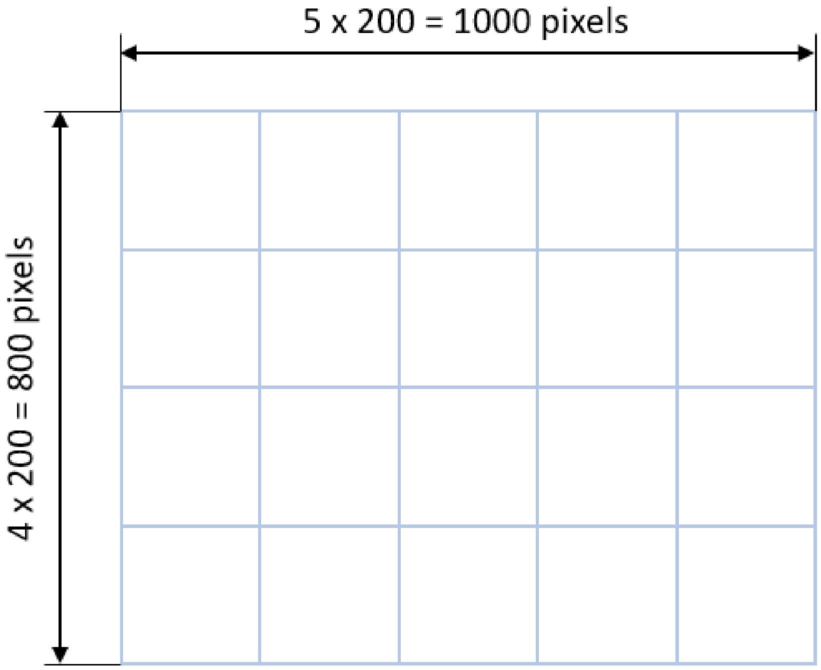
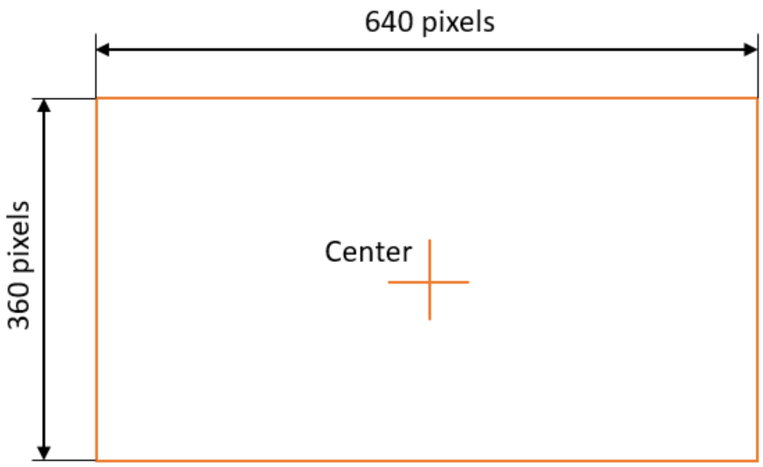
How to deal with tilemaps, camera and scroll

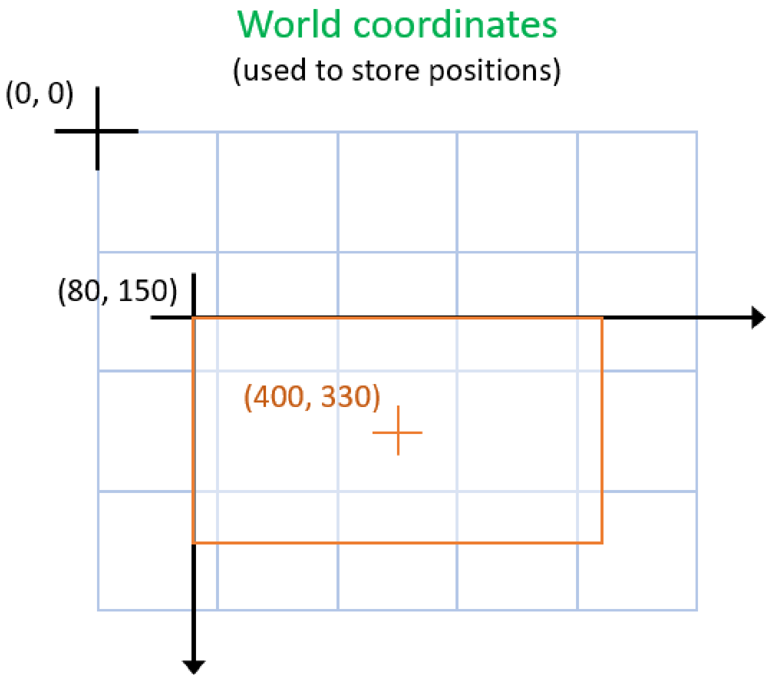
Here I will explain how you could handle camera position if you have a scenario made as a tilemap. A tilemap is a simple 2D array where each position is the ID of a tile. As an easy example, let's say you have a tilemap of 5x4 tiles, and tile size is 200x200 pixels. This makes its total size 1000x800 pixels.



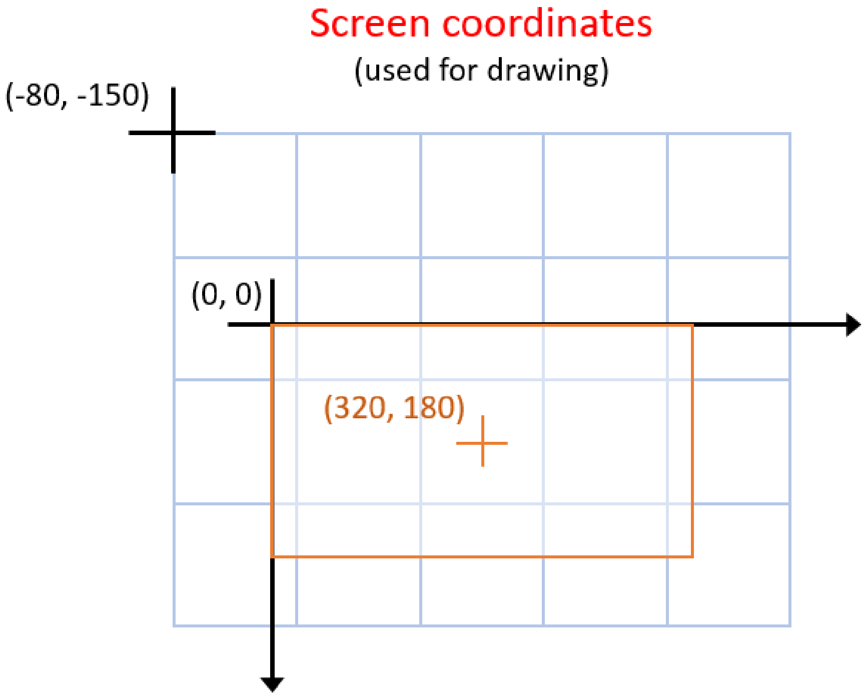
This is larger than the Vircon32 screen (640x360 pixels), so we can't show the whole map at once. We then need to have a "camera" that will represent our screen view. Our camera view is a rectangular region the size of the screen. We could say its center is the camera position, or in other words, the point at which the camera is pointing. But we will see everything inside the rectangle.



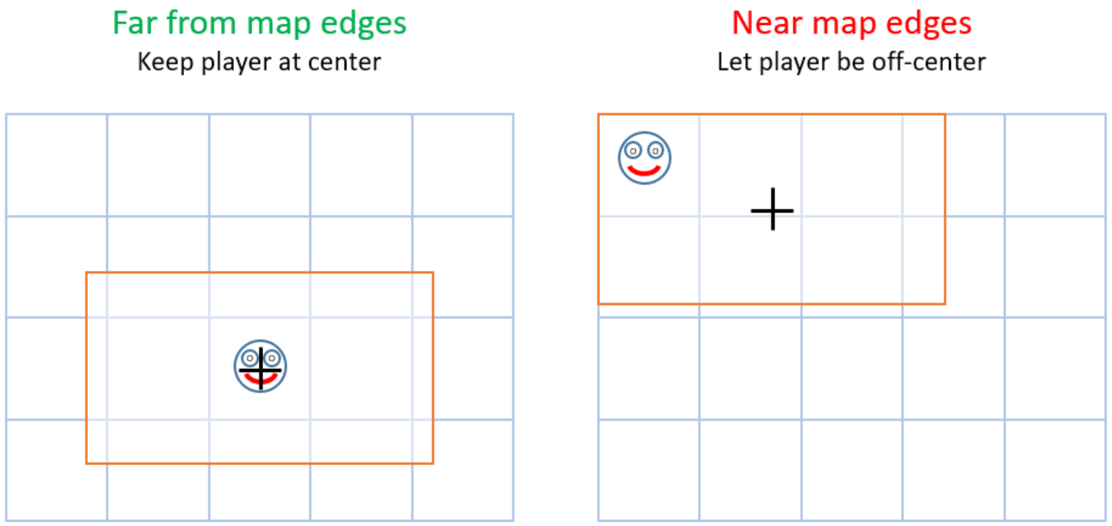
Usually there is some element inside the game level (like our character) that we want to keep centered on the screen. So we first need to position the character inside the game world and then we will place the camera center over the character. This determines which part of the map we will be able to see.



Once we have all elements positioned within the game world, we will need to convert all positions to screen coordinates in order to draw. This is done using the vector between both reference points (top-left corner of map and screen). This will determine tile drawing positions for the view we have chosen.



When our character is near the edge of the map, centering the view on it would make the camera show parts outside the map itself. In most games we will want to avoid that and keep the view within the level. To achieve this we need to stop having our character in the center of the screen.



Far from map edges

Keep player at center

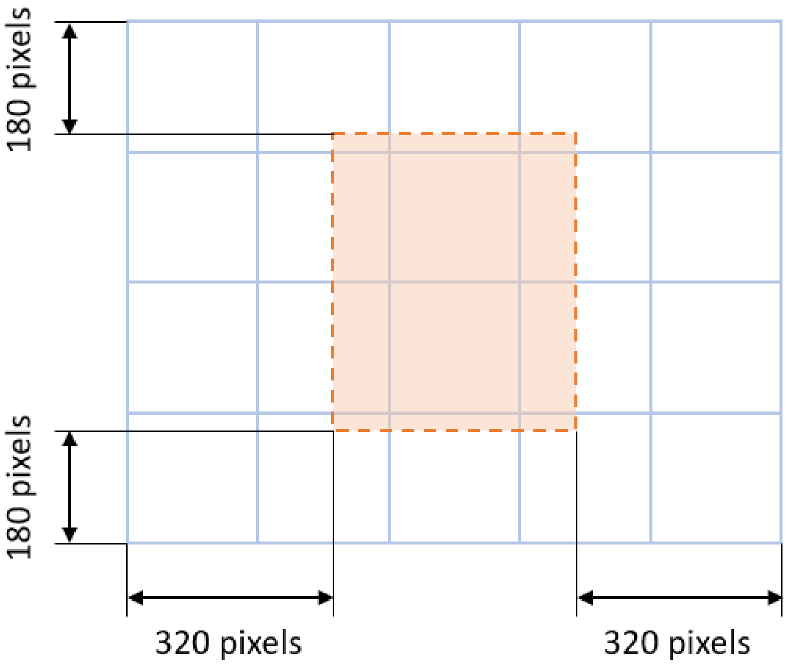
Near map edges

Let player be off-center

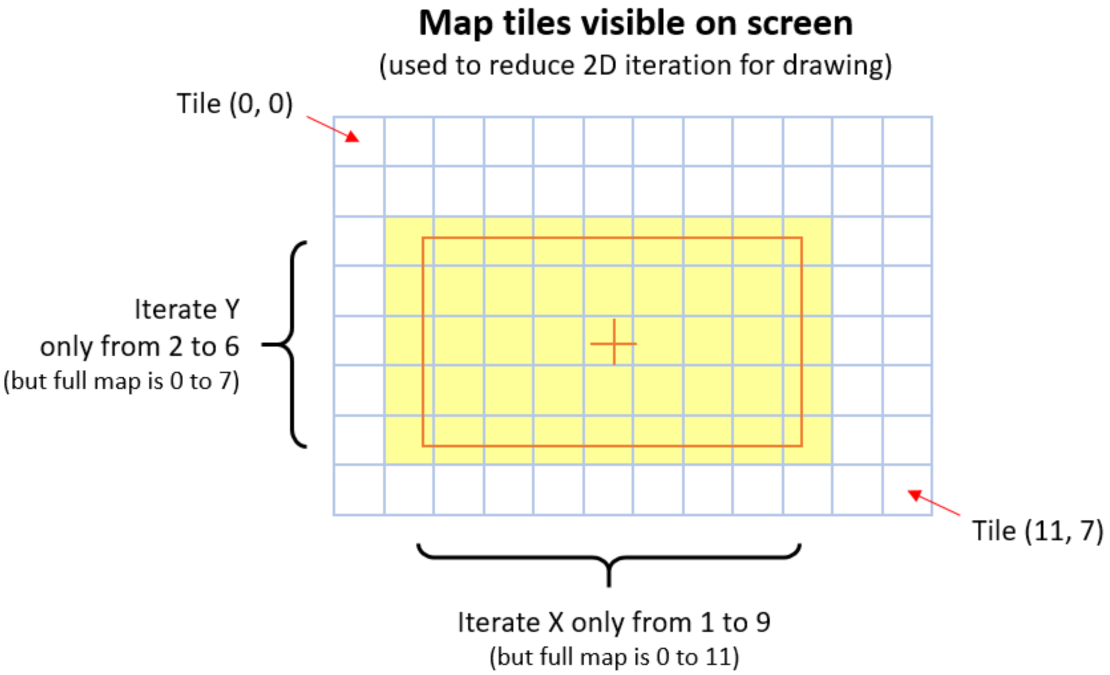
To do this we will have to determine which region within the map would be valid for our camera to point, and clamp camera center to those limits. We can find that region by taking the map edges and reducing them by half the width and height of the screen (i.e. the distance of camera edges to its center).

Region where camera center must be

(if the screen must keep within the map)



Lastly, drawing the tilemap in itself is quite easy: you can just do a loop in X and Y to draw every tile. Still, if your map is large, most of it will be out of the screen. In this case you will need to optimize, and draw only the tile range in X and Y that can actually be seen.


Map tiles visible on screen
(used to reduce 2D iteration for drawing)
Tile (0, 0)
Iterate Y
only from 2 to 6
(but full map is 0 to 7)
Tile (11, 7)
Iterate X only from 1 to 9
(but full map is 0 to 11)