Compsci.373 Tutorial 8

COMPUTER GRAPHICS IV

Today’s outline

- A look at the code for the final Graphics assignment
- Texture mapping
- Some exercises

The Assignment skeleton code

Your next assignment will be up soon!

This assignment will follow the same format as your last assignment, but there will be 10 theory questions, and 4 programming questions.

The Assignment skeleton code

There isn’t much more we need to look at in order to get the Ray Caster rendering to the screen, so lets have a look at the last pieces of skeleton code.

Lets look at the skeleton code

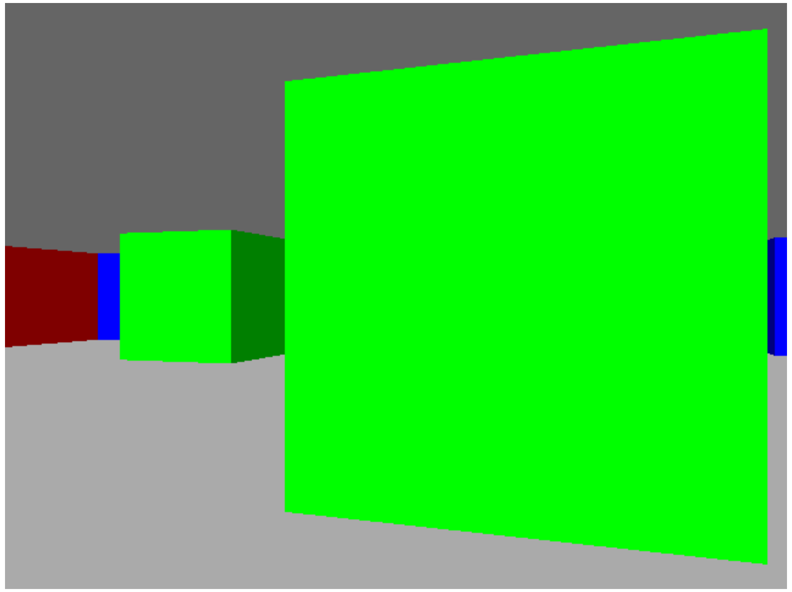
A quick recap on Ray Casting

Recall the Ray Casting procedure up to now:

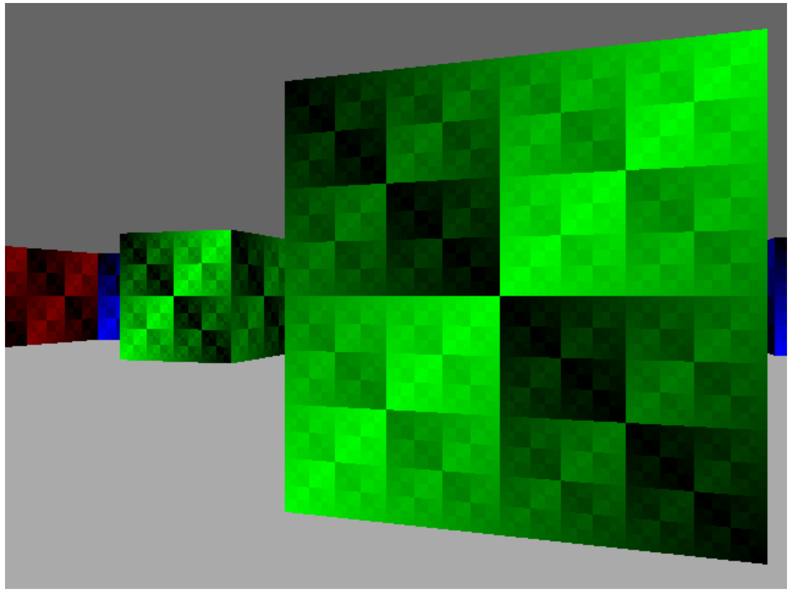
1. Define our rays
2. Normalize our rays
3. Find the step sizes for our rays
4. Step the rays into the world until they hit something
5. Undistort the ray lengths
6. Calculate wall strip heights

Texture mapping

All that's left is to define texture mapping. This is surprisingly very simple once you have your ray in the world.



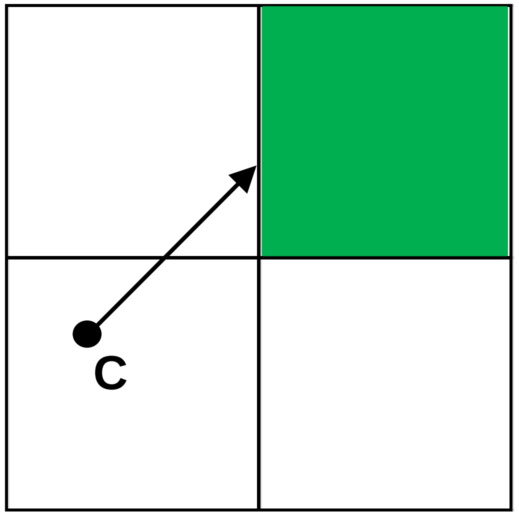
What we have so far

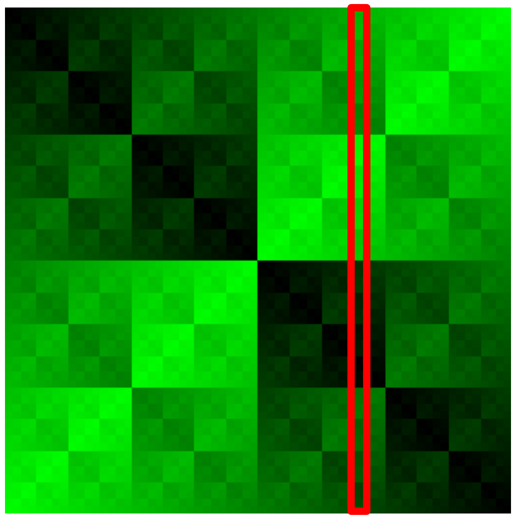


What we want

Texture mapping

We know where in the world a ray hits, but we need to calculate where along the wall the ray hit. If we can do this, then we can calculate the texture column we want to draw.



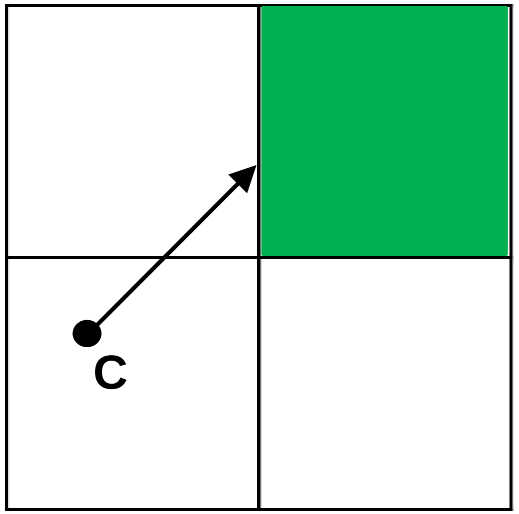


The texture column we want to use

Texture mapping

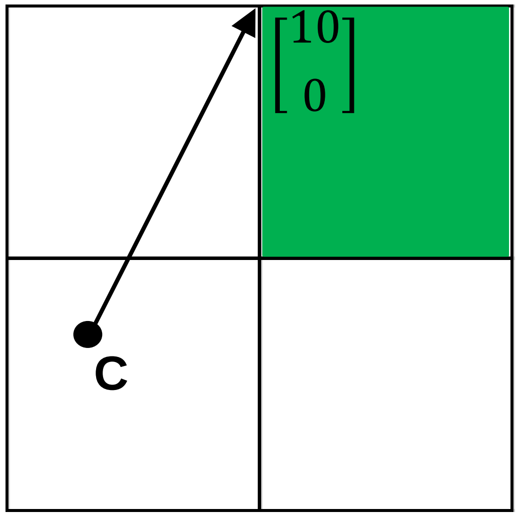
To do this, we turn to integer division. For an example, let us focus on vertical walls (walls hit by vertical rays).

Because the wall extends along the y axis, we need to find the local y coordinate of the ray hit point within the grid square.



If we perform the integer division of the hit coordinate by the size of a wall, then we will get the row number of the wall that was hit by the ray. If we take the remainder of this operation, then we get the y offset of the hit within the grid square, which is the position local to the wall where the ray hit!

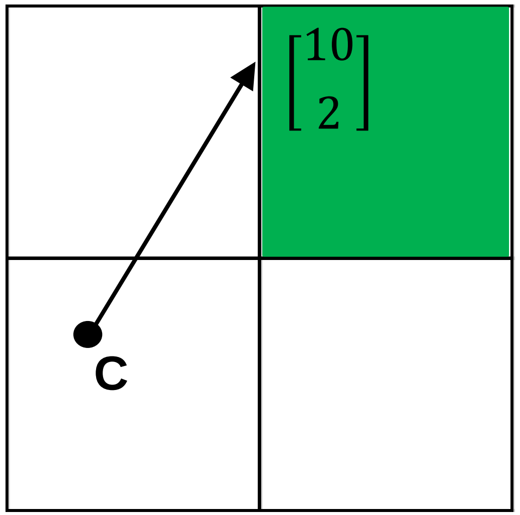
Texture mapping



Texture column = 0

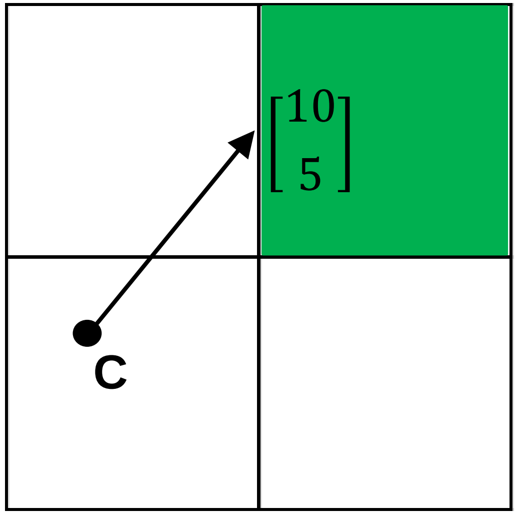
Note: The ray hit position is equal to $\mathbf{C} + \boldsymbol{R}_n$

Texture mapping



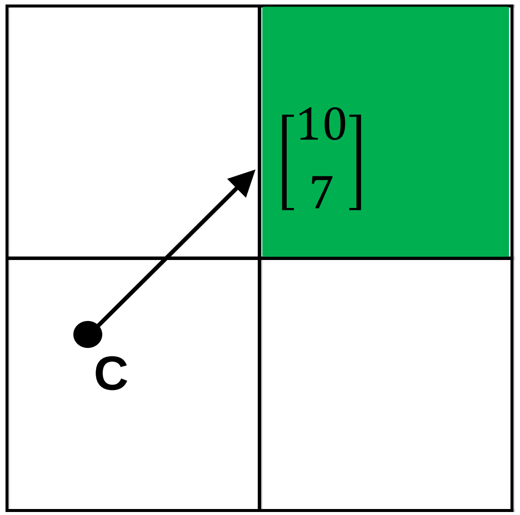
Texture column = 2

Texture mapping



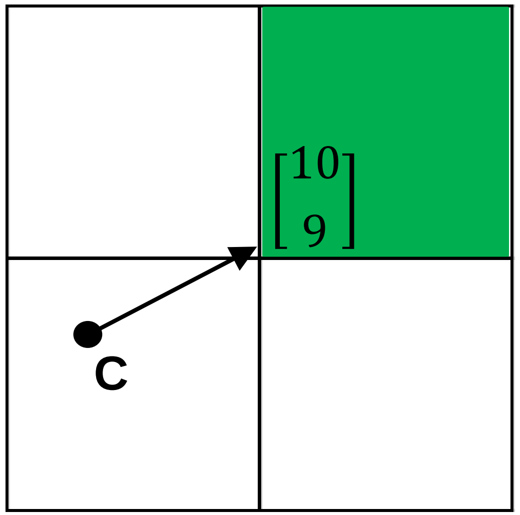
Texture column = 5

Texture mapping



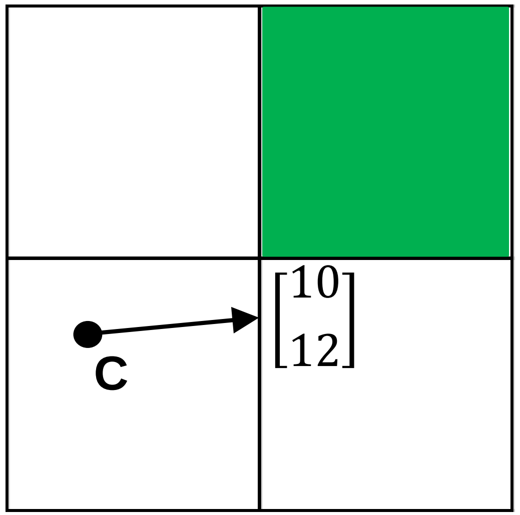
Texture column = 7

Texture mapping



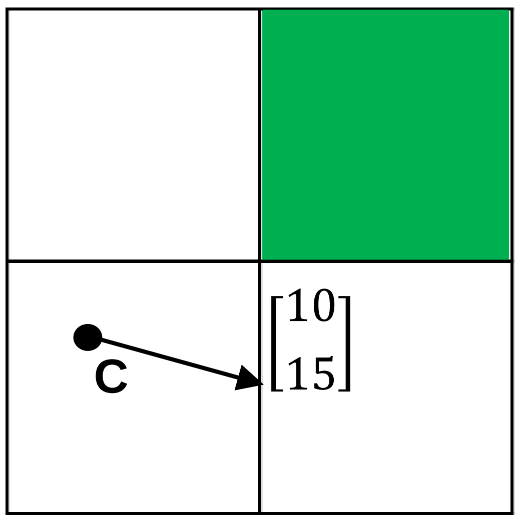
Texture column = 9

Texture mapping



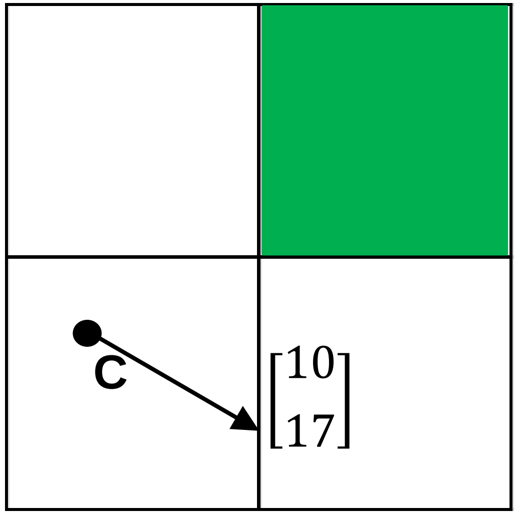
Texture column = 2

Texture mapping



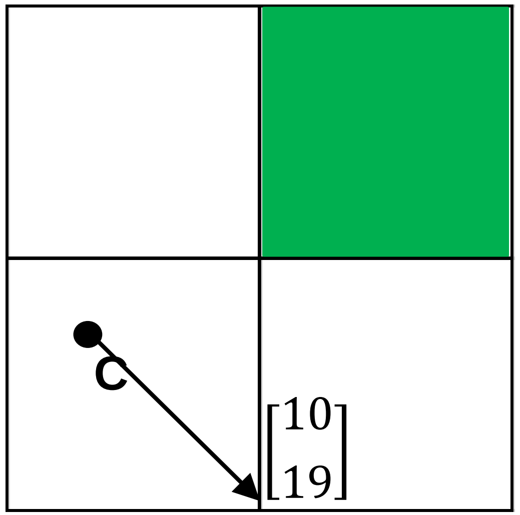
Texture column = 5

Texture mapping



Texture column = 7

Texture mapping



Texture column = 9

Texture mapping

For vertical walls, this can be generalized using modulo arithmetic. If **y** is the y coordinate of the intersection point, then the texture column will be equal to:

y MOD wall_size

For horizontal walls, the procedure is the same, except that we use the x coordinate of the hit point.

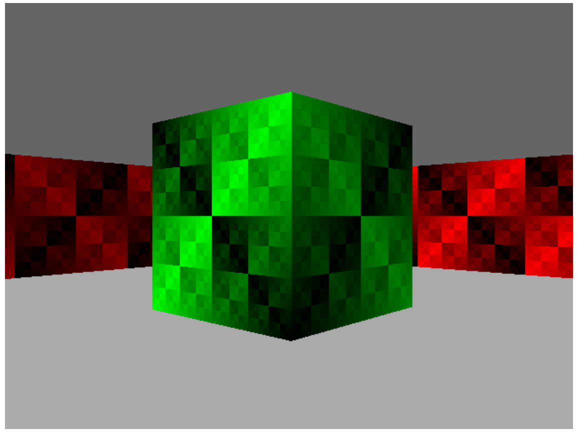
x MOD wall_size

Texture mapping

We will also need to reverse the texture in two cases:

- For vertical walls when the ray is facing left
- For horizontal walls when the ray is facing down

To see why, it helps to put yourself inside one of the grid squares of the world, and visualize which direction the textures should go in for each of the four walls in the grid square.



This happens when we don't apply reversals

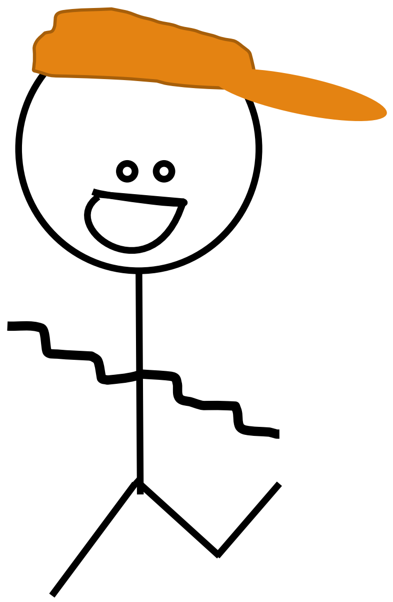
Some exercises

Given a player position of $\begin{bmatrix}15\\16\end{bmatrix}$ and a ray $\begin{bmatrix}5\\6\end{bmatrix}$.

1. What is the texture coordinate given by the ray intersection?
2. What about if the ray was $\begin{bmatrix}-5\\6\end{bmatrix}$?
3. What about if the ray was $\begin{bmatrix}3\\4\end{bmatrix}$?
4. What about if the ray was $\begin{bmatrix}3\\-6\end{bmatrix}$?

That's all from me!

And now back to Trevor!



Quick Curve Exercises

(a) Given: $\mathrm{p}(t) = (3t, t^3)$ and $q(t) = (4t + 3, 2t^2 + 4t + 1)$. Consider t to be restricted between the range [0,1]. Are these curves C^0, G^1 and/or C^1?

(b) Given the half circle curve below. Find a parametric equation to define this curve.