IN628 2020 Practical 11 – Sprite/NPC Scrolling Tile Map 2

Introduction

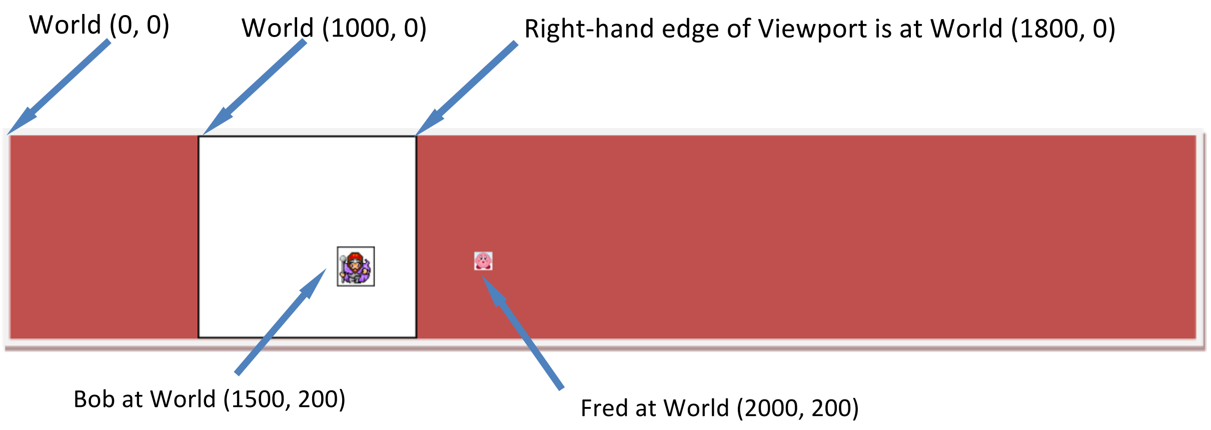
In a scrolling game, the game world is much bigger than the computer screen. We have termed the visible portion of the world the 'viewport'. In our finished game, the player character (i.e. the character controlled by the user) will be fixed in the centre of the screen and the viewport will position itself around him (or her, or it).

Non-player sprites will be distributed in the world per the design of the game. These sprites' xPos and yPos properties represent their world coordinates. The sprites only appear on the screen when the viewport contains the portion of the world at which they are located (i.e. when the player character is close to them). And, of course, the location they are drawn on the screen will not be equal to their location in the world. Therefore, at each game cycle, we must determine for each NPC sprite:

1. If the sprite is contained in the viewport area (and is thus visible)
2. Where on ***the screen*** the sprite should be drawn.

Determining NPC Location

Assume, for example, that our world is 8,000 pixels wide and 600 pixels high. Assume that the viewport is 800x600, and its upper left corner is currently at (1000, 0) in the world. Assume that NPC Bob is at (1500, 200) in the world (that is, Bob's xPos is 1500 and his yPos is 200). NPC Fred is at (2000, 200). This situation is illustrated below (this image is not to scale).


World (0, 0)
World (1000, 0)
Right-hand edge of Viewport is at World (1800, 0)
Bob at World (1500, 200)
Fred at World (2000, 200)

Bob is in the viewport area, and should therefore be drawn to the screen. Fred is not in the viewport area and should not be drawn to the screen.

Although Bob's World position is (1500, 200), that is not the location that should be passed into the DrawImage statement in his draw method. Since the viewport represents the visible screen area, Bob needs to be drawn ***at his position in the viewport, not his position in the world***. Since the viewport's x-position in the world is 1000 and Bob's is 1500, Bob is 500 pixels to the right of the viewport edge. Similarly, he is 200 pixels down from the top of the viewport. Therefore, the coordinates to pass into the DrawImage command should be 500 and 200.

In pseudocode, we say:

```
1. sprite viewport location X = sprite world location X – viewport world location X
2. sprite viewport location Y = sprite world location Y – viewport world location Y
3. if the point (sprite viewport location X, sprite viewport location Y) is in the viewport
   area, draw the sprite at that location.
```

Managing NPC Drawing

By now, your Sprites should be managed by a linked list. It is therefore sensible to make these computations the job of the SpriteList class. Add a new method to your SpriteList class that takes the world location and dimensions of the viewport as input parameters. In pseudocode:

SpriteList::DrawSpritesInViewport(in vWorldX, int vWorldY, int vPixelWidth, int vPixelHeight)

```
1. Traverse the list.
2. For each sprite
    a. Compute his viewport position by subtracting vWorldX and vWorldY from
       the sprite's xPos and yPos values respectively.
    b. If that location is in the viewport area, draw the sprite at that location.
```

In this code you will need to use the Sprite::ForcedDraw(int x, int y) method that you wrote last week (or your overloaded Draw method, whichever applies) to force the Sprite to draw at the computed location, rather than at its world coordinates xPos and yPos.

Testing Your Code

Create a large tile map and a smaller scrolling viewport. Randomly scatter a lot of Sprites around in the world. As you move the viewport (either manually, or by moving the Player and centering the viewport around it) the Sprites should appear as the viewport passes over them, and then disappear as it moves away from them. If you move the viewport back to them, they should reappear. When they are in the viewport area, it should look as though they are maintaining their position in the world. That is, the sprites should stay in their world locations and be drawn at the correct screen position when the viewport is displaying the part of the world they are in.