Archived:Como implementar um Mapa Relativo usando GPS em Flash Lite 3.0

Aquivado: Este artigo foi arquivado, pois o conteúdo não é mais considerado relevante para se criar soluções comerciais atuais. Se você achar que este artigo ainda é importante, inclua o template {{ForArchiveReview|*escreva a sua justificativa*}}.

Não recomendamos o desenvolvimento em Flash Lite para aparelhos atuais da Nokia. Todos os artigos relacionados a Flash Lite foram arquivados. A plataforma Nokia Asha e os aparelhos recentes S40 não possuem mais o Flash Lite. O uso de Flash Lite em Symbian é limitado. Por exemplo, informações relativas à plataforma Nokia Belle podem ser encontradas neste link (em Inglês). Informações específicas de Flash Lite para aparelhos S40 ANTIGOS e Symbian podem ser encontradas na Flash Lite Developers Library (em Inglês).

Um Mapa Relativo é conhecido como um mapa em escala de acordo com as distâncias em metros do mundo real. Por exemplo: Você recupera suas coordenadas através de GPS e, logo em seguida, move-se 100 metros a frente e quer saber a distância em pixels na tela do seu celular de acordo com a escala que você escolheu. Melhor ainda seria mostrar onde você está no Mapa Relativo de acordo com sua locomoção. Bem, irei explicar passo-a-passo como fazê-lo desde, implementar as funções geométricas para calcular a relação entre a distância real e a distância relativa até recuperar assincronamente informações de localização através do GPS.

Designe dos Quadros

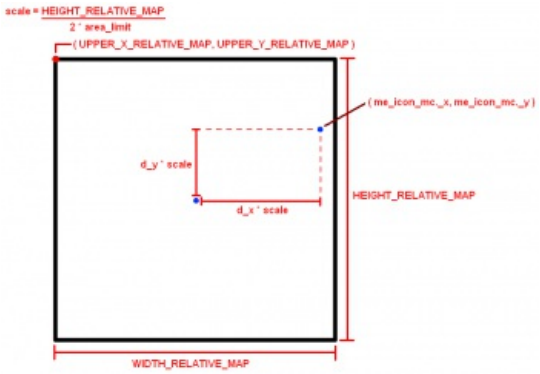
1.1 - Para começar crie um novo projeto móvel e selecionar Flash Lite 3.0 e ActionScript 2.0 como versão do player e versão do ActionScript, respectivamente.

1.2 - Crie 4 (quatro) KeyFrames como mostrados abaixo:

Dica: Para facilitar a criação das telas da aplicação crie o primeiro frame igual ao Frame 1 mostrado acima e, então, adicione os conseguintes.; Com isso você não precisará repetirar código para componentes repetidos em frames diferentes que contém a mesma funcionalidade.

Esquema do Mapa Relativo

Primeiramente vamos entender o que é 'area_limit'; É o valor máximo em metros que você pode se locomover verticalmente ou horizontalmente. Como o nome diz 'scale' representa a escala entre a altura do retângulo desenhado no frame e o dobro de 'area_limit'. Esse valor significa que para cada metro M percorrido no mundo real teremos P pixels percorridos na tela do celular. Então, achadas as distâncias 'd_x' e 'd_y' em metros multiplicaremos cada uma delas por 'scale' para obter as distâncias horizontal e vertical em pixels. Para maiores detalhes ver o esquema mostrado abaixo:



http://developer.nokia.com/community/wiki/Archived:Como_implementar_um_Mapa_Relativo_usando_GPS_em_Flash_Lite_3.0 (C) Copyright Nokia 2014. All rights reserved.

Explicação do Código

Aqui irei discorrer um pouco sobre os códigos usados em cada frame para a implementação do Mapa Relativo.

3.1 - No primeiro frame devemos desabilitar os botões de teclado físico (pois a versão mostrada aqui é para touch screen) e fazer com que a aplicação execute em modo de tela inteira. Criar as variáveis que representam latitude e longitude do ponto central e o limite de área em metros do mapa. Os limites podem ser quaisquer valores; caso queira mudá-los basta modificar os labels do 'ComboBox'. Por último codificar as ações dos botões, tudo como mostrado no código abaixo:

```
fscommand2("DisableKeypadCompatibilityMode"); // Disable keypad
fscommand2("Fullscreen", true); // Set FullScreen mode
fscommand2("SetQuality", "high"); // Set high quality

var UPPER_X_RELATIVE_MAP:Number = 10;
var UPPER_Y_RELATIVE_MAP:Number = 130;
var HEIGHT_RELATIVE_MAP:Number = 340;
var WIDTH_RELATIVE_MAP:Number = 340;
var ICON_HEIGHT:Number = 6;
var ICON_WIDTH:Number = 6;

var longitude_center:Number = new Number();
var latitude_center:Number = new Number();
var area_limits:Number = new Number();

get_center_location_btn.onRelease = function() {
 area_limits = int(area_limit.text.split(" ")[0]);
 gotoAndStop(2);
};

exit_btn.onRelease = function() {
 fscommand2("Quit");
};

stop();
```

3.2 - No frame dois (2) instanciamos um objeto Service o qual será responsável por recuperar os dados de localização de GPS na plataforma S60, porém antes é necessário importar a biblioteca 'com.nokia.lib.Service'. Feito isso codifique a ação do botão 'cancel_gps' para cancelar o acesso ao GPS naquele exato momento.

Tip: Para melhor controle de fluxo de telas use o método GetLocation assíncrono.

```
import com.nokia.lib.Service;

var location = new Service("Service.Location", "ILocation");
var inParams = {LocationInformationClass:"GenericLocationInfo"};

location.GetLocation(inParams,onReceive);

function onReceive(transactionID:Number, eventID:String, outParam:Object) {
 if (outParam.ErrorCode == 0) {
  var outList = outParam.ReturnValue;
  longitude_center = outList.Longitude;// Contains longitudinal data
  latitude_center = outList.Latitude;// Contains latitudinal data
  gotoAndStop(4);
 } else {
  var errorId = outParam.ErrorCode;
```

 (C) Copyright Nokia 2014. All rights reserved.

```
  gotoAndStop(3);
 }
}

cancel_gps.onRelease = function() {
 var inParamsCancel = {CancelRequestType:"GetLocCancel"};
 var outParamsCancel = location.CancelNotification(inParamsCancel);
 var errorIdCancel = outParamsCancel.ErrorCode;
 gotoAndStop(1);
};
```

3.3 - No terceiro frame temos a tela mostrada quando ocorre qualquer erro no acesso as informações de localização do GPS.

Observação: A tela escura mostrada a frente da tela é um botão com nome de instância 'screen_unlocker_btn'. Quando ocorrer erro basta tocar na tela que você retornará a primeira tela (frame 1).

```
screen_unlocker_btn.onRelease = function() {
 gotoAndStop(1);
};
```

3.4 - Para finalizar temos o código para o quarto frame, onde teremos que usar a função 'Trace' para recuperarmos a localização de GPS a cada tempo predefinido, no caso do código abaixo foi atribuido o valor 1000000 que equivale a 1 segundo. A cada execução da função 'onNotify' atribuiremos a posição do MovieClip 'me_icon_mc' que representa sua posição no mapa relativo e ao campo de texo dinâmico 'distance_from_start' atribuiremos a distância entre você e o ponto central. Importante, caso você saia dos limites do mapa seu ícone ficará representado na borda do mapa relativo e uma mensagem será mostrado no campo 'alert'.

```
import com.nokia.lib.Service;

var location = new Service("Service.Location", "ILocation");
var updateOptions = {UpdateInterval:1000000};
var inParamsTrace = {LocationInformationClass:"GenericLocationInfo",
Updateoptions:updateOptions};

location.Trace(inParamsTrace,onNotify);

function onNotify(transactionID:Number, eventID:String, outParam:Object) {
 if (outParam.ErrorCode == 0) {
  var outList = outParam.ReturnValue;
  longitude = outList.Longitude;// Contains longitudinal data
  latitude = outList.Latitude;// Contains latitudinal data

  setMeIconPosition(latitude,longitude);

  // Calculates the distance between center and the treasure
  var sourceDistance = {Longitude:longitude, Latitude:latitude, Altitude:0.0};
  var destinationDistance = {Longitude:longitude_center, Latitude:latitude_center,
Altitude:0.0};
  var inParamsDistance = {MathRequest:"FindDistance",
DistanceParamSource:sourceDistance, DistanceParamDestination:destinationDistance};
  var outParamsDistance = location.Calculate(inParamsDistance);

  if (outParamsDistance.ErrorCode == 0) {
   var distance = outParamsDistance.ReturnValue;
   var distStr = distance.toString()+"00";
   distance_from_start.text = distStr.split(".")[0]+"."+substring(distStr.split(".")[1],
```

http://developer.nokia.com/community/wiki/Archived:Como_implementar_um_Mapa_Relativo_usando_GPS_em_Flash_Lite_3.0 (C) Copyright Nokia 2014. All rights reserved.

```
0, 2);
  } else {
   var errorId = outParamsDistance.ErrorCode;
  }

 } else {
  var errorId = outParam.ErrorCode;
 }
}

function setMeIconPosition(latitude:Number, longitude:Number):Void {

 var d_y = location.Calculate({MathRequest:"FindDistance",
DistanceParamSource:{Longitude:longitude_center, Latitude:latitude_center,
Altitude:0.0}, DistanceParamDestination:{Longitude:longitude_center, Latitude:latitude,
Altitude:0.0}}).ReturnValue;
 var d_x = location.Calculate({MathRequest:"FindDistance",
DistanceParamSource:{Longitude:longitude_center, Latitude:latitude_center,
Altitude:0.0}, DistanceParamDestination:{Longitude:longitude, Latitude:latitude_center,
Altitude:0.0}}).ReturnValue;
 var scale = HEIGHT_RELATIVE_MAP/(2*area_limits);// or WIDTH

 if (latitude_center<latitude) {
  if (d_y>area_limits) {
   vertical_value = UPPER_Y_RELATIVE_MAP-(ICON_HEIGHT/2);
  } else {
   vertical_value =
UPPER_Y_RELATIVE_MAP+(HEIGHT_RELATIVE_MAP/2)-(scale*d_y)-(ICON_HEIGHT/2);
  }
 } else {
  if (d_y>area_limits) {
   vertical_value = UPPER_Y_RELATIVE_MAP+HEIGHT_RELATIVE_MAP-(ICON_HEIGHT/2);
  } else {
   vertical_value =
UPPER_Y_RELATIVE_MAP+(HEIGHT_RELATIVE_MAP/2)+(scale*d_y)-(ICON_HEIGHT/2);
  }
 }

 if (longitude_center<longitude) {
  if (d_x>area_limits) {
   horizontal_value = UPPER_X_RELATIVE_MAP+WIDTH_RELATIVE_MAP-(ICON_WIDTH/2);
  } else {
   horizontal_value =
UPPER_X_RELATIVE_MAP+(WIDTH_RELATIVE_MAP/2)+(scale*d_x)-(ICON_WIDTH/2);
  }
 } else {
  if (d_x>area_limits) {
   horizontal_value = UPPER_X_RELATIVE_MAP-(ICON_WIDTH/2);
  } else {
   horizontal_value =
UPPER_X_RELATIVE_MAP+(WIDTH_RELATIVE_MAP/2)-(scale*d_x)-(ICON_WIDTH/2);
  }
 }

 me_icon_mc._y = vertical_value;
 me_icon_mc._x = horizontal_value;
```

http://developer.nokia.com/community/wiki/Archived:Como_implementar_um_Mapa_Relativo_usando_GPS_em_Flash_Lite_3.0 (C) Copyright Nokia 2014. All rights reserved.

```
 if (d_y>area_limits || d_x>area_limits) {
  out_of_bounds = true;
  alert.text = "Out of bounds!";
 }
}
```

Código-Fonte para Download

- File:CristovaoDutra RelativeMap.zip

http://developer.nokia.com/community/wiki/Archived:Como_implementar_um_Mapa_Relativo_usando_GPS_em_Flash_Lite_3.0 (C) Copyright Nokia 2014. All rights reserved.