



## Contents

- [1 Nokia series 40 and S60 devices with Flash Lite Game example](#)
- [2 Device issues](#)
- [3 Flash Lite issues / tips](#)
- [4 Flash Lite 1.1 Game with simulated Array](#)
- [5 Code](#)
- [6 Download](#)
- [7 Author](#)

## Nokia series 40 and S60 devices with Flash Lite Game example

So you have decided that you would like to make a Flash Lite game for Nokia devices.

Before you start coding the game you should be aware of a few things about Mobile Devices and Flash Lite.

### Device issues

Developing for a device means, we have a few limitations that we must keep in mind, when develop Flash Lite games or content.

- The screen resolution on a device tends to be much lower then that of a desktop computer. The mobile screen size is 128X160, 176X208, 240X320, 320X240 pixels in resolution. Compare to the desktop resolution of 800X600.
- The CPU tends to be slower on a device.
- The input options are different on a mobile devices and a desktop.
- The amount of RAM available to a mobile device compared to a desktop computer can differ very much. Making games that 2MB of Bitmap, Vector and sound run on desktop computer but not feasible on a mobile devices.
- Two key press at a same time not work. For example player run and jump same time. No mouse click and drag navigation.

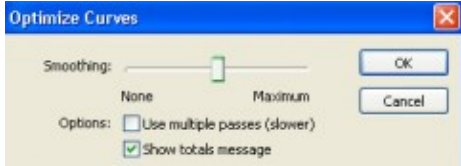
### Flash Lite issues / tips

- Use vector graphics rather than bitmap graphics. Flash is vector based, so the final Flash file will have small size and would also automatically resize for large screen.
- It is also good idea to use bitmap images at sometimes. When the graphic to be drawn is of small size and would not be rescaled, its wise to use Bitmaps as it would eliminate the need for computation.
- Use vector graphics, but not use lots of curves, add less anchor points and do not use gradients .
- Use device fonts and use pixel fonts For example hooge 05\_53



## Device fonts

- Fonts are distorted when flash Lite file automatically scaled and small make them fonts blur or not readable.
- Optimize the graphics in such a way that clarity and presentation are not reduced but the computation involved is reduced.



## Optimize Curves

- Use variable and predefine values, not use too much hard-code.
- To optimize the graphics use:

```
fscommand2("SetQuality", quality)
```

- Reuse your functions

**One of the most important thing in game development is - How well Flash Lite game actually play on mobile devices.** . This is determined by -

- **How Responsive it is**
- **How smooth it is**

Always test the game on your handset as there are certain parts of the script that can be tested only on a physical device. Always remember that the emulator or Adobe Device Central is not completely efficient and is an approximation of the performance.

## Flash Lite 1.1 Game with simulated Array

When you are developing Flash Lite 1.1 game. you may think to use the array. But in flash Lite 1.1, you don't use the array. what to do?

- The solution is to develop **simulate array** by creating a set of variables or instance name.
- The eval function was used to simulate array.
- You can also use eval function to dynamically set and retrieve the value of a variable or instance name.

For example: A player eat bugs and add point with use of simulate array and predefine values.

## How\_to\_develop\_Flash\_Lite\_games\_for\_Nokia\_series\_40\_and\_S60\_devices

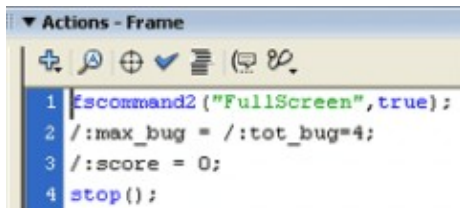
- There is also an alternative approach for [Simulating Arrays in better way in Flash Lite 1.1](#)

- First you have an idea of game.
- you want to create a nice background for game.
- Define a 12X12 grid.
- player size 12X12 for four sides. **For Up, Down, Left, Right positions .**
- Make a instance name **player** set frame label up, Left, Right, Down. Stop them four frames.
- Create a bug and give a instance name **b1, b2, b3, b4**
- Make sure that both player and bug have registration point in left top corner.
- Now lets declare variables in frame.
- Maximum bug=total bugs.

```
/:max_bug=tot_bug=4
```

- Text variable for score.

```
/:score=0
```



### Actions Frame

Define event handler

- First move the player around screen.

Move a player in four direction. Move X and Y position 12 pixel.

```
tellTarget ("/player") {
gotoAndStop(2);
if (_root.player._x>22) {
//_x and _y position and speed
12;          posX = _x-
_y;          posY =
_x = posX;
tellTarget ("/player") {
gotoAndStop(2);
}
```

- Predefine values of bug x and y position on screen.

```
▼ Actions - Frame
1 bug10 = 24;
2 bug11 = 60;
3 //
4 bug20 = 108;
5 bug21 = 84;
6 //
7 bug30 = 36;
8 bug31 = 120;
9 //
10 bug40 = 72;
11 bug41 = 132;
```



## Actions Frame

- Check bugs on screen.

Use For loop to check the bugs on screen. create simulated array to check the bugs and collision with player x and y position. If collision happened add score and decreased one bug on screen make bug visible false on screen. if player x, y coordinate and bug X, Y coordinate collide.

- Check if bug maximum equal to 0 then go to completed frame.

```
if (_root.max_bug<=0) {
tellTarget ("_level0") {
gotoAndPlay("completed");
}
}
```



## Nokia 3110

## Code

```
//Left Key
on (keyPress "<Left>") {
```

Code

## How\_to\_develop\_Flash\_Lite\_games\_for\_Nokia\_series\_40\_and\_S60\_devices

```
tellTarget ("/player") {
gotoAndStop(2);
if (_root.player._x>22) {
//_x and _y position and speed
        12;          posX = _x-
        _y;          posY =
_x = posX;
tellTarget ("/player") {
gotoAndStop(2);
}
//Checking for bug
for (i=1; i<=/:tot_bug; i++) {
//Simulated Array
if (eval("/:bug" add i add "0") == posX) {
if (eval("/:bug" add i add "1") == posY) {
// Add score
_root.score += 10;
_root.max_bug -= 1;
tellTarget ("/b" add i) {
_visible = false;
}
//check if bug max equal to 0 then go to completed frame
if (_root.max_bug<=0) {
tellTarget ("_level0") {
gotoAndPlay("completed");
}
}
break; // this will terminate/exit the loop

}
}
}
}
}
}
//Right key
on (keyPress "<Right>") {
tellTarget ("/player") {
gotoAndStop(3);
if (_root.player._x<140) {
//_x and _y position and speed
        _x+12;      posX =
        _y;          posY =
_x = posX;
tellTarget ("/player") {
gotoAndStop(3);
}
//Checking for bug
for (i=1; i<=/:tot_bug; i++) {
//Simulated Array
if (eval("/:bug" add i add "0") == posX) {
if (eval("/:bug" add i add "1") == posY) {
//Add score
_root.score += 10;
_root.max_bug -= 1;
trace("b" add i);
tellTarget ("/b" add i) {
_visible = false;
}
}
//check if bug max equal to 0 then go to completed frame
if (_root.max_bug<=0) {
tellTarget ("_level0") {
```

## How\_to\_develop\_Flash\_Lite\_games\_for\_Nokia\_series\_40\_and\_S60\_devices

```
gotoAndStop("completed");
}
}
break;// this will terminate/exit the loop

}
}
}
}
}

///// Up key
on (keyPress "<Up>") {
//Path
tellTarget ("/player") {
gotoAndStop(1);
if (_root.player._y>24) {
//_x and _y position and speed
        _x;          posX =
        12;          posY = _y-
_y = posY;
tellTarget ("/player") {
gotoAndStop(1);
}
//Checking for bug
for (i=1; i<=/:tot_bug; i++) {
//Simulated Array
if (eval("/:bug" add i add "0") == posX) {
if (eval("/:bug" add i add "1") == posY) {
//Add score
_root.score += 10;
_root.max_bug -= 1;
trace("b" add i);
tellTarget ("/b" add i) {
_visible = false;
}
//check if bug max equal to 0 then go to completed frame
if (_root.max_bug<=0) {
tellTarget ("_level0") {
gotoAndPlay("completed");
}
}
}
break;// this will terminate/exit the loop

}
}
}
}
}

/////Down Key
on (keyPress "<Down>") {
//Path
tellTarget ("/player") {
gotoAndStop(4);
if (_root.player._y<176) {
//_x and _y position and speed
        _x;          posX =
        _y+12;       posY =
_y = posY;
tellTarget ("/player") {
```

## How\_to\_develop\_Flash\_Lite\_games\_for\_Nokia\_series\_40\_and\_S60\_devices

```
gotoAndStop(4);
}
//Checking for bug
for (i=1; i<=/:tot_bug; i++) {
//Simulated Array
if (eval("/:bug" add i add "0") == posX) {
if (eval("/:bug" add i add "1") == posY) {
_root.score += 10;
_root.max_bug -= 1;
trace("b" add i);
tellTarget ("/b" add i) {
_visible = false;
}
//check if bug max equal to 0 then go to completed frame
if (_root.max_bug<=0) {
tellTarget ("_level0") {
gotoAndPlay("completed");
}
}
}
break;// this will terminate/exit the loop

}
}
}
}
}
}
```



Nokia 95

## Download

- You can download an example with source code here: [Flashlite game](#)

## Author

--Narender Singh Raul