Floating Point Guidelines

Guidelines

Guideline 1:

> «Do **not** test two floating point numbers for **equality**, if at least one of them was rounded before.»

Guideline 1 – Example

Guideline 1:

> «Do **not** test two floating point numbers for **equality**, if at least one of them was rounded before.»

This is `false`

Example:

```
float a = 0.1f;
if (10*a == 1.0f)
    std::cout << "no output\n";
```

Guideline 1 – Example

Guideline 1:

> «Do **not** test two floating point numbers for **equality**, if at least one of them was rounded before.»

Example:

```
float a = 0.1f;
if (10*a == 1.0f)
    std::cout << "no output\n";
```

This is `false`

Problem:

`0.1f` not representable

Guideline 1 – Example

Guideline 1:

> «Do **not** test two floating point numbers for **equality**, if at least one of them was rounded before.»

This is `false`

Example:

```
float a = 0.1f;
if (10*a == 1.0f)
  std::cout << "no output\n";
```

Problem:

`0.1f` not representable

$$\underbrace{1.10011001100110011001100}_{\text{24bit}}11\ldots \cdot 2^{-4}$$

$$0.1 = 1.10011001100110011001100\,11\ldots \cdot 2^{-4}$$

$$\text{(rounding)} \rightarrow 0.10000000149\ldots = 1.10011001100110011001101 \cdot 2^{-4}$$

Guidelines

Guideline 1:

«Do **not** test two floating point numbers for **equality**, if at least one of them was rounded before.»

Guideline 2:

«**Avoid** the **addition** of numbers of extremely **different sizes**!»

Guideline 2 – Example

Guideline 2:

«**Avoid** the **addition** of numbers of extremely **different sizes**!»

Example:

```
float a = 67108864.0f + 1.0f;

if (a > 67108864.0f)
    std::cout << "This is not output ... \n";
```

Guideline 2 – Example

Guideline 2:

«**Avoid** the **addition** of numbers of extremely **different sizes**!»

Problem:

Significand too short

Example:

```
float a = 67108864.0f + 1.0f;

if (a > 67108864.0f)
    std::cout << "This is not output ... \n";
```

Guideline 2 – Example

Guideline 2:

«**Avoid** the **addition** of numbers of extremely **different sizes**!»

Problem:

Significand too short

Example:

```
float a = 67108864.0f + 1.0f;

if (a > 67108864.0f)
    std::cout << "This is not output ... \n";
```

$$
\begin{aligned}
67108864 &= \overbrace{1.00000000000000000000000}^{\text{24bit}} \cdot 2^{26} \\
+1 &= 0.00000000000000000000000001 \cdot 2^{26} \\
\hline
67108865 &= 1.00000000000000000000000001 \cdot 2^{26}
\end{aligned}
$$

Guideline 2 – Example

Guideline 2:

«**Avoid** the **addition** of numbers of extremely **different sizes**!»

Problem:

Significand too short

Example:

```
float a = 67108864.0f + 1.0f;

if (a > 67108864.0f)
    std::cout << "This is not output ... \n";
```

24bit

$$
\begin{aligned}
67108864 &= \overbrace{1.00000000000000000000000}\ \cdot 2^{26} \\
+1 &= 0.00000000000000000000000001 \cdot 2^{26} \\
\hline
67108865 &= 1.00000000000000000000000001 \cdot 2^{26} \\
\text{(rounding)} \rightarrow 67108864 &= 1.00000000000000000000000\ \cdot 2^{26}
\end{aligned}
$$

Guidelines

Guideline 1:

«Do **not** test two floating point numbers for **equality**, if at least one of them was rounded before.»

Guideline 2:

«**Avoid** the **addition** of numbers of extremely **different sizes**!»

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes**!»

Guideline 3 – Example

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes**!»

Example:

```
float volume_exact  = 35.828125f;
float volume_approx = 35.328125f;

float diff =   volume_exact
             - volume_approx;
```

Due to rounding errors

Guideline 3 – Example

Guideline 3:

«**Avoid** the **subtraction** of numbers of **similar sizes**!»

Example:

```
float volume_exact  = 35.828125f;
float volume_approx = 35.328125f;

float diff =   volume_exact
             - volume_approx;
```

Due to rounding errors

`diff` **absolutely not** `0`

Danger:

Affects later computations.