Sprite tester

1. Content

2. SpriteTester

The project SpriteTester gives you the ability to test your sprites. You just need to adapt some initializations as mentioned in the TODO list of Game.cpp and copy your sprite sheets in the Resources folder.

The Tibo sprite has been used as an example. Some of these data members are explained further in this document.

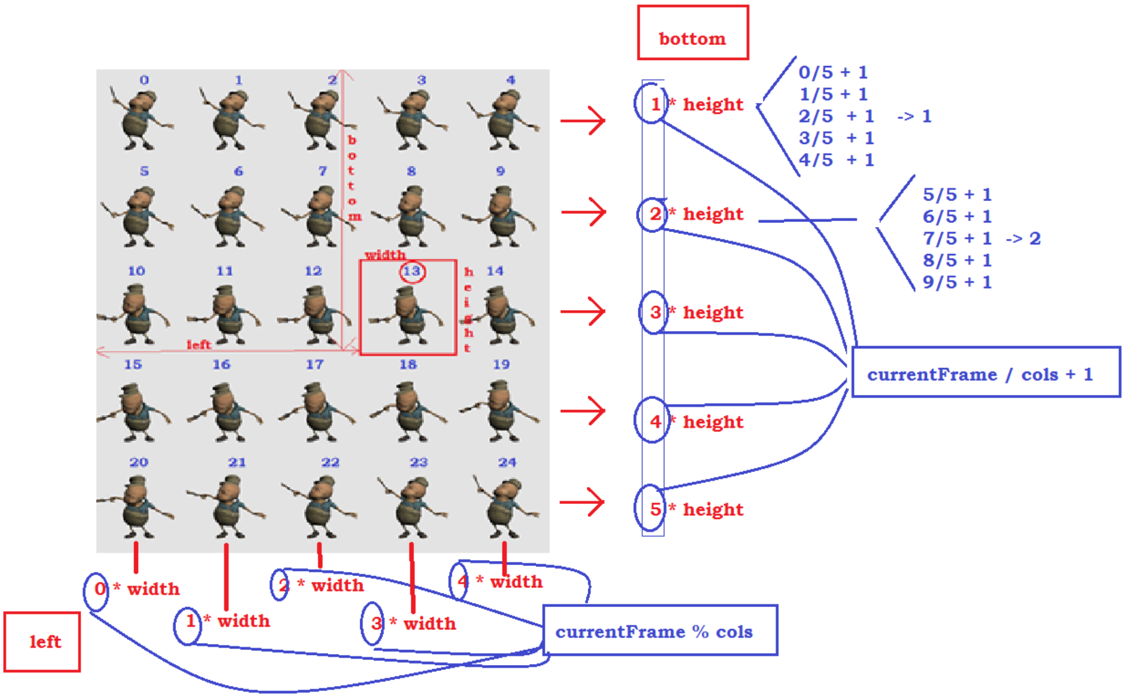
2.1. Controls

The **framerate** and **scale** are adjustable at runtime with the **up/down arrow** and **Page up/down** keys.

With the **h-key** you enable/disable showing this information on the console.

2.2. Brush up on sprites

2.2.1. Source rectangle



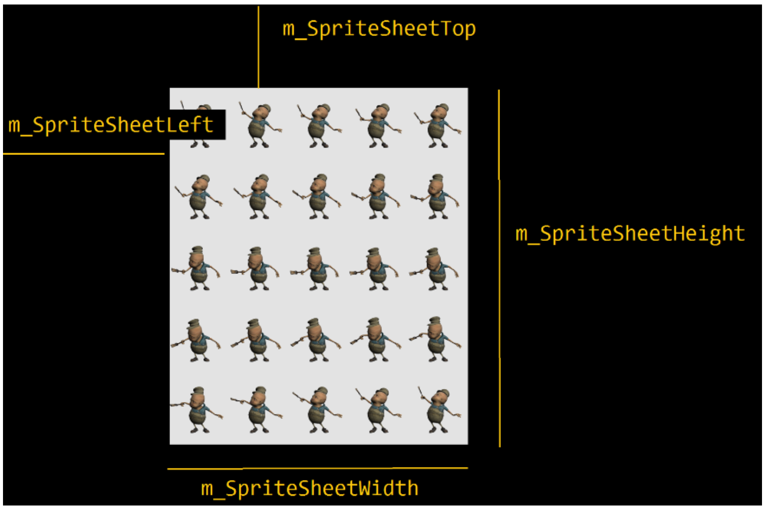
We can draw only a (rectangular) part of a Texture by specifying the srcRect parameter of the Texture::Draw method. The Tibo image above shows how we can determine this source rectangle for a given frame.

- The **frame numbers** are indicated in **blue**, we start counting at the top left corner.
- The **clipping area** – for frame number 13 - is indicated by the **red** rectangle.
- At the right side is indicated how we deduce the **bottom** value from the current frame number.
- At the bottom is indicated how we deduce the **left** value from the current frame number.
- The **width** is the total width divided by the number of frame columns.
- The **height** is the total height divided by the number of frame rows.

2.2.2. Source rectangle in case of borders

Above are the formulas when there are no other animations or no borders in the sprite sheet. However this is not always the case. That's why we introduced the 4 data members **m_SpriteSheetLeft**, **m_SpriteSheetTop**, **m_SpriteSheetWidth**, **m_SpriteSheetHeight** in the SpriteTester. They contain the location, width and height of the frames under test.

The picture below shows a texture containing the same Tibo frames but with black borders around them. It clarifies the meaning of these 4 data members.



This leads to the following formulas for the source rectangle

width	m_SpriteSheetWidth / m_Cols
height	m_SpriteSheetHeight / m_Rows
left	m_SpriteSheetLeft + m_CurFrame % m_Cols * m_SrcRect.width
bottom	m_SpriteSheetTop + (m_CurFrame / m_Cols + 1)* m_SrcRect.height

When the spritesheet has no borders then:

- m_SpriteSheetLeft and m_SpriteSheetTop are both 0 and
- m_SpriteSheetWidth contains the width of the texture
- m_SpriteSheetHeight contains the height of the texture