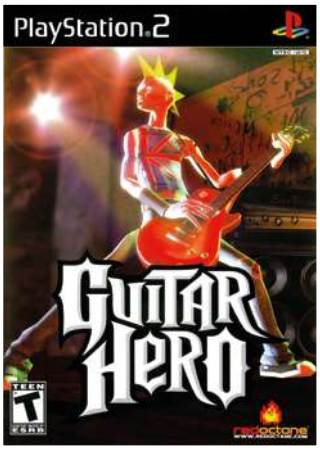

Guitar Hero, while hugely popular on the PS2, also had versions on numerous other consoles and computers.

Source Code

Make a Guitar Hero-style rhythm action game

Channel your inner rock god with our plucky homage

AUTHOR
MARK VANSTONE

In 2005, a new game let players live out their rock star fantasies. Having already made specialised game controllers for Konami, American firm RedOctane enlisted the help of Harmonix to help them create *Guitar Hero*: a game where the aim was to play along with a music track by hitting the right buttons on a guitar-shaped controller. The game was an instant hit, spawning over 20 sequels and spin-offs over the course of around five years.

The guitar controller featured five coloured buttons representing different notes, a strumming bar that had to be hit in time with the notes being played, and a whammy bar to change the pitch of the notes. The buttons corresponded with a scrolling display of coloured circles that moved down the screen as the music played. If the player 'strummed' the guitar at the correct time while holding down the correct coloured button, they scored points and kept the (virtual) audience entertained.

For our Pygame Zero version, we don't have access to a custom-designed controller, so we'll have to use a standard computer keyboard. We can set up the letters **Z**, **X**, **C**, **V**, and **B** as the five buttons. Then we'll use the **SPACE** bar to strum the notes as they arrive at the bottom of the fretboard. We can make a list of Actors for the coloured circles which we'll then move down the screen every time `update()` is called. We have five strings numbered 0 to 4 to attach our coloured circles to, and these need to be organised so they hit the bottom target points exactly when the note plays.

We don't want the coloured circles to get out of sync with the music, so the timing of the movement is key. We must make sure that the speed is consistent, regardless of the speed of computer or other things going on which may make the frame rate change. To do this, we need to work out the time between each frame, known as the delta time, and then move the coloured circles based on that value. If you're using your own music for the backing track, you may need to play around with these values, but the way this sample is set up, the tuples in the `counterPoints` list are the string number and then the number of seconds at which the note should be played from the beginning of the song. This setup means the player can get ready to play the note as it comes down the fretboard, press the appropriate button just before it gets to the bottom target, and if they press the **SPACE** bar as the coloured circle goes over the target, we fire off a 'shine' animation to indicate they've scored points. We can give them a little leeway so that they don't have to be too precise with the **SPACE** bar press.

To make the music sample easy to play along with, I've edited a shortened version of one of my own fairly slow songs where the player plays the bassline. There are a couple of tricky bits in it, but you should be able to hit all the notes without too many attempts. You can use any MP3 or WAV file for this, just drop it in the music directory and change the line that reads `music.play_once('themoment')`. Then all you need to do is change the timings in the `counterPoints` list.

There are other things that you could add, like having error notes play if the wrong buttons are held down. There's also that whammy bar on the original controller that may require a bit of lateral thinking to replicate on a keyboard, but we'll leave that challenge to you.

Python Hero

Here's Mark's code for a *Guitar Hero*-like rhythm action game. To get it running on your system, you'll first need to install Pygame Zero. Full instructions can be found at **wfmag.cc/pgzero**.

Download the code from GitHub: **wfmag.cc/wfmag62**

```
# Guitar Hero
import pgzrun
import time

curTime = time.time()
deltaTime = 0
startTime = 0
score = 0
bps = 100.5
firstTime = True
counters = []
shine = [0,0,0,0,0]
counterPoints = [(2,9.3),(0,11.5),(3,13.8),(1,16.2),(2,18.5),(0,20.7),
                 (2,23),(0,25),(3,27.6),(1,29.8),(2,32.1),(0,34.5),
                 (4,36.6),(1,38.9),(3,41),(1,43.5),(2,44.6),(3,45.7),(0
,48),
                 (4,50.3),(1,52.5),(1,53.5),(2,53.9),(3,54.5),
                 (4,54.8),(2,57.1),(4,59.4),(2,61.7),(4,64.1),(2,66.6),
(0,69.8)
                 ]
for c in counterPoints:
    counters.append(Actor('counter'+str(c[0]), center=(300+(c[0]*50),
(c[1]-9.9)*-50)))
    counters[len(counters)-1].state = 1

def draw():
    screen.blit("background", (0, 0))
    drawCounters()
    screen.blit("fade", (0, 0))
    for s in range(0,5):
        if shine[s] > 0:
            screen.blit("shine", (230+(s*50), 450))
            shine[s] -= 1
    screen.draw.text("SCORE:"+str(score), center= (400,575), owidth=0.5,
ocolor=(255,255,255), color=(0,0,255) , fontsize=40)
    if curTime - startTime > 70:
        screen.draw.text("WELL DONE! YOU ARE A", center= (400,280),
owidth=0.5, ocolor=(255,255,255), color=(255,0,0) , fontsize=40)
        screen.draw.text("PYGAME ZERO HERO", center= (400,320),
owidth=0.5, ocolor=(255,255,255), color=(255,0,0) , fontsize=40)

def update():
    global deltaTime, curTime, firstTime, startTime
    if firstTime:
        music.play_once('themoment')
        startTime = time.time()
        firstTime = False
    deltaTime = time.time()-curTime
    curTime = time.time()
    updateCounters()

def on_key_down(key):
    if key.name == "SPACE":
        for c in counters:
            if c.y > 490 and c.y < 525:
                if c.x == 300 and keyboard.z: noteCorrect(0)
                if c.x == 350 and keyboard.x: noteCorrect(1)
                if c.x == 400 and keyboard.c: noteCorrect(2)
                if c.x == 450 and keyboard.v: noteCorrect(3)
                if c.x == 500 and keyboard.b: noteCorrect(4)

def drawCounters():
    for c in counters:
        if c.y < 520 and c.y > -20:
            c.draw()

def updateCounters():
    for c in counters:
        c.y += (bps/2)*deltaTime

def noteCorrect(column):
    global score
    shine[column] = 10
    score += 10

pgzrun.go()
```

Our rocking Pygame Zero homage to *Guitar Hero*.