#contentWrapper #fs, #sidebarContent #fs, #contentWrapper div [id * = 'myExtraContent'], #sidebarContent div [id * = 'myExtraContent'] {display: block;}

Kermith's workshop (https://translate.googleusercontent.cc depth=1&hl=en&prev=search&pto=aue&rurl=translate.goog

The other way to see supervision ...

Pocket-Target Project



A few years ago, from 2007 to 2008, I had developed an application under mobile windows in visual-basic. This application allowed the capture of points during my archery training. Unfortunately, Microsoft's strategy and the rapid obsolescence of Windows Phone at the time, decided to abandon the project. The arrival of Livecode in its Community version gave me the desire to

resume my project. At first, I have to appropriate this very literal and quite confusing language at the beginning. The project will be ported to Mac, Linux and Windows initially. Then certainly on Android, I know one that has been waiting for this for a long time. And finally, according to my means, on iMachine because you need a paid version of Livecode.

I will present my very modest first project to you. This will serve as the basis for a more ambitious project which I will enrich in the coming months. I will discuss the simplified entry of an archery practice score sheet.

To start, we will start with a training of two series of six ends of six arrows which corresponds to a Federal or FITA shot. For more information on the practice of archery, see the FFTA website.

Project display

Our program will not yet include a database. We will use a multidimensional table to store the entry of the impacts of arrows.

Arrow impact table

Table name Series Fly Arrow

1st dimension2nd dimension3rd dimension

gArrayImpact1 to 2 1 to 6 1 to 6

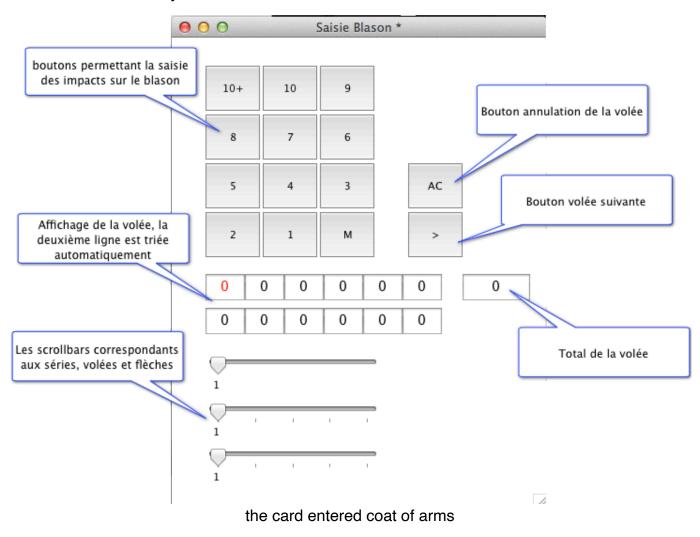
For now, we will opt for a training of two series each comprising six ends of six arrows.

We will use simple button objects for entering impacts. Fields will be used to display the

result.

Volee.livecode mainstack creation

The stack will include a card named SaisieBlason. Create the buttons corresponding to the entry of the impacts on the coat of arms, the label fields corresponding to the values of the impacts of arrows, the three scroolbars representing from top to bottom: the series, the volley and the arrows. The M button represents a missed, an arrow outside the coat of arms. The AC button cancels the entire flight. The next button takes you to the next flight. It will only be visible when the entry is on the last arrow of the coat of arms.



How the program works

Program initialization

On centralisera le code au maximum dans le handler de la carte SaisieBlason, celui-ci

contiendra les déclarations de variables, les événements (command handler), les commandes personnalisées et les fonctions de notre programme. On commencera par l'initialisation du tableau gArrayImpact et des objets de la pile. Pour plus de souplesse, déclarons notre tableau en portée global.

global gArrayImpact

Nous allons utiliser l'événement preopenstack pour faire l'initialisation des variables et objets. A chaque chargement de la pile, on appliquera l'initialisation. Pour initialiser le tableau, on utilisera une commande personnalisée avec le passage du tableau en argument.

on preopenstack
#initialisation du tableau gArrrayImpact
initialisation_tableau gArrayImpact

#initialisation des scrollbars

set the endValue of Scrollbar "ScrollNumFleche" to 6
Set the thumbPosition of Scrollbar "ScrollNumFleche" to 1
set the endValue of Scrollbar "ScrollNumSerie" to 2
Set the thumbPosition of Scrollbar "ScrollNumSerie" to 1
set the endValue of Scrollbar "ScrollNumVolee" to 6
Set the thumbPosition of Scrollbar "ScrollNumVolee" to 1

#initialisation bouton volée suivante

Set the visible of button "BtnSuivant" to false

#remise à zéro de l'affichage Affiche_Volee end preopenstack

Voici la commande personnalisée Affiche_Volee. On peut la comparer à une procédure. On récupère la position de notre volée avec les scrollbars. On identifie la flèche courante pour la saisie en affichant le texte du label field en rouge. Une commande personnalisée trie_volee nous permettra d"'afficher les valeurs de la volée triées en commençant par la valeur la plus forte. Nous terminerons par le calcul de la somme.

on Affiche_Volee # Déclaration des variables local tNumSerie, tNumVolee, tNumFleche, tArrayTri

récupération de la position dans le tableau Série, Volée, Flèche put the thumbPosition of scrollbar "ScrollNumSerie" into tNumSerie put the thumbPosition of scrollbar "ScrollNumVolee" into tNumVolee put the thumbPosition of scrollbar "ScrollNumFleche" into tNumFleche

focus en rouge de la flèche active

set the textColor of Field "LblImpact1" to black set the textColor of Field "LblImpact2" to black set the textColor of Field "LblImpact3" to black set the textColor of Field "LblImpact4" to black set the textColor of Field "LblImpact5" to black set the textColor of Field "LblImpact6" to black switch tNumFleche

case 1

set the textColor of Field "LblImpact1" to red

break

case 2

set the textColor of Field "LblImpact2" to red

break

case 3

set the textColor of Field "LblImpact3" to red

break

case 4

set the textColor of Field "LblImpact4" to red

break

case 5

set the textColor of Field "LblImpact5" to red

break

case 6

set the textColor of Field "LblImpact6" to red

break

end switch

affectation des valeurs des impacts de flèche dans les label field put gArrayImpact[tNumSerie][tNumVolee][1] into Field "LbIImpact1" put gArrayImpact[tNumSerie][tNumVolee][2] into Field "LbIImpact2" put gArrayImpact[tNumSerie][tNumVolee][3] into Field "LbIImpact3"
put gArrayImpact[tNumSerie][tNumVolee][4] into Field "LbIImpact4"
put gArrayImpact[tNumSerie][tNumVolee][5] into Field "LbIImpact5"
put gArrayImpact[tNumSerie][tNumVolee][6] into Field "LbIImpact6"

tri de la volée du plus fort au plus faible (convention des feuilles de marque)

#affectation d'un tableau provisoire

put gArrayImpact[tNumSerie][tNumVolee][1] into tArrayTri[1]
put gArrayImpact[tNumSerie][tNumVolee][2] into tArrayTri[2]
put gArrayImpact[tNumSerie][tNumVolee][3] into tArrayTri[3]
put gArrayImpact[tNumSerie][tNumVolee][4] into tArrayTri[4]
put gArrayImpact[tNumSerie][tNumVolee][5] into tArrayTri[5]
put gArrayImpact[tNumSerie][tNumVolee][6] into tArrayTri[6]

#commande personnalisée de tri

trie_volee tArrayTri

Calcul de la volée put 0 into Field "LblVolee" repeat with count = 1 to 6 add gArrayImpact[tNumSerie][tNumVolee][count] to Field "LblVolee" end repeat

end Affiche_Volee

La procédure de tri trie_volee utilise des particularité du langage Livecode récupéré sur le <u>forum de Livecode (https://translate.googleusercontent.com/translate_c?</u> depth=1&hl=en&prev=search&pto=aue&rurl=translate.google.com&sl=fr&sp=nmt4&u=http://foi

on trie_volee @tArray

#récupération des clés du tableaux dans la variable spéciale it
get the keys of tArray

#tri des clés en fonction des valeurs du tableau
sort lines of it numeric descending by tArray[each]

#conversion en array du resultat
split it by return

#affichage par rapport au tableau trié it

```
put tArray[it[1]] into Field "LblImpactGroupe1"
put tArray[it[2]] into Field "LblImpactGroupe2"
put tArray[it[3]] into Field "LblImpactGroupe3"
put tArray[it[4]] into Field "LblImpactGroupe4"
put tArray[it[5]] into Field "LblImpactGroupe5"
put tArray[it[6]] into Field "LblImpactGroupe6"
end trie_volee
```

La saisie des impacts

Chaque bouton sera pourvu d'un événement dans son handler. Pour les boutons 10 à 1, le chiffre correspond à la valeur de l'impact.

on mouseUp
Saisie_impact 10
end mouseUp

la valeur bouton 10+ est 10 et pour le bouton M la valeur sera 0.

La commande personnalisée Saisie_impact sera stockée dans le handler de la carte.

```
on Saisie_impact NbValue
local tNumSerie, tNumVolee, tNumFleche
put the thumbPosition of scrollbar "ScrollNumSerie" into tNumSerie
put the thumbPosition of scrollbar "ScrollNumVolee" into tNumVolee
put the thumbPosition of scrollbar "ScrollNumFleche" into tNumFleche
put NbValue into gArrayImpact[tNumSerie][tNumVolee][tNumFleche]
if the thumbPosition of Scrollbar "ScrollNumFleche" < 6 then
add 1 to tNumFleche
Set the thumbPosition of Scrollbar "ScrollNumFleche" to tNumFleche
else
if tNumSerie < 2 and tNumVolee < 6 then
Set the visible of button "BtnSuivant" to true
end if
end if
Affiche Volee
```

end Saisie_impact

A chaque nouvelle saisie nous recalculons les impacts de flèches. On détermine si nous nous positionnons sur la dernière flèche l'affichage ou non du bouton volée suivante.

Le déplacement avec les scroolbars

Le programme permet de se positionner automatiquement sur n'importe quel impact du tableau. On utilisera l'événement scrollbarDrag dans chaque handler des scrollbars.

on scrollbarDrag newValueAffiche_Voleeend scrollbarDrag

Nous appellerons la commande personnalisée Affiche_Volee en utilisant les nouvelles valeurs des scrollbars.

L'annulation d'une volée

Pour annuler une volée, nous utiliserons l'handler du bouton AC

on mouseUp
annule_volee
end mouseUp

Nous appellerons une commande personnalisée nommé annule_volee stockée dans le handler de la carte.

on annule_volee
local tNumSerie, tNumVolee, tNumFleche

put the thumbPosition of scrollbar "ScrollNumSerie" into tNumSerie

put the thumbPosition of scrollbar "ScrollNumVolee" into tNumVolee

put 0 into gArrayImpact[tNumSerie][tNumVolee][1]

put 0 into gArrayImpact[tNumSerie][tNumVolee][2]

put 0 into gArrayImpact[tNumSerie][tNumVolee][3]

put 0 into gArrayImpact[tNumSerie][tNumVolee][4]

put 0 into gArrayImpact[tNumSerie][tNumVolee][5]

put 0 into gArrayImpact[tNumSerie][tNumVolee][6]

Set the thumbPosition of Scrollbar "ScrollNumFleche" to 1 Affiche_Volee **end** annule_volee

Volée suivante

Le bouton > permettra de passer à la volée suivante lorsque le bouton est visible.

on mouseUp
Nouvelle_Volee
end mouseUp

Nous appellerons une commande personnalisée nommé Nouvelle_volee stockée dans le handler de la carte.

on Nouvelle_Volee
local tNumSerie, tNumVolee, tNumFleche
put the thumbPosition of scrollbar "ScrollNumSerie" into tNumSerie
put the thumbPosition of scrollbar "ScrollNumVolee" into tNumVolee
put the thumbPosition of scrollbar "ScrollNumFleche" into tNumFleche
if the the thumbPosition of Scrollbar "ScrollNumVolee" < 6 then
add 1 to tNumVolee
put 1 into tNumFleche
else
add 1 to tNumSerie
put 1 into tNumVolee
put 1 into tNumFleche

end if

Set the thumbPosition of Scrollbar "ScrollNumVolee" to tNumVolee **Set** the thumbPosition of Scrollbar "ScrollNumFleche" to tNumFleche **Set** the visible of button "BtnSuivant" to false

Affiche_Volee

Set the thumbPosition of Scrollbar "ScrollNumSerie" to tNumSerie

end Nouvelle_Volee

Conclusion

And here is how to make a small functional application in a very short time. Well, it's true, it took me a few hours of work to learn the syntax and find the sorting solutions, but with experience I should improve.

Here is the <u>source code (../../resources)</u>, you can improve it, I'm sure! feel free to post your improvements or questions. Next step, the graphics ...

