -una porzione della page.(0) localizzata nella parte alta, in cui inserirei titolo dell'applicazione: -una motio (1) per l'impostazione dei dati necessari a risolvere la problematica per la quale si intende creare il "software"; -un pannello (3) in cui vengono rappresentati graficamente, per verificare la congruità geometrica, i dati da acquisire;



Fase 11. Visualizzazione ed acquisizione delle coordinate del puntatore del mouse

Avvia codice Fase 11 →

In questa fase, al software, fin qui approntato, vengono aggiunte nuove opzioni che consentono di visualizzare e "catturare" le coordinate dei punti relativi alla posizione del mouse su pannello grafico.

Si possono utilizzare allo scopo gli onmousemove="mouseXY()" eventi onclick="mouseclickXY()" che, inseriti nel tag "SecondoDiv" <div></div>, consentono di acquisire le coordinate dei punti del piano cartesiano con cui, convenzionalmente, è stato definito il utilizzare pannello da per le rappresentazioni grafiche connesse al software che si vuole generare.

<body></body>
<pre></pre> <div <="" id="SecondoDiv" p=""></div>
<pre>style="position: absolute; top:75px; left:325px; width: 1000px; height: 1000px;"</pre>
<pre>onmousemove="mouseXY()" onclick="mouseclickXY()"> </pre>

Questi sono legati rispettivamente alle "function", mouseXY() mouseclickXY()da inserire nella sezione <script></script> , che con il significato dei simboli sono di seguito riportate:

	······································
1) Vengono assegnate alle variabili x, y, con l'evento "event.clientX" "event.clientY", le coordinate corrispondenti alla posizione del mouse nel pannello "SecondoDivtex".	<pre>function mouseXY() { var x = event.clientX var y = event.clientY</pre>
2) Queste, vengono riferire al sistema di riferimento precedentemente definito. Nella fase di trasformazione è indispensabile definire, la precisione con la quale acquisire tali valori.	TrasfPunt0X=(x-326-100)/fattscala TrasfPunt0Y=(0.7*AltezzaFinestra+75-y)/fattscala
3)Ciò, in riferimento alla scala graduata con cui sono rappresentati gli assi cartesiani, significativamente, si attua, considerando solamente una cifra decimale dopo la virgola. Per concretizzare tale intento basta moltiplicare i valori acquisiti per 10 e dopo avere estratto con l'istruzione Math.floor() la parte intera, dividere per lo stesso numero.	TrasfPunt0X = TrasfPunt0X*10; TrasfPunt0X = Math.floor(TrasfPunt0X)/10 TrasfPunt0Y=TrasfPunt0Y*10; TrasfPunt0Y=Math.floor(TrasfPunt0Y)/10
4) Le stesse vengono visualizzate nelle caselle testo Text1 e Text2	<pre>Text1.value = " x=" + TrasfPunt0X.toFixed(2) Text2.value = " y=" + TrasfPunt0Y.toFixed(2) }</pre>
5) Agendo con il mouse nel piano suddetto, le coordinate connesse alla posizione del puntatore al momento del click, vengono "catturate" ed inserite nelle caselle Text3 e Text4, per utilizzarle in altre fasi di programmazione.	<pre>function mouseclickXY() { Text3.value = Text1.value Text4.value = Text2.value } </pre>

Fase 12. Linea definita a due click nel piano del disegno

Avvia codice Fase $12 \rightarrow$

Dopo avere aggiunto le opportune istruzioni nel tag per modificare, con l'aggiunta della casella testo Text9, la parte alta della videata,

Impostazione grafica linea					
della linea nel pannello grafico	"/>				
Impostazione grafica linea	Impostazione grafica linea: Imposta coordinate oppure clicca il primo punto della linea nel pannello grafico				
X= 7.16 Y= 10.44 Xclick= Yclick=	Avvio elaborazione Disegna linea su sfondo Pulisci video				
Text1 Text2 Text3 Text3	t4				

si procede all'utilizzo dei punti acquisiti con i click che si possono operare nel pannello per la rappresentazione grafica.

Allo scopo si opera con la "function mouseclickXY()". nel cui ambito si possono inserire le istruzioni connesse all'uso delle coordinate conseguenti a detto evento. Nel caso in esame il relativo codice è quello che serve per disegnare una linea scaturente da azioni effettuabili dall'esterno: con il mouse.

function mouseclickXY() { 1) Nella casella Text9 vien inserito il messaggio Text9.value= "="Impostazione grafica linea: iniziale relativo all'acquisizione del primo punto e Imposta coordinate oppure clicca il primo Nelle caselle apposite vengono inserite le punto della linea nel pannello grafico" coordinate del punto cliccato, o meglio quelle che Text3.value = Text1.value all'istante del click, seguendo il movimento del Text4.value = Text2.value mouse, vengono editate nelle caselle Text1 e Text2,

Nella stessa "!function", previa definizione iniziale delle variabili var cliccarett1=0, var cliccarett2=0, nella sezione apposita, con tre controlli "if" si verifica in quale fase ci si trova. A seconda del numero de click, ci si può trovare nelle condizioni:

2) legate al click del primo punto	<pre>if(cliccarett1==0){cliccarett1=1; Text5.value= Text3.value; Text6.value = Text4.value; Text9.value= "cliccare il punto 2"; return}</pre>
3) in quelle inerenti il click del secondo punto	<pre>if (cliccarett2==0){cliccarett2=1;Text7.value= Text3.value; Text8.value = Text4.value;}</pre>
4) e in quelle conseguenti al click di tutti i due punti.	<pre>if (cliccarett1 == 1 && cliccarett2 ==1 {Button3_onclick();cliccarett1=0;cliccarett2=0} }</pre>

Tutto ciò, produrrà rispettivamente: (2) il trasferimento dei dati acquisiti con il mouse, nelle caselle Text6 Text7, previste per l'inserimento numerico delle coordinate del primo punto; (3) il trasferimento dei dati acquisiti con il mouse, nelle caselle Text8 Text9, previste per l'inserimento numerico delle coordinate del secondo punto e (4) l'attivazione del pulsante Button3 atto a disegnare la linea ed azzerare le variabili "cliccarett1" e "cliccarett2" che, con questi assegnati valori, ridefiniscono la condizione iniziale per impostare una nuova retta.

Imposta coordinate "linea"[mt]
Punto 1
Xp1
Yp1
Text6
Text7
Punto 2
Xp2
Yp2
Text9