# Coding Convention

v0.0.5

Last updated: 21st February 2008

## 1. Variable naming

| | |
|---|---|
| m_ | : class member |
| s_ | : static |
| b | : byte |
| c | : char |
| s | : short |
| i | : int |
| l | : long |
| is/was/has/... | : boolean (because it's a yes-no question) |
| a<x> | : array of <x>. For example: int [] m_aiCards (array of int) |
| str | : String |
| spr | : ASprite object |
| ani | : CAnim object |
| img | : Image |
| gra | : Graphics |
| CAPITALIZE | : constant |

Of course you don't need to always follow those prefix conventions. In some cases where a variable's name is meaningful enough, there's no need to add a redundant prefix for it. For example:

```
static Graphics   s_currentGraphics;
```

Capitalize all letters of constant's name. Example:

```
final static int  CANVAS_WIDTH      = 176;
```

*** A few examples:**

Member variables

```
private static int     s_iCurrentScore;
public boolean         m_wasKeyPressed;
public static Image    s_imgBackBuffer = Image.createImage(...);
public Graphics        m_graBackBuffer =
m_imgBackBuffer.getGraphics();
```

Local variables:

```
String strItemPrice;
long lCurrentTime; // in mili seconds
byte bBitMask;
```

### * Preprocessor macros:

- Use uppercase for preprocessor directives. For example:

```
    #define     USE_SOUND         1
    #define     USE_SIMPLE_AI     0
    #define     KEEP_BACKGROUND   0
```

- Whenever possible, use defines that mean something. For example:

```
good:
         #if USE_CUSTOM_UTF8_DECODING

NOT good:
         #ifdef SONYERICSSON_K800 || SONYERICSSON_K500
NOT good either:
         #ifdef FIX_BUG_FREEZE_WITH_UTF8_STRING
```

- Use positive defines. For example:

```
good:
         #if KEEP_BACKGROUND
   or
         #if !KEEP_BACKGROUND

NOT good:
         #if DONT_KEEP_BACKGROUND
```

## 2. Method naming

Capitalize the first letter of each word. For example:

```
UpdateCarPosition()
DrawPlayerHand()
Keypad_Update()
…
```

## 3. Comments

Use Doxygen documenting style.
(see *GameloftCodingGuideline107.pdf*, part 10)

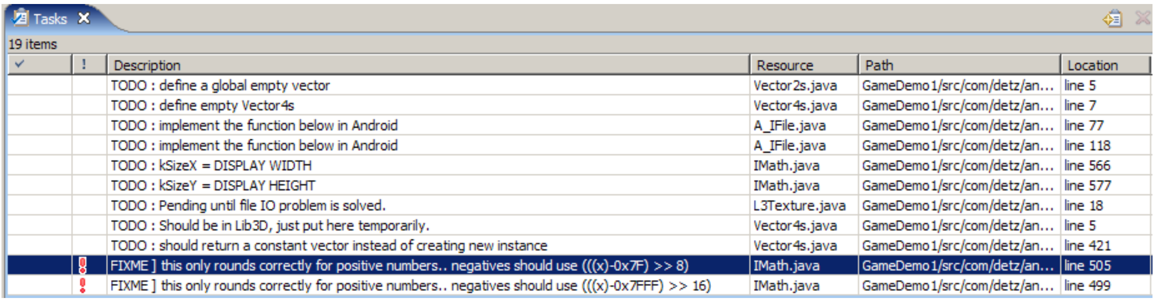
### * Tags:

Certain tags are used in comments to assist in indexing common issues. Such tags can be searched with common programming tools, such as the UNIX *grep* utility or can be filtered by some IDE such as *Eclipse*.

- **TODO** is used to indicate planned enhancements.
- **FIXME** is used to mark potential problematic code that requires special attention and/or review.
- **NOTE** is used to document inner workings of code and indicate potential pitfalls.

- **BUGFIX #*id_number*** is used to indicate that the piece of code is added/modified in order to fix some bug.

Example of Eclipse's tasks Window:



19 items

| ✓ | ! | Description | Resource | Path | Location |
|---|---|---|---|---|---|
| | | TODO : define a global empty vector | Vector2s.java | GameDemo1/src/com/detz/an... | line 5 |
| | | TODO : define empty Vector4s | Vector4s.java | GameDemo1/src/com/detz/an... | line 7 |
| | | TODO : implement the function below in Android | A_IFile.java | GameDemo1/src/com/detz/an... | line 77 |
| | | TODO : implement the function below in Android | A_IFile.java | GameDemo1/src/com/detz/an... | line 118 |
| | | TODO : kSizeX = DISPLAY WIDTH | IMath.java | GameDemo1/src/com/detz/an... | line 566 |
| | | TODO : kSizeY = DISPLAY HEIGHT | IMath.java | GameDemo1/src/com/detz/an... | line 577 |
| | | TODO : Pending until file IO problem is solved. | L3Texture.java | GameDemo1/src/com/detz/an... | line 18 |
| | | TODO : Should be in Lib3D, just put here temporarily. | Vector4s.java | GameDemo1/src/com/detz/an... | line 5 |
| | | TODO : should return a constant vector instead of creating new instance | Vector4s.java | GameDemo1/src/com/detz/an... | line 421 |
| | ! | FIXME ] this only rounds correctly for positive numbers.. negatives should use (((x)-0x7F) >> 8) | IMath.java | GameDemo1/src/com/detz/an... | line 505 |
| | ! | FIXME ] this only rounds correctly for positive numbers.. negatives should use (((x)-0x7FFF) >> 16) | IMath.java | GameDemo1/src/com/detz/an... | line 499 |

Example 1:

```
// [phong] TODO: implement binary search for better optimization
public int findElement(int[] array, int element)
{
	for (int i = 0; i < array.length; i++)
	{
		if (array[i] == element)
		{
			return i;
		}
	}

	return -1;
}
```

Example 2:

```
public static void UpdatePlayerPos(int dx, int dy)
{
	m_player.m_posX += dx;
	m_player.m_posY += dy;

	// [phong] FIXME: this only works on 240x320 design!!
	if (m_player.m_posY > 212)
	{
		m_isGameOver = true;
	}
	////
}
```

Example 3:

```
// NOTE: Make sure this is called before using SetStringParamStringID
void ResetStringParamString()
{
	for (int i = 0; i < FONT_MAX_ESCAPE_STRINGS; i++)
	{
		m_strValues[i] = null;
```

```
      }
}
```

## Example 4:

```
// [phong] BUGFIX #469756 (Day 3: Random freeze when the cat breaks
// the coffee cup)
if (tile >= m_tileMapLayer[0].length)
{
      return;
}
////
```