

3D OpenGL Game

CS 532 – B1

Fall 2015

Mark J. Musante

mark.musante@gmail.com

Minecraft



How far did I get?

- Python 2.7 or 3.x
- “Infinite” world
- 30fps
- Using the mouse to look around
- W/A/S/D keys to move around
- Can place and remove blocks
- Space bar to jump
- “Fall” when moving over empty space

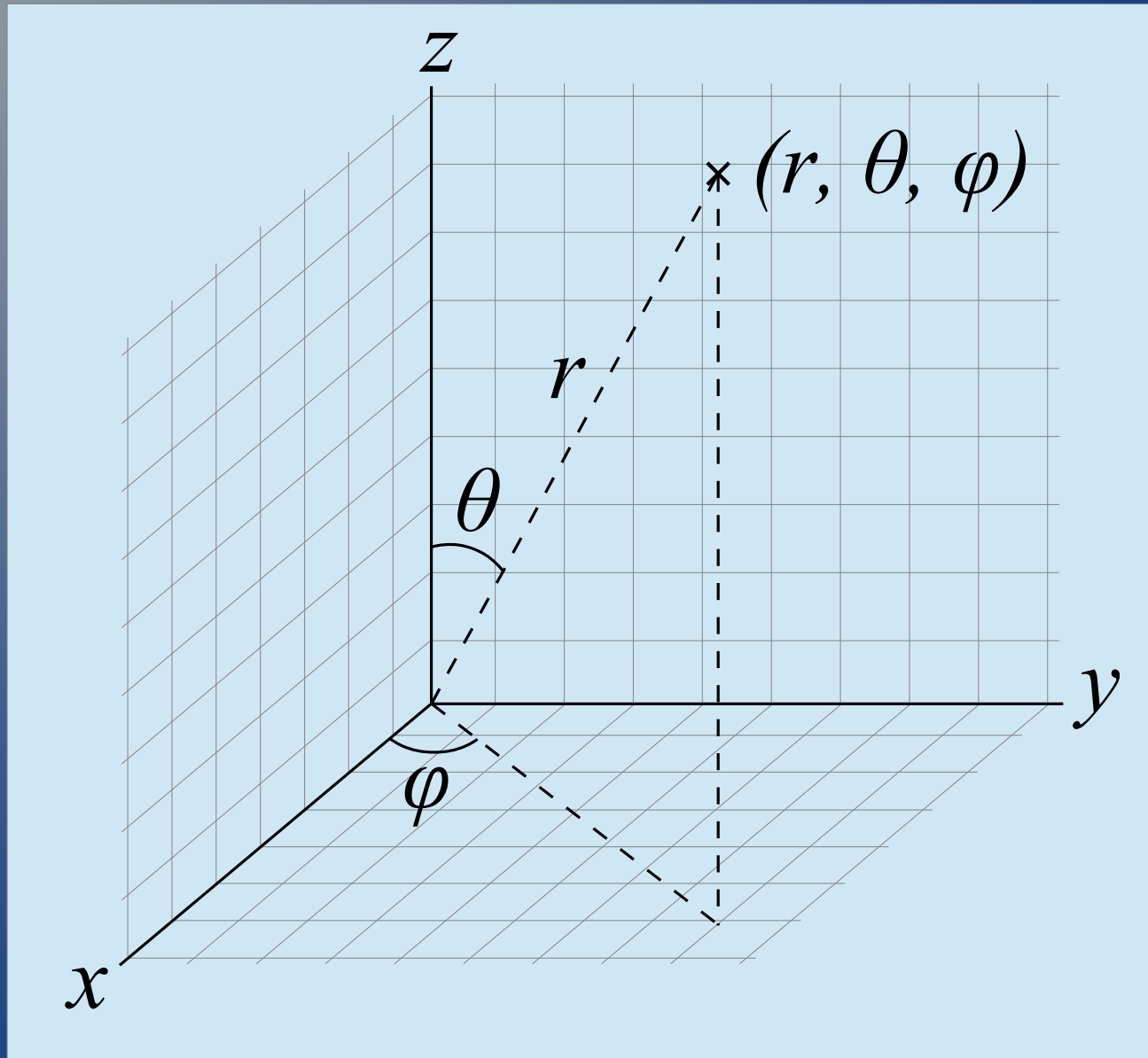
“Infinite” world

- Using python's “tuple” to define the game space
- A hash defines everything:
 - Key: “tuple” coordinates
 - e.g. (4, 5, 6) for the key means the block is at x=4, y=5, z=6
 - Value: type of block
 - Grass, dirt, stone. Others would be easy to add

Getting 30fps

- Using glutTimerFunc()
 - Glut will call it once only
 - Each time, call glutTimerFunc() again
 - Using 33ms gets us a bit over 30fps
- Inside glutTimerFunc():
 - Calculate new x,y,z coordinates for player
 - (Jumping? Falling? Moving into a block?)
 - Call glutPostRedisplay()

Using the mouse to look around



* Image taken from Wikipedia

Using the mouse to look around

- Grab x/y coordinates from the mouse
- Convert to spherical coordinates
 - mouse-X gives us phi
 - mouse-Y gives us theta
 - $x = \sin(\text{phi}) * \cos(\text{theta})$
 - $y = \sin(\text{phi}) * \sin(\text{theta})$
 - $z = \cos(\text{phi})$
- Use spherical coordinates to pass to `gluLookAt()`

Using W/A/S/D

- W=forward, S=backward, A=left, D=right
- Re-use the x/y/z values from gluLookAt to move the player in the direction they're looking
- Timer function:
 - On key down, start moving
 - On key up, stop moving
- glutSetKeyRepeat() to turn off auto key repeat

Placing and removing blocks

- Left click removes a block in the world
- Right click places a block from the world
- How do we know which block?
 - Again, re-use “LookAt” coordinates
 - Start at player location, move outward until a block is intersected
 - If removing, then remove it
 - If placing, insert a block “before” it
- Draw a cube outline as a UI element

Jumping

- When keydown is the space bar:
 - If we're not falling or jumping, start a jump
 - Save `start_time`
 - Jump is just $\sin()$ from 0 to π .
 - The part of the jump we're in is: $(\text{current_time} - \text{start_time})$ times a factor
- When jump is complete, check space bar status and re-jump if no keyup event.

Falling

- If player moves over a space without a block, start falling
- Falling just decrements the y-value of the player's position until it intersects a block

Demo

Remaining Work

(To make it feel complete)

- Falling should accelerate to terminal velocity
- Generate different landscapes
- Mouse exclusivity
- Normalise walking speed
- Lighting effects
- Ability to place down different blocks