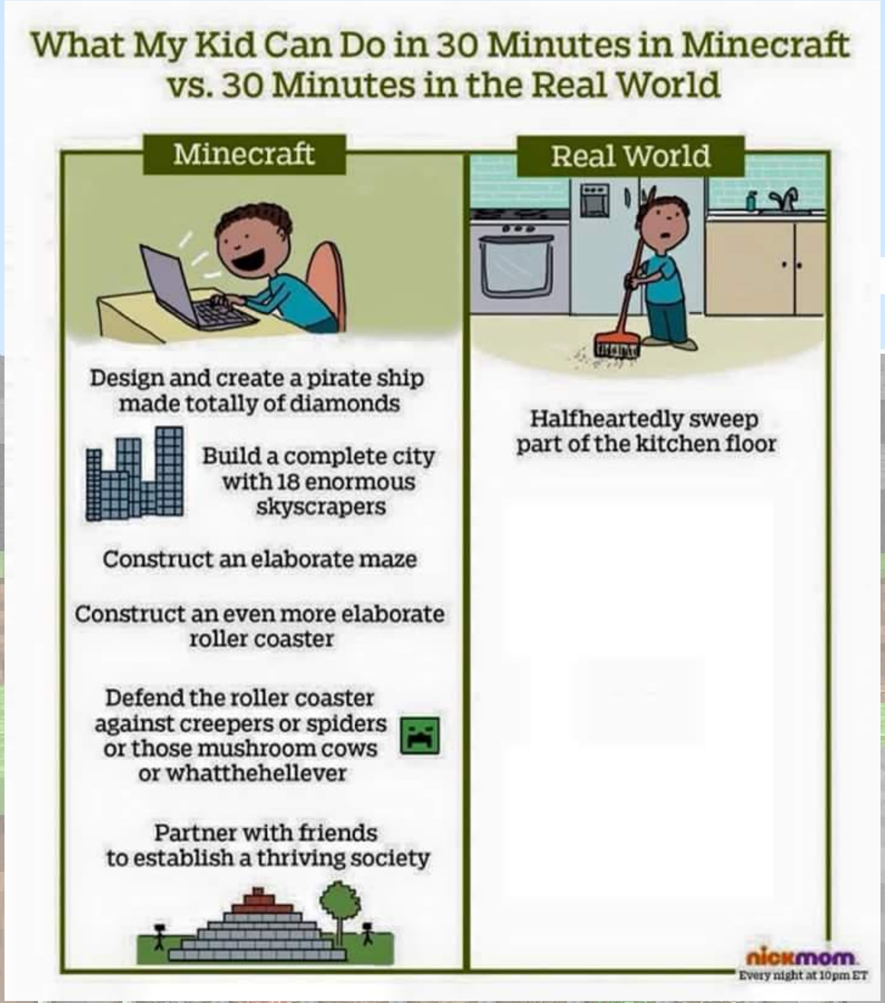
WELCOME

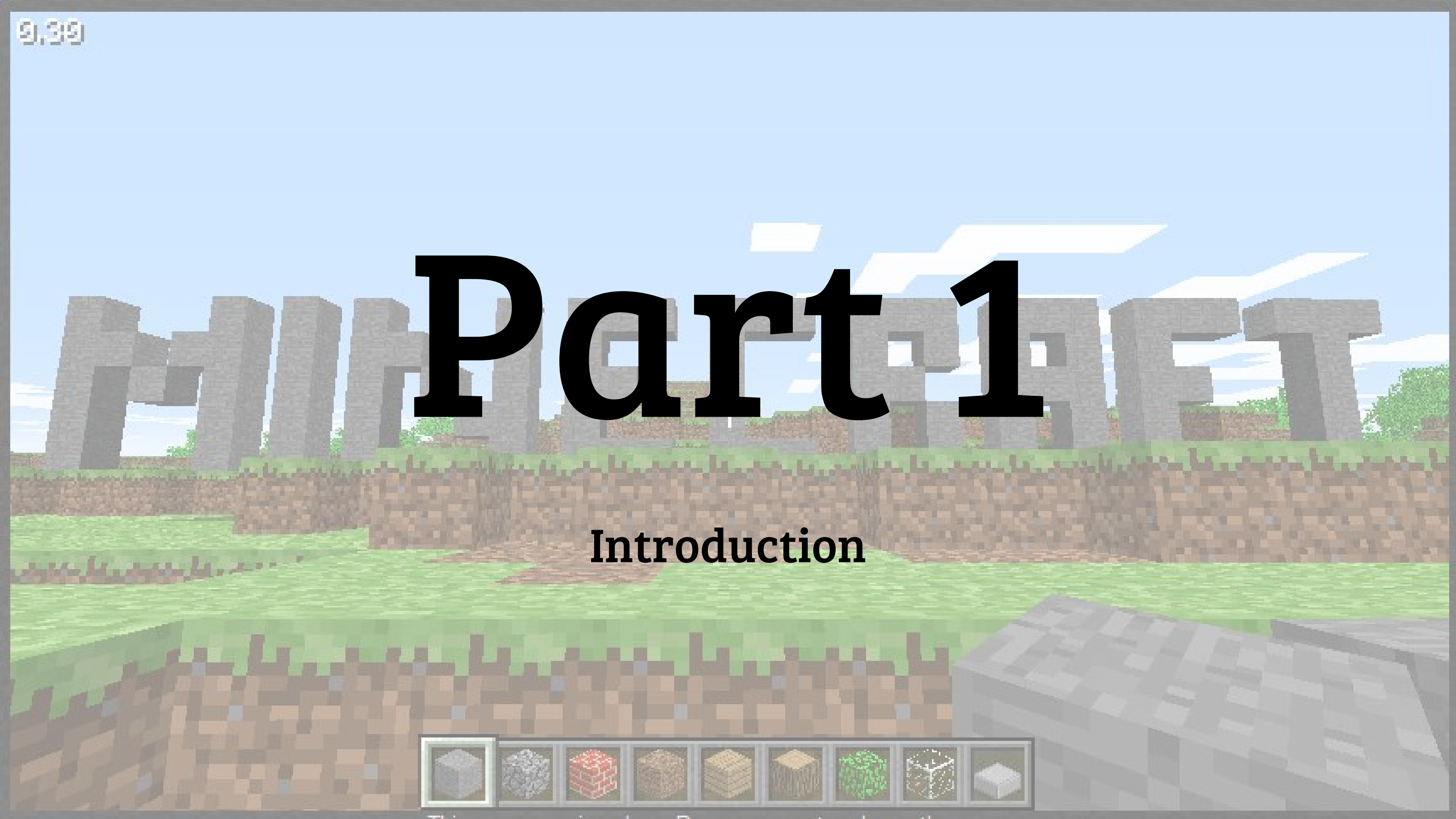
Hacking Minecraft with Python on Raspberry Pi



What My Kid Can Do in 30 Minutes in Minecraft vs. 30 Minutes in the Real World

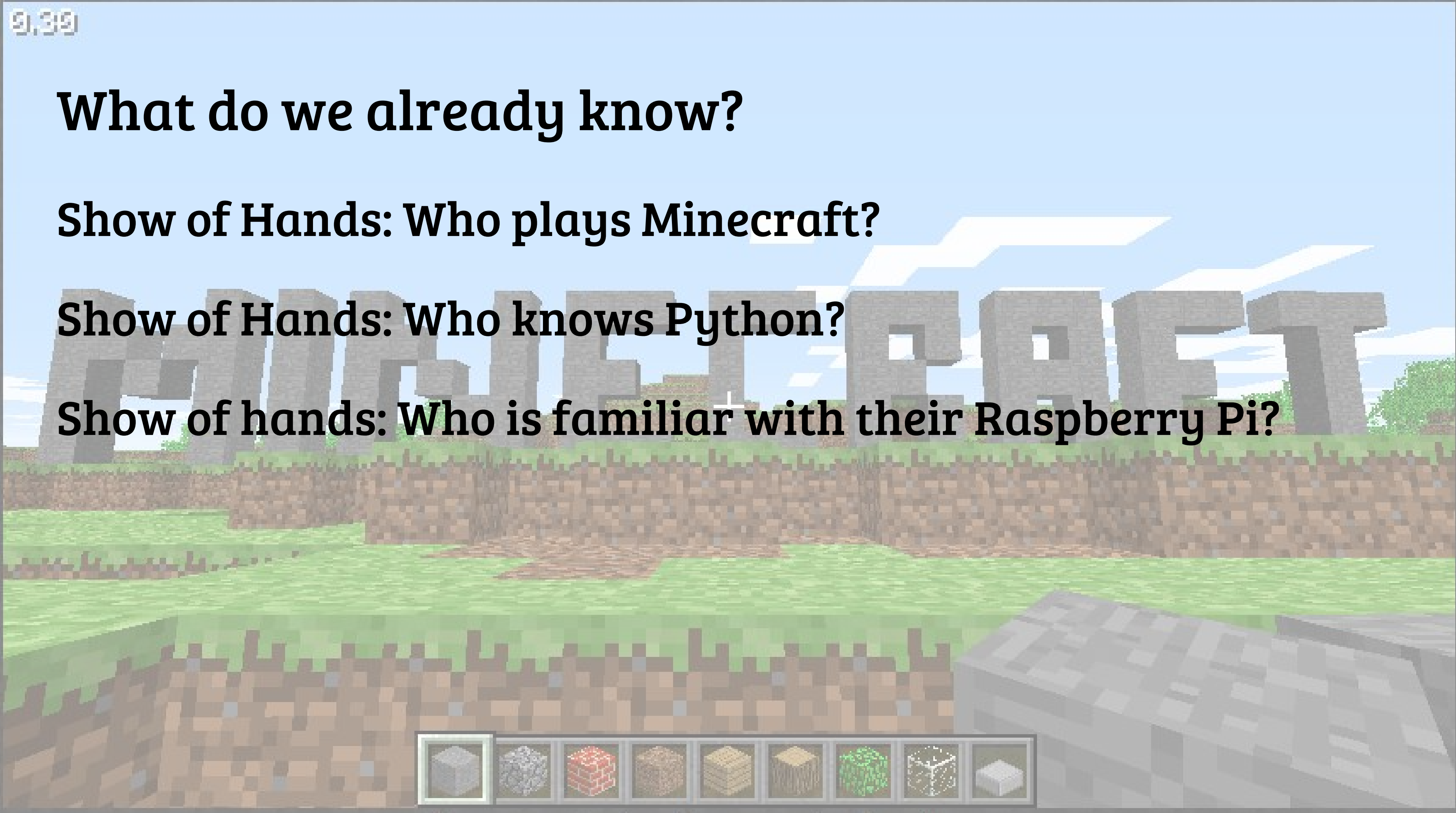
Minecraft	Real World
Design and create a pirate ship made totally of diamonds Build a complete city with 18 enormous skyscrapers Construct an elaborate maze Construct an even more elaborate roller coaster Defend the roller coaster against creepers or spiders or those mushroom cows or whatthehellever Partner with friends to establish a thriving society	Halfheartedly sweep part of the kitchen floor

9:39



Part 1

Introduction

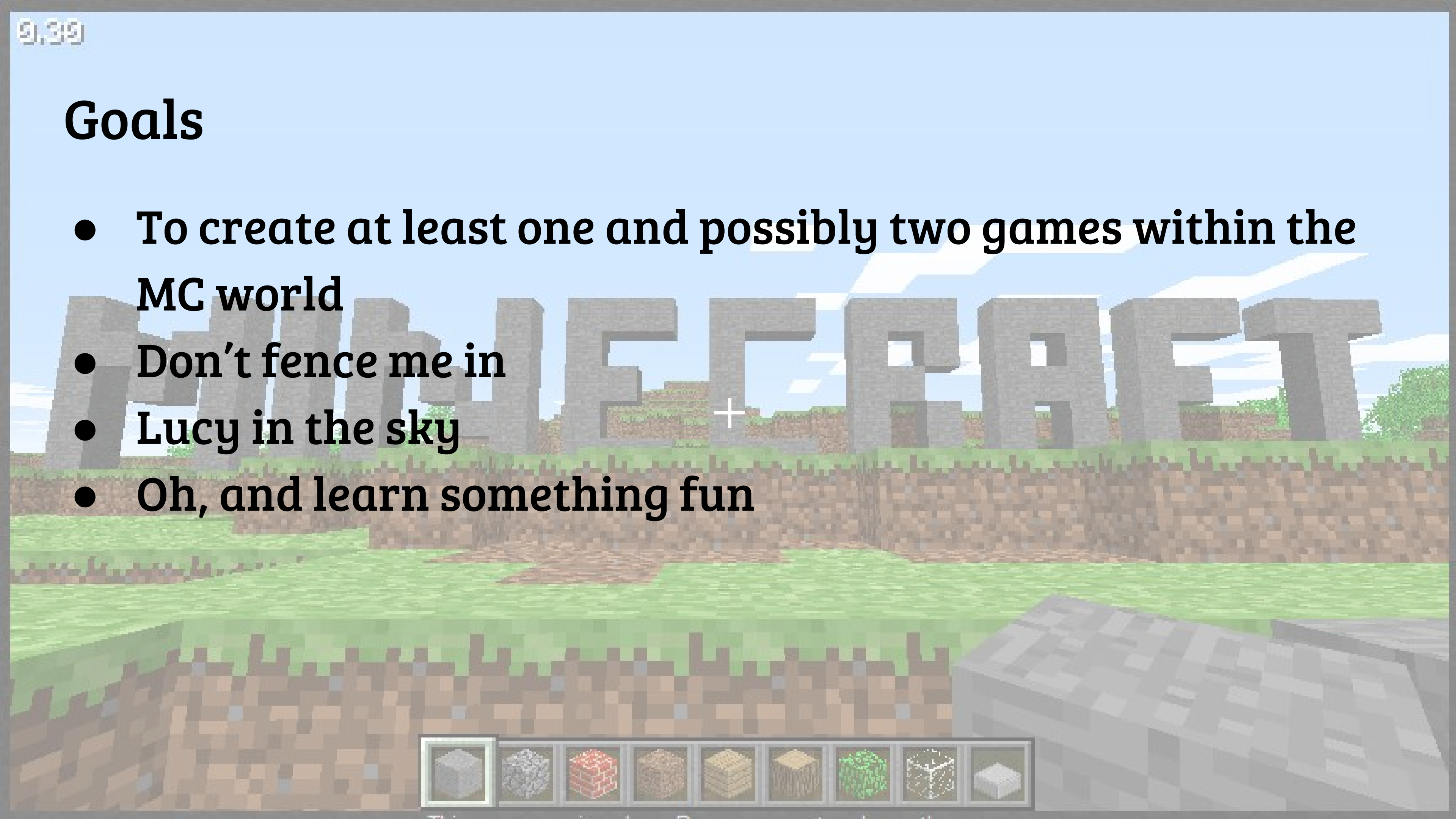


What do we already know?

Show of Hands: Who plays Minecraft?

Show of Hands: Who knows Python?

Show of hands: Who is familiar with their Raspberry Pi?

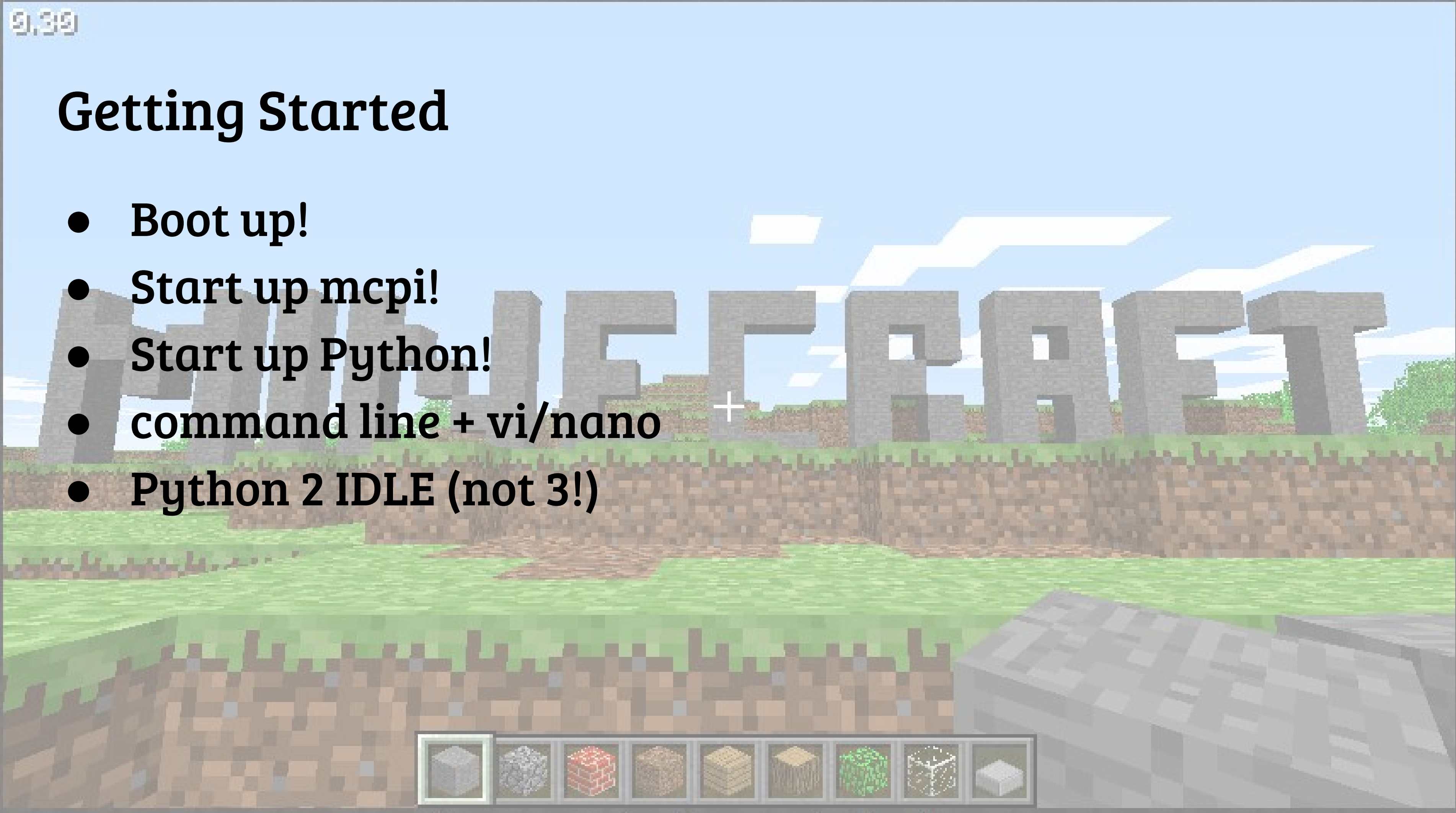


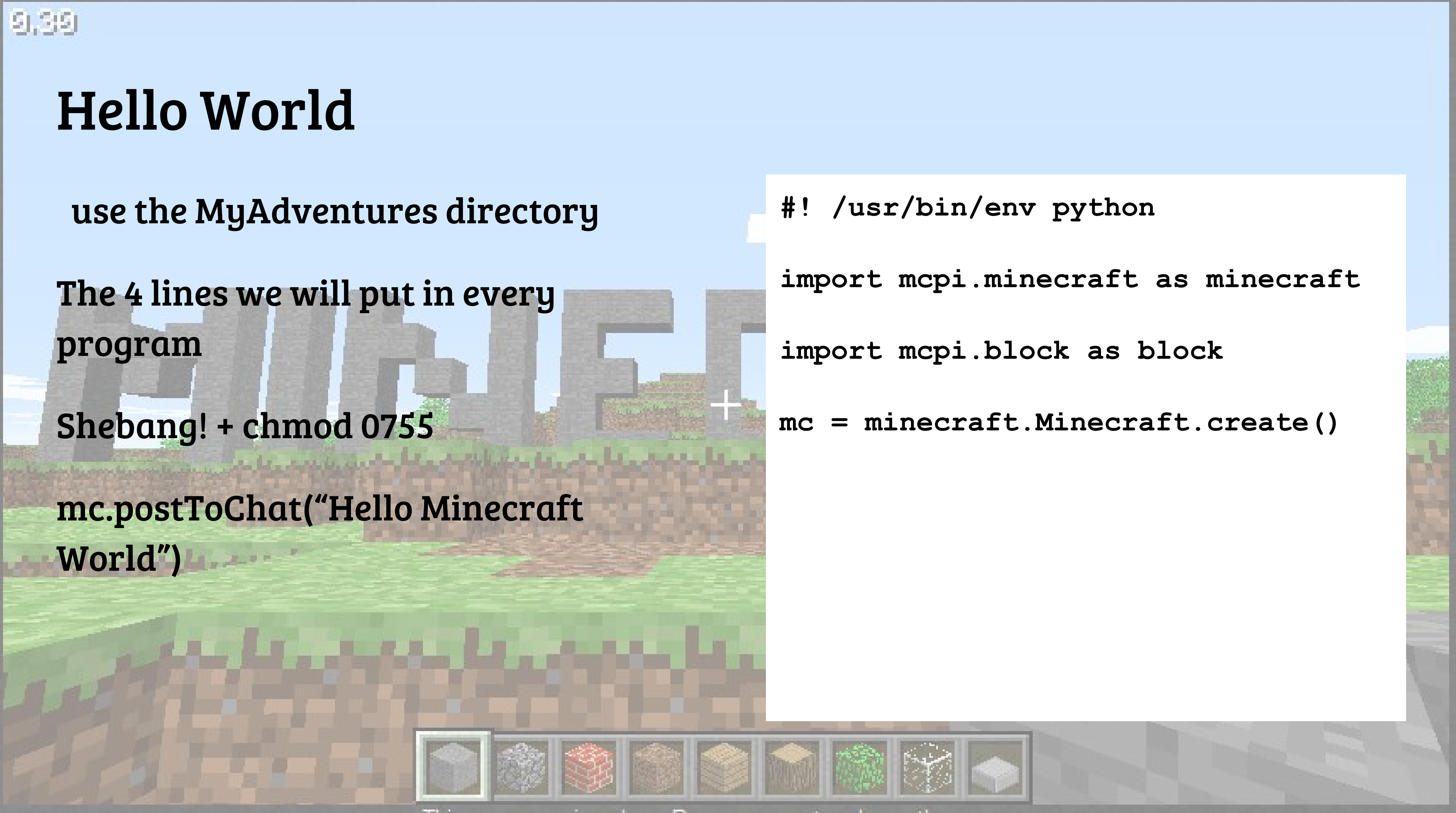
Goals

- To create at least one and possibly two games within the MC world
- Don’t fence me in
- Lucy in the sky
- Oh, and learn something fun

Getting Started

- Boot up!
- Start up mcpi!
- Start up Python!
- command line + vi/nano
- Python 2 IDLE (not 3!)





Hello World

use the MyAdventures directory

The 4 lines we will put in every program

Shebang! + chmod 0755

mc.postToChat(“Hello Minecraft World”)

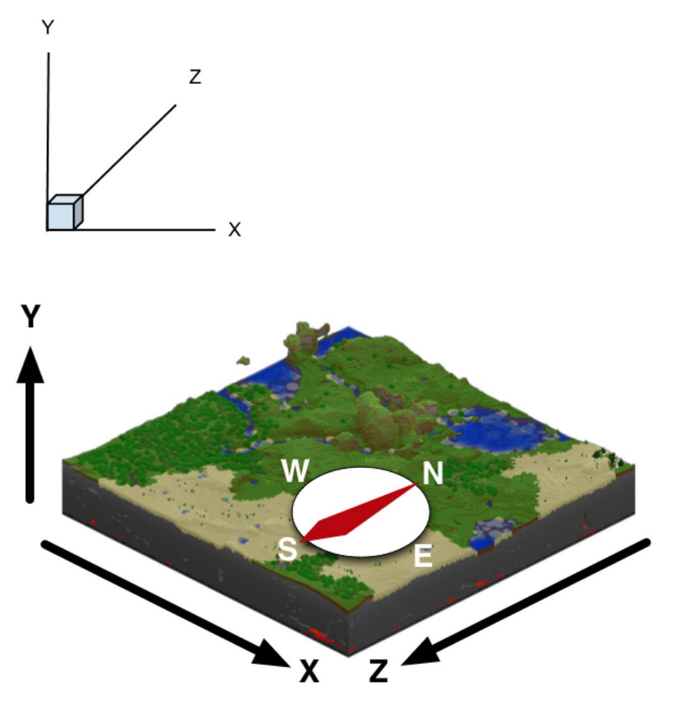
```
#! /usr/bin/env python

import mcpi.minecraft as minecraft

import mcpi.block as block

mc = minecraft.Minecraft.create()
```

Part 2

The Minecraft API

Where am I, Steve?

- X,Y,Z coordinates in Minecraft
- Look at the top left of your screen
- X is +East and -West
- Y is +Up and -Down
- Z is +North and -South



Getting your position with Python

Ask minecraft where a player is

mc.playerGetTilePos() returns the set of X,Y,Z coordinates of the cube the player is standing on

```
pos=mc.Player.getTilePos()

print pos.x, pos.y, pos.z

# Fancier ...

print “x is %d y is %d z is %d” %
(pos.x, pos.y, pos.z)
```

Moving your player with Python

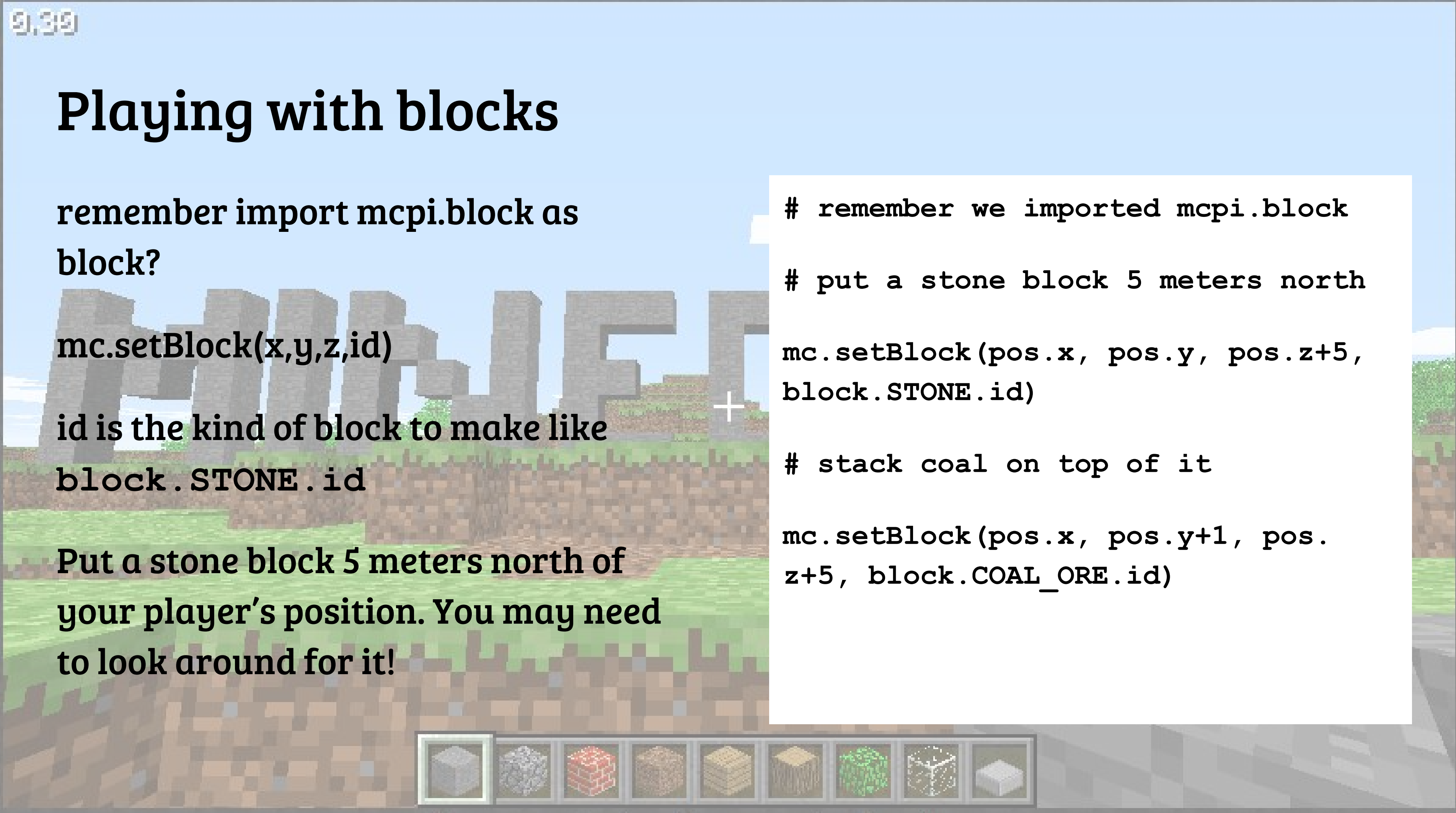
Tell minecraft where to put a player

mc.player.setTilePos(x,y,z) moves Steve to the tile at the set of coordinates. Be Careful!

What do you think this line will do?

what if you subtract 20 from pos.y?

```
# remember pos is Steve’s location

mc.player.setTilePos(pos.x, pos.y+20, pos.y)

mc.player.setTilePos(pos.x, pos.y-20, pos.y)
```



Playing with blocks

remember import mcpi.block as block?

mc.setBlock(x,y,z,id)

id is the kind of block to make like `block.STONE.id`

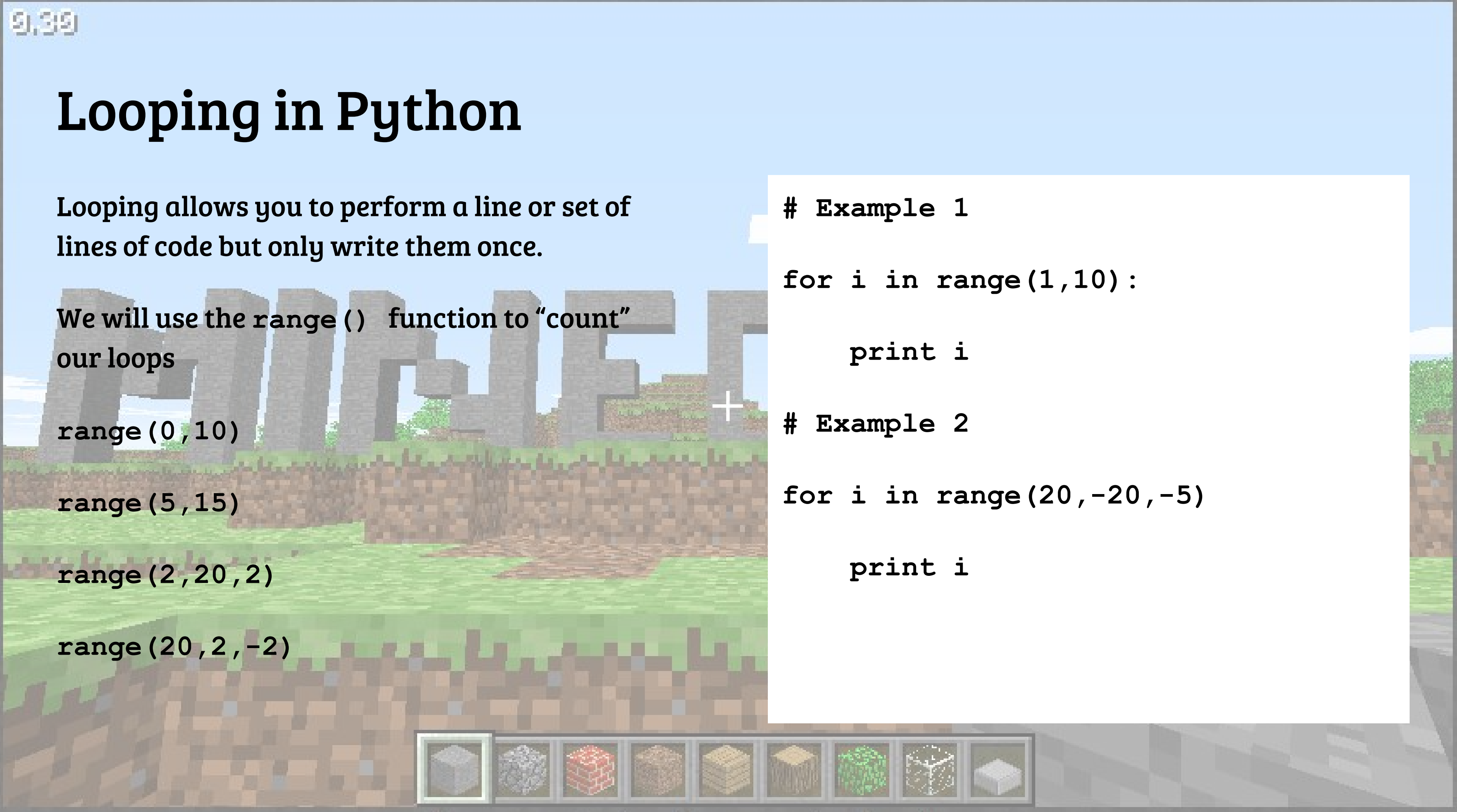
Put a stone block 5 meters north of your player's position. You may need to look around for it!

```
# remember we imported mcpi.block

# put a stone block 5 meters north

mc.setBlock(pos.x, pos.y, pos.z+5,
block.STONE.id)

# stack coal on top of it

mc.setBlock(pos.x, pos.y+1, pos.
z+5, block.COAL_ORE.id)
```

Part 3

Basic Python

Looping in Python



Looping allows you to perform a line or set of lines of code but only write them once.

We will use the `range()` function to “count” our loops

```
range(0,10)

range(5,15)

range(2,20,2)

range(20,2,-2)
```

```
# Example 1

for i in range(1,10):

    print i

# Example 2

for i in range(20,-20,-5)

    print i
```

Branching in Python

When you want to take a different action based on a condition.

```
# Example 1
if pos.y > 20:
    print “Steve is too high!”
# Example 2
if pos.y > 20:
    print “Steve is too high!”
else
    print “Steve is Okay”
```

Oh noes! Math!

Minecraft is a 3D world made up of blocks. Sometimes we want to know how far blocks are from each other or how far we are from a block. Thanks to an old greek dude called Pythagoras we have a formula that we can use. If we have two points: (x1, y1, z1) and (x2, y2, z2) the distance between them is:

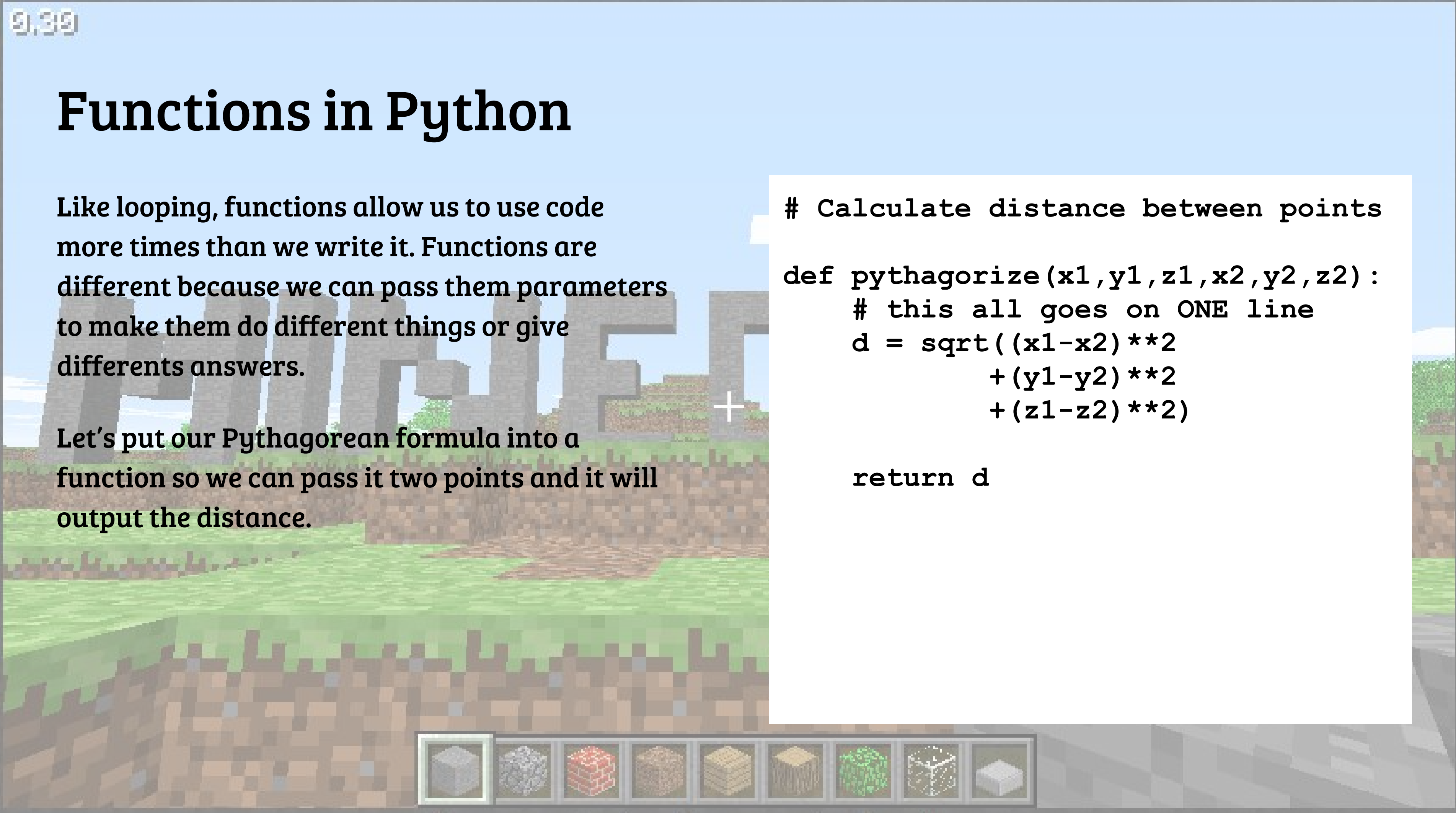
$$\sqrt{(x1-x2)^2+(y1-y2)^2+(z1-z2)^2}$$

# in Python x^2 is x**2

# in Python √x is sqrt(x) so…

```
d=sqrt((x1-x2)**2
      +(y1-y2)**2
      +(z1-z2)**2)
```

# That all has to go on one line!



Functions in Python

Like looping, functions allow us to use code more times than we write it. Functions are different because we can pass them parameters to make them do different things or give differents answers.

Let’s put our Pythagorean formula into a function so we can pass it two points and it will output the distance.

```
# Calculate distance between points

def pythagorize(x1,y1,z1,x2,y2,z2):
    # this all goes on ONE line
    d = sqrt((x1-x2)**2
            +(y1-y2)**2
            +(z1-z2)**2)

    return d
```

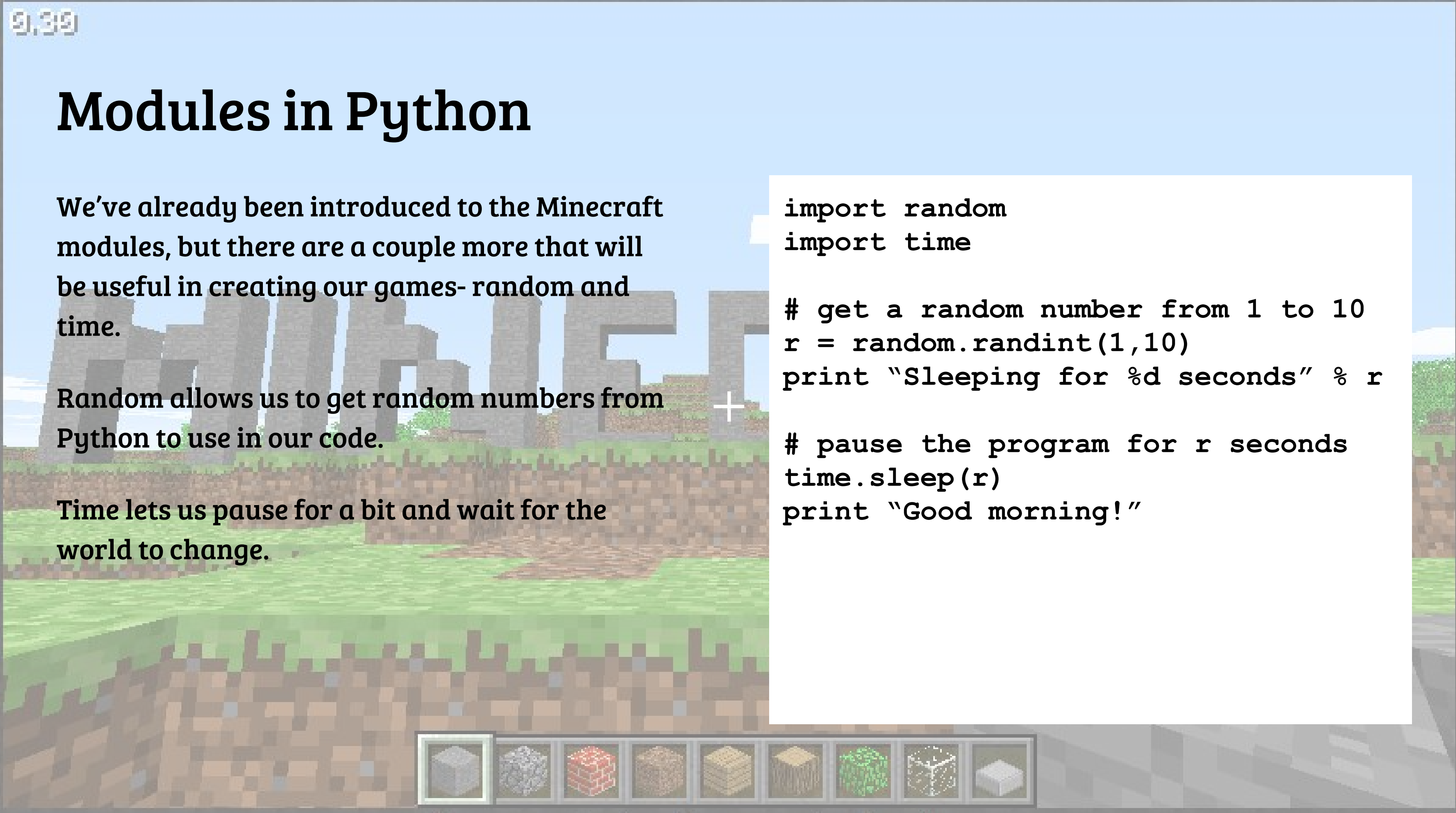
Modules in Python

We’ve already been introduced to the Minecraft modules, but there are a couple more that will be useful in creating our games- random and time.

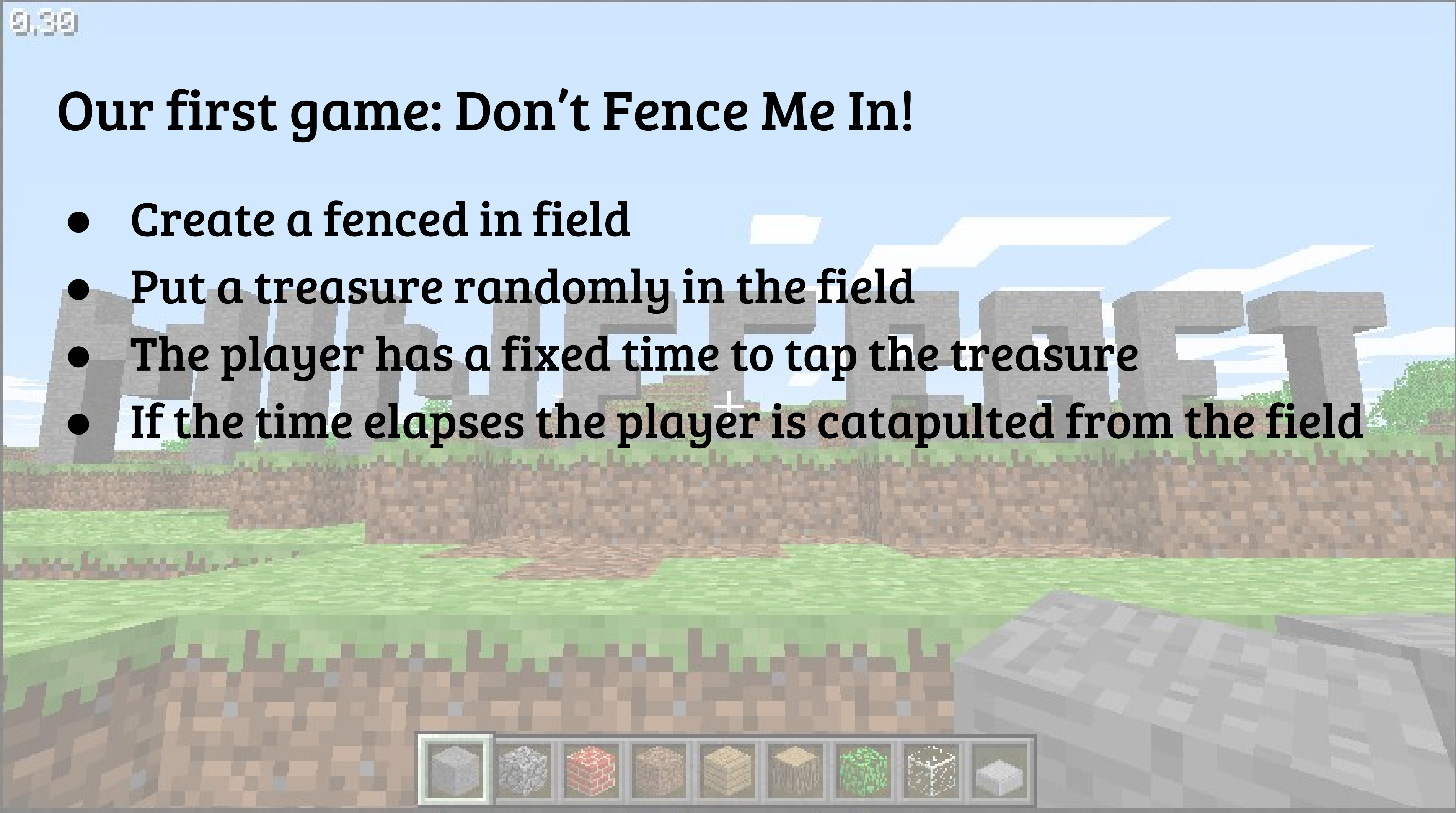
Random allows us to get random numbers from Python to use in our code.

Time lets us pause for a bit and wait for the world to change.

```
import random
import time

# get a random number from 1 to 10
r = random.randint(1,10)
print “Sleeping for %d seconds” % r

# pause the program for r seconds
time.sleep(r)
print “Good morning!”
```

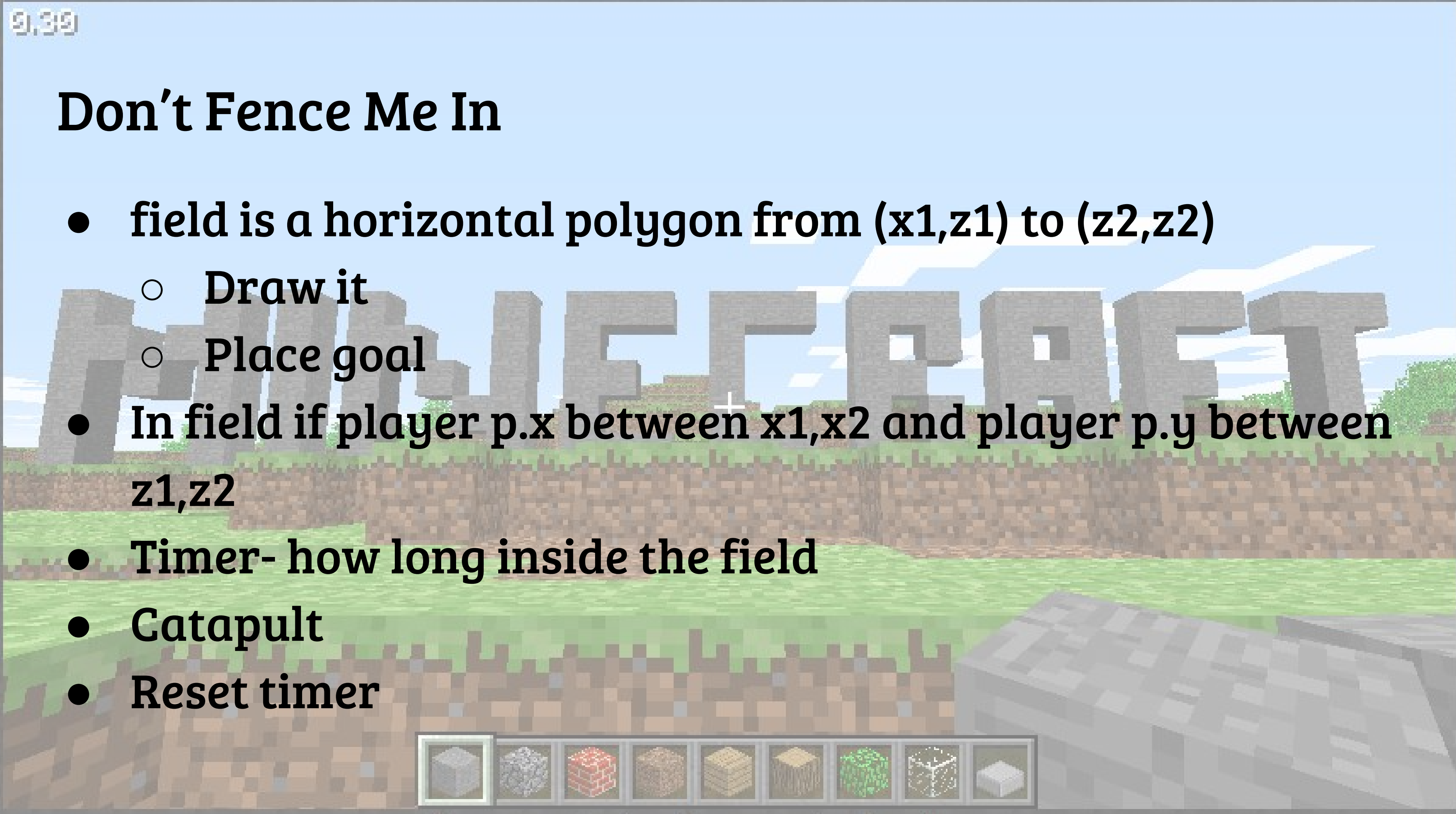


BREAK TIME!



Our first game: Don’t Fence Me In!

- Create a fenced in field
- Put a treasure randomly in the field
- The player has a fixed time to tap the treasure
- If the time elapses the player is catapulted from the field

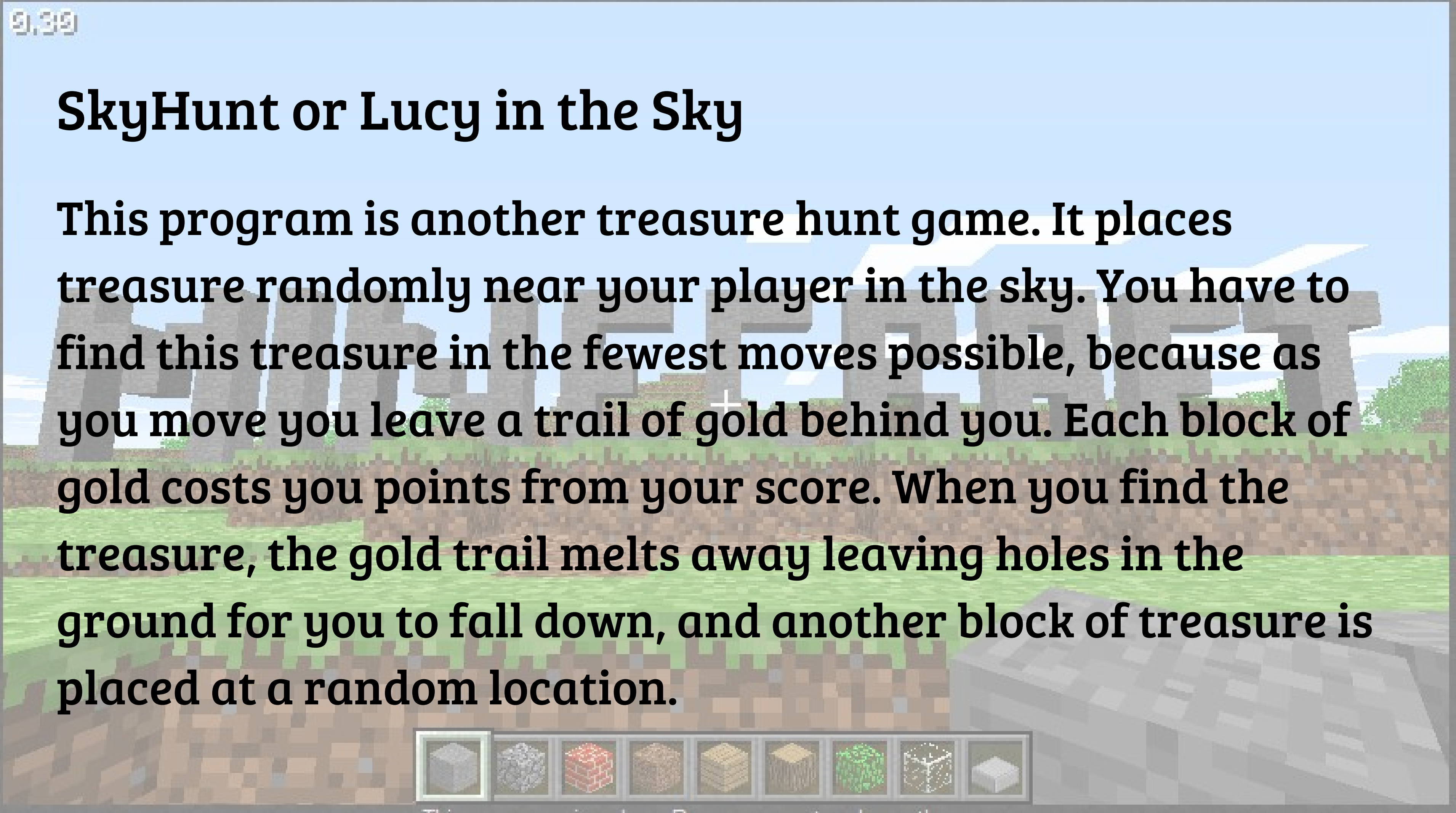


Don’t Fence Me In

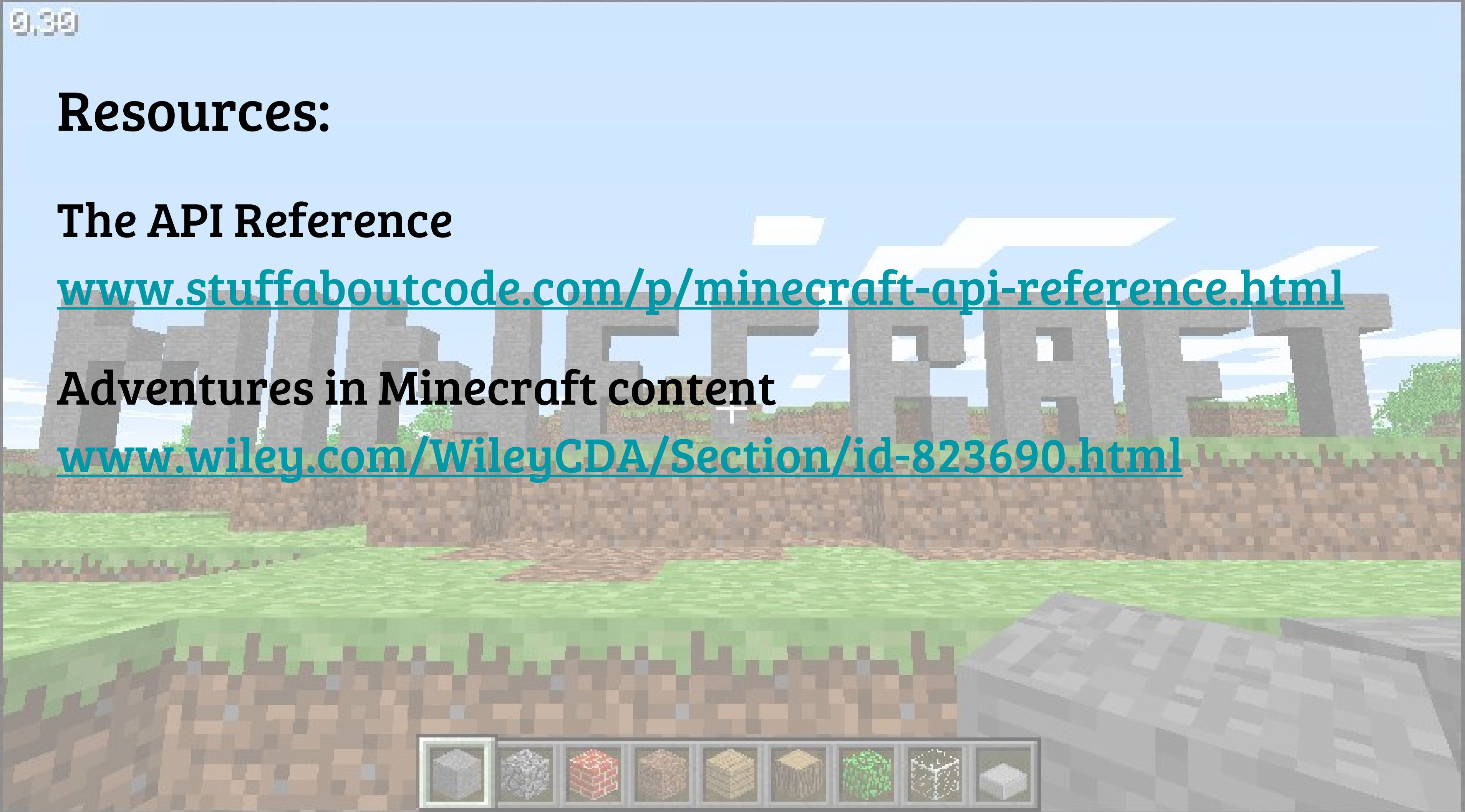
- field is a horizontal polygon from (x1,z1) to (z2,z2)
    - Draw it
    - Place goal
- In field if player p.x between x1,x2 and player p.y between z1,z2
- Timer- how long inside the field
- Catapult
- Reset timer

BREAK TIME!

SkyHunt or Lucy in the Sky



This program is another treasure hunt game. It places treasure randomly near your player in the sky. You have to find this treasure in the fewest moves possible, because as you move you leave a trail of gold behind you. Each block of gold costs you points from your score. When you find the treasure, the gold trail melts away leaving holes in the ground for you to fall down, and another block of treasure is placed at a random location.



Resources:

The API Reference

www.stuffaboutcode.com/p/minecraft-api-reference.html

Adventures in Minecraft content

www.wiley.com/WileyCDA/Section/id-823690.html