
My First Pygame

Terry Howald

What is Pygame?

- Cross-platform set of Python modules designed for writing video games
 - Includes 2D computer graphics and sound libraries to be used by Python
 - Initial release October 2000, LGPL
-

Why Pygame?

- Pygame talk sparked my interest
 - Readily available tutorials and resources
 - I can do this!
-

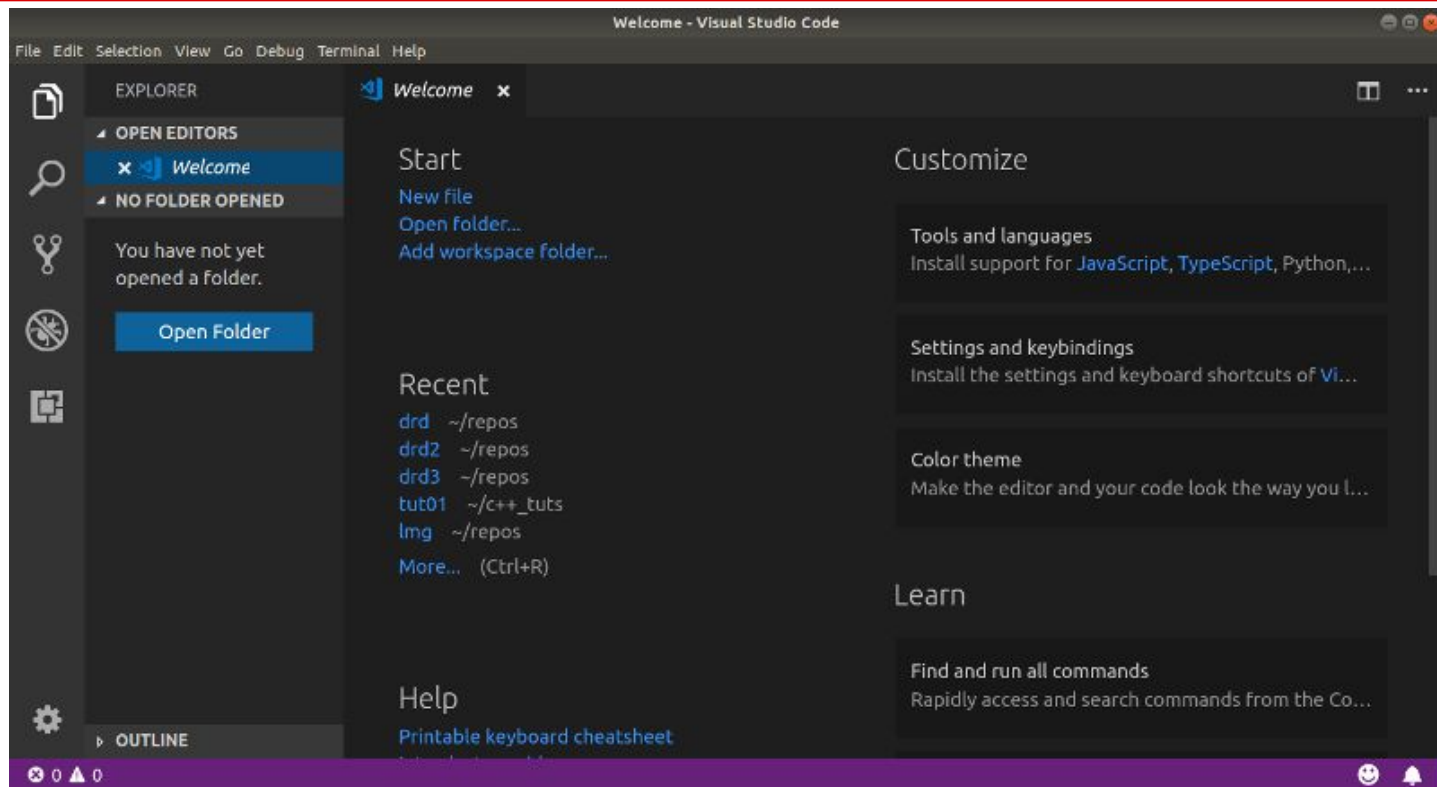
How do I get started?

- Navigate to pygame.org, GettingStarted
 - Install/Update Python 3
 - Install Pygame library
 - `python3 -m pip install -U pygame --user`
 - `sudo apt install python3-pygame`
-

Development tools

- Ubuntu 18.04
 - Python 3.6
 - Pygame 1.9.4
 - Visual Studio Code
 - Python extension for Visual Studio Code
 - Git/GitHub for source control
 - Gimp, Inkscape, Audacity
-

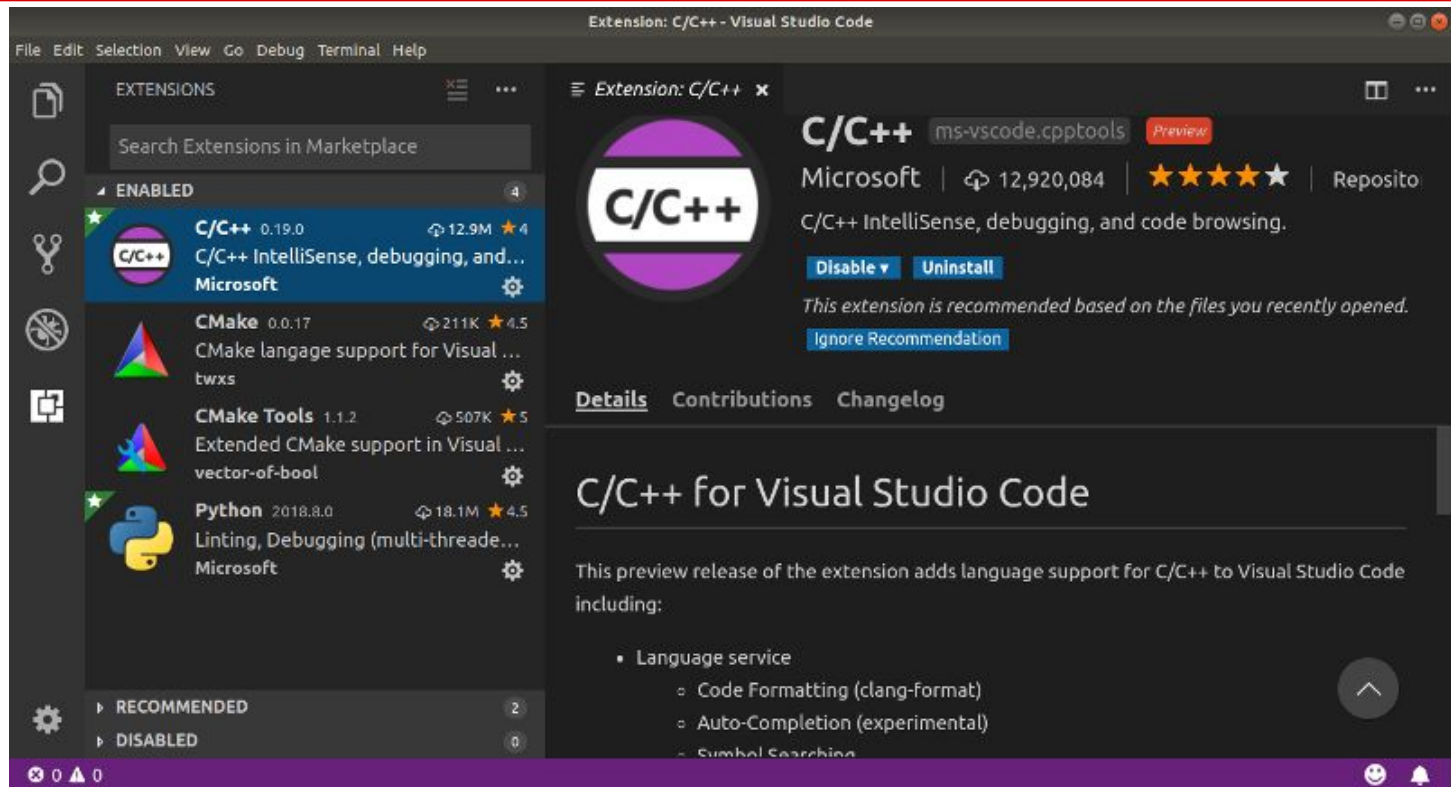
Visual Studio Code



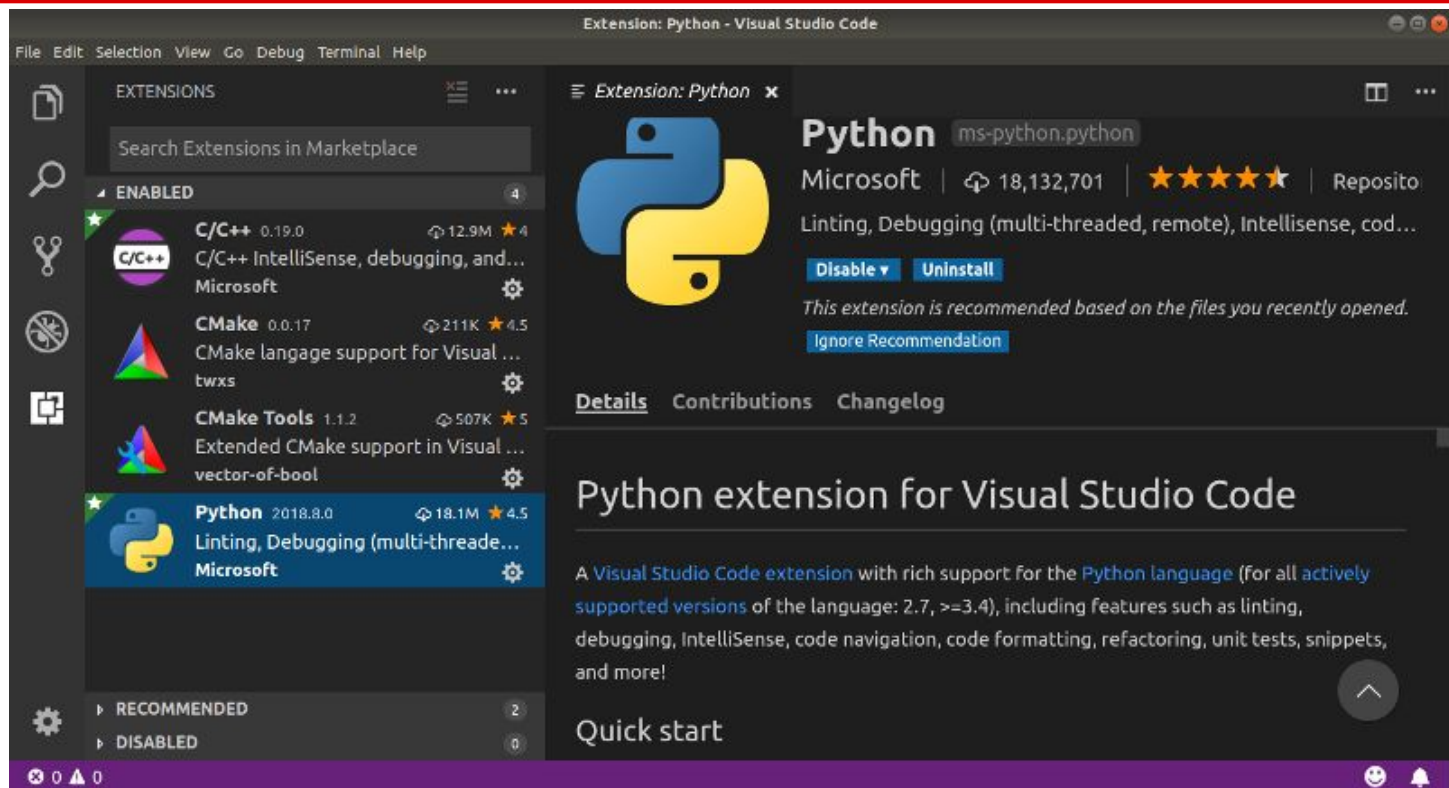
Gone are the days...



C++ extension



Python extension



Game idea 1.0



Game idea 1.0



Game idea 1.0



Game idea 1.0

- Shuttle stranded on planet
 - 7 crew members
 - 5 phasers on shuttle
 - X creatures with spears and boulders
 - Crew must discharge phasers and lift off before shuttle is destroyed by creatures
-

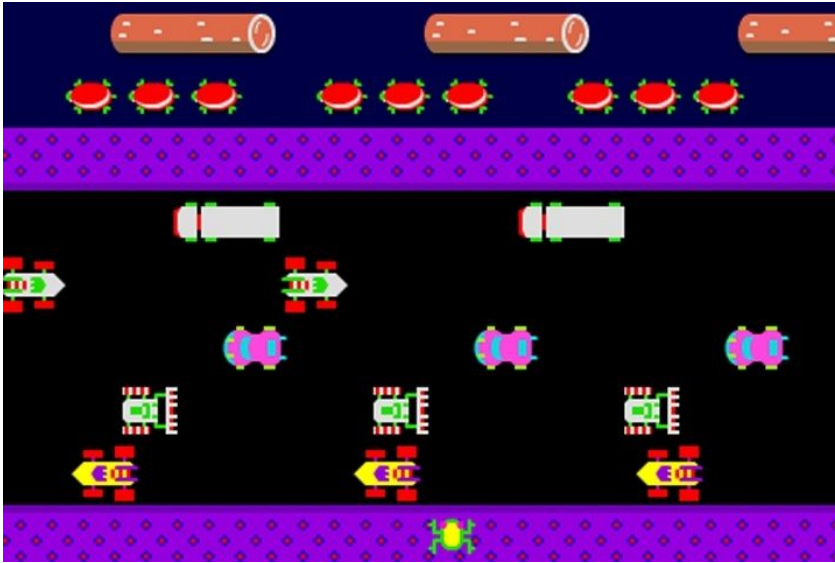
Let's get started 1.0!

Types of 2D games

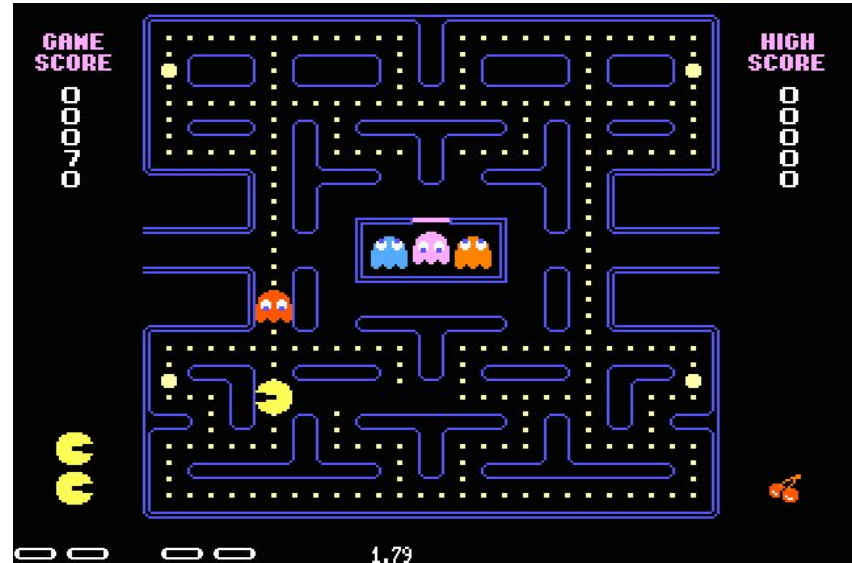
- Single Screen - Frogger, Pac-Man, Dig Dug
 - Scrollers - Gun Smoke, Heavy Barrel
 - Side Scrollers - Super Mario Brothers
 - Platform Games - Donkey Kong, Mario Bros
 - Adventure - The Legend of Zelda
-

Focus on a single screen game

Frogger

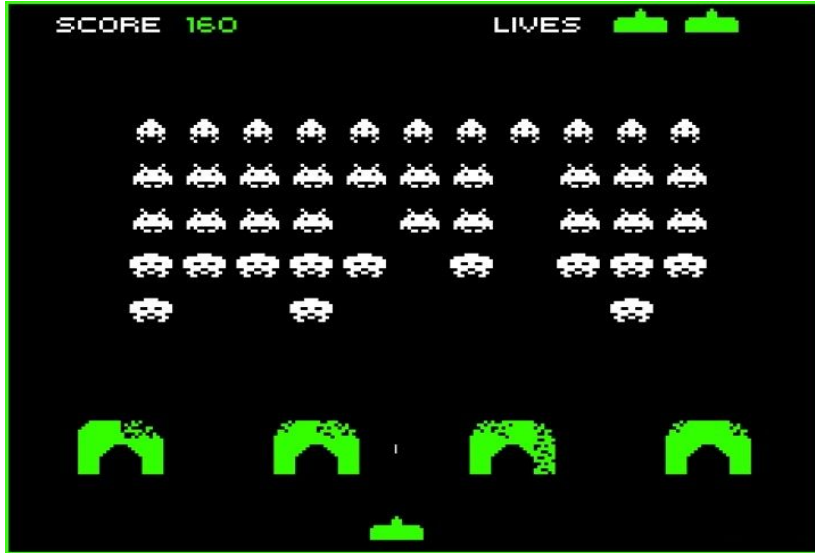


Pac-Man



Focus on a single screen game

Space Invaders



Dig Dug



Game idea 2.0



Game idea 2.0



Game idea 2.0



Game idea 2.0

- Single screen has tunnels in rock
 - Security team patrols the tunnels
 - Creature is able to tunnel through the rock
 - Creature's eggs are all over the tunnels
 - Security team destroys the eggs
 - Creature must destroy security team before all the eggs are crushed
-

Let's get started 2.0!



Let's get started 2.0!

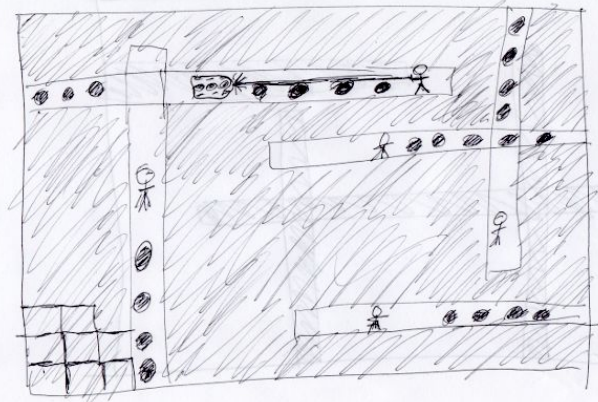


Die, Redshirt, Die!

Game Elements



Game Screen



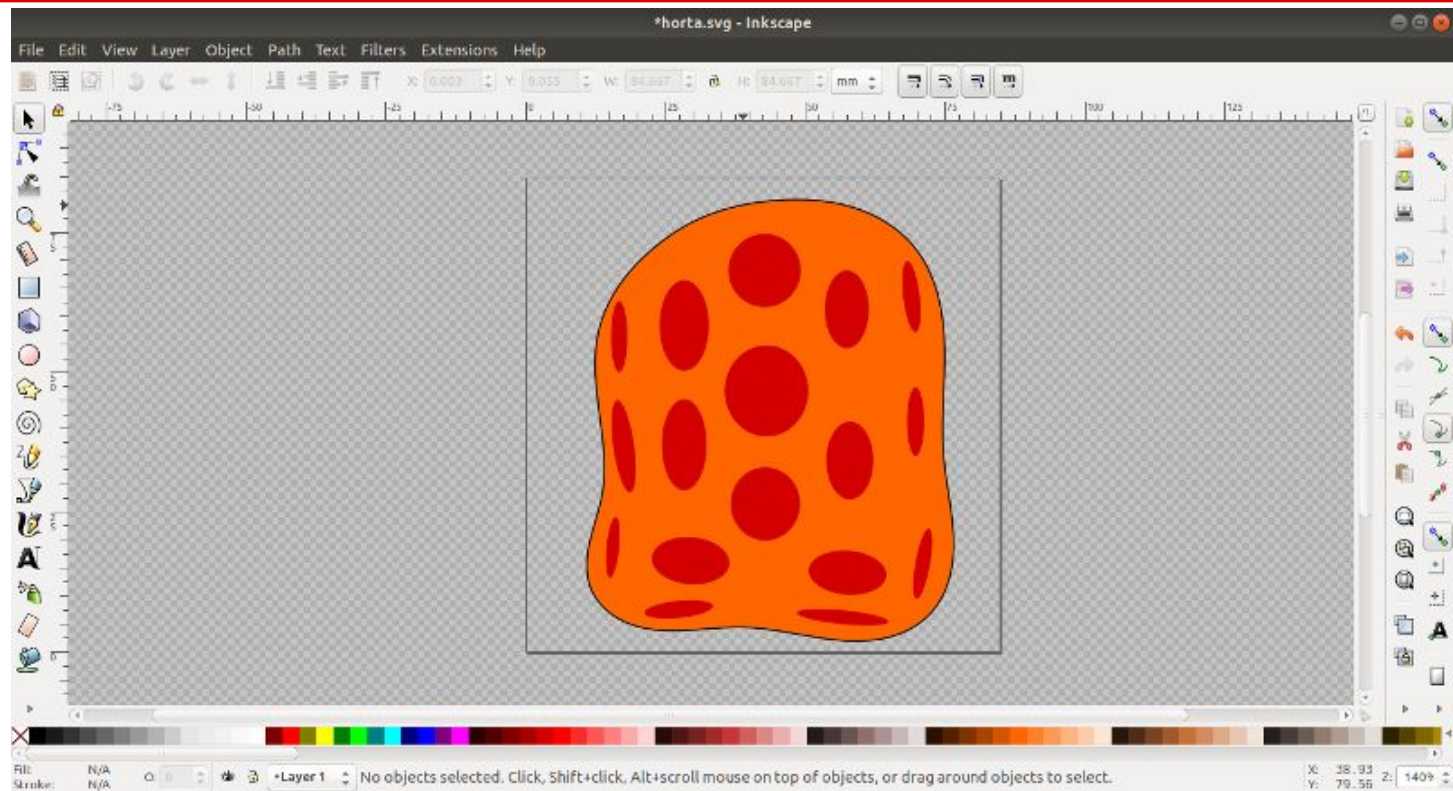
Tile-based game

- Game consists of small square graphic images (tiles) laid out in a grid
 - The grid of tiles makes up the screen
 - Tile size of 32 x 32 pixels
 - Screen size 800 x 640 pixels
 - Grid size 25 x 20
-

Create tiles in Inkscape

- Player
 - Enemy
 - Eggs
 - Rock
-

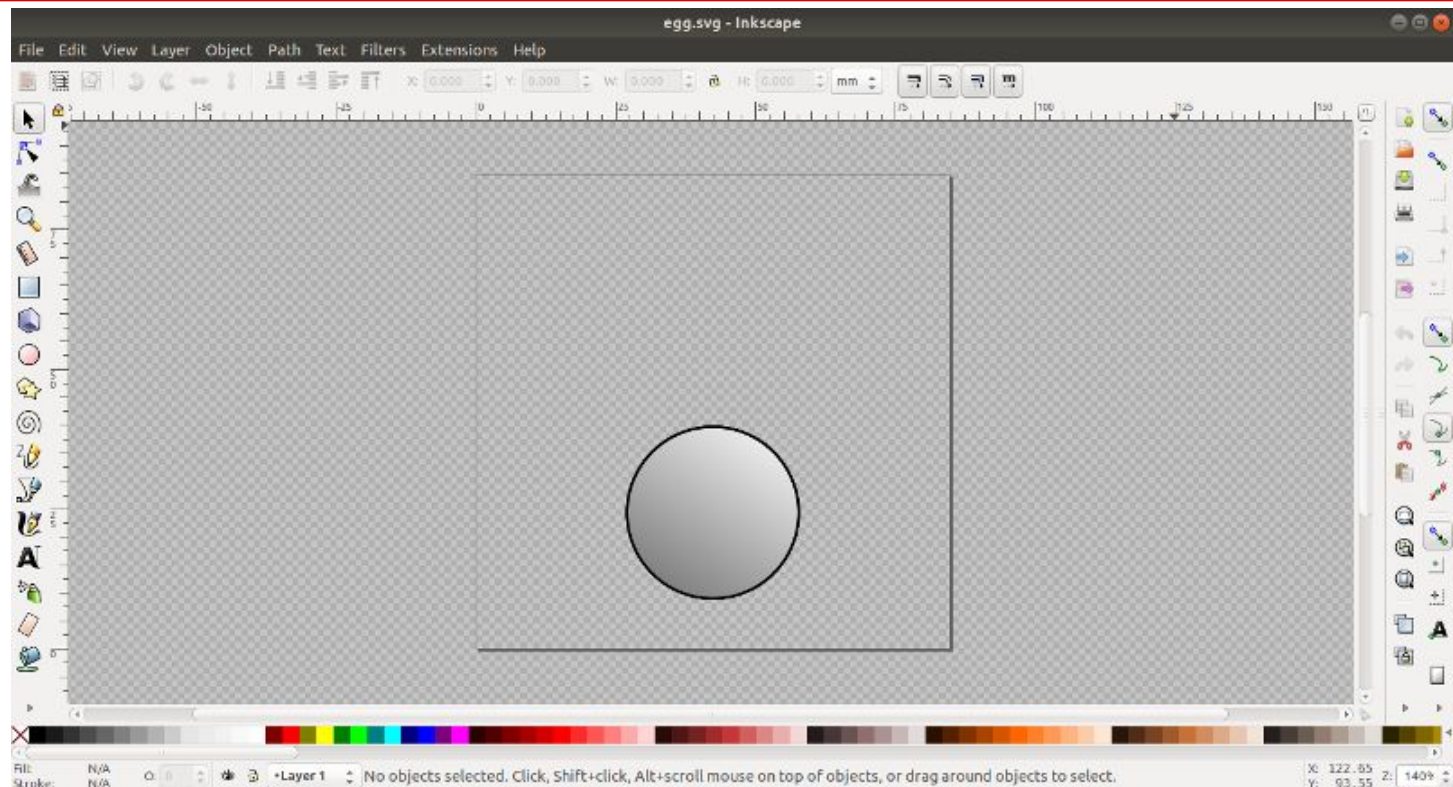
Player tile



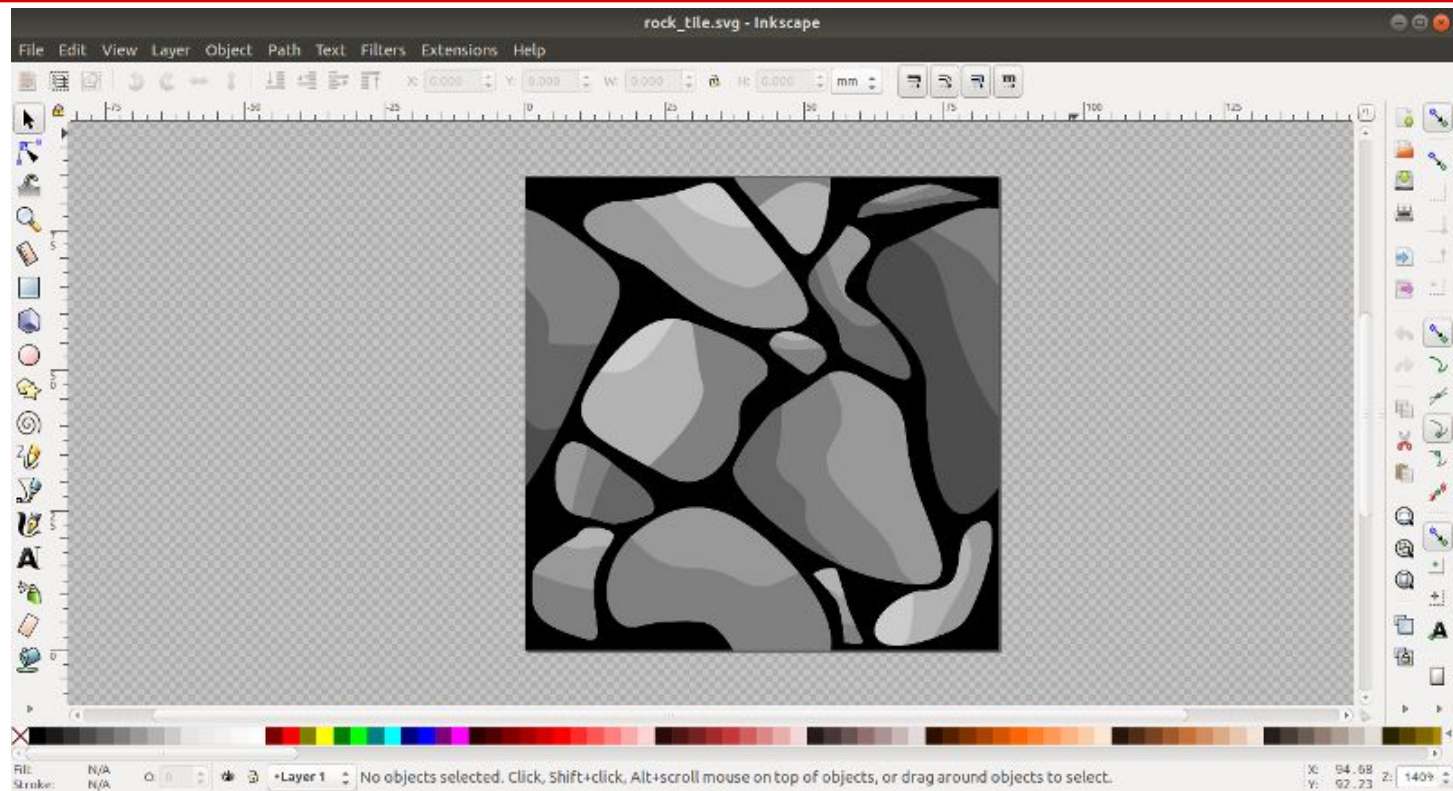
Enemy tile



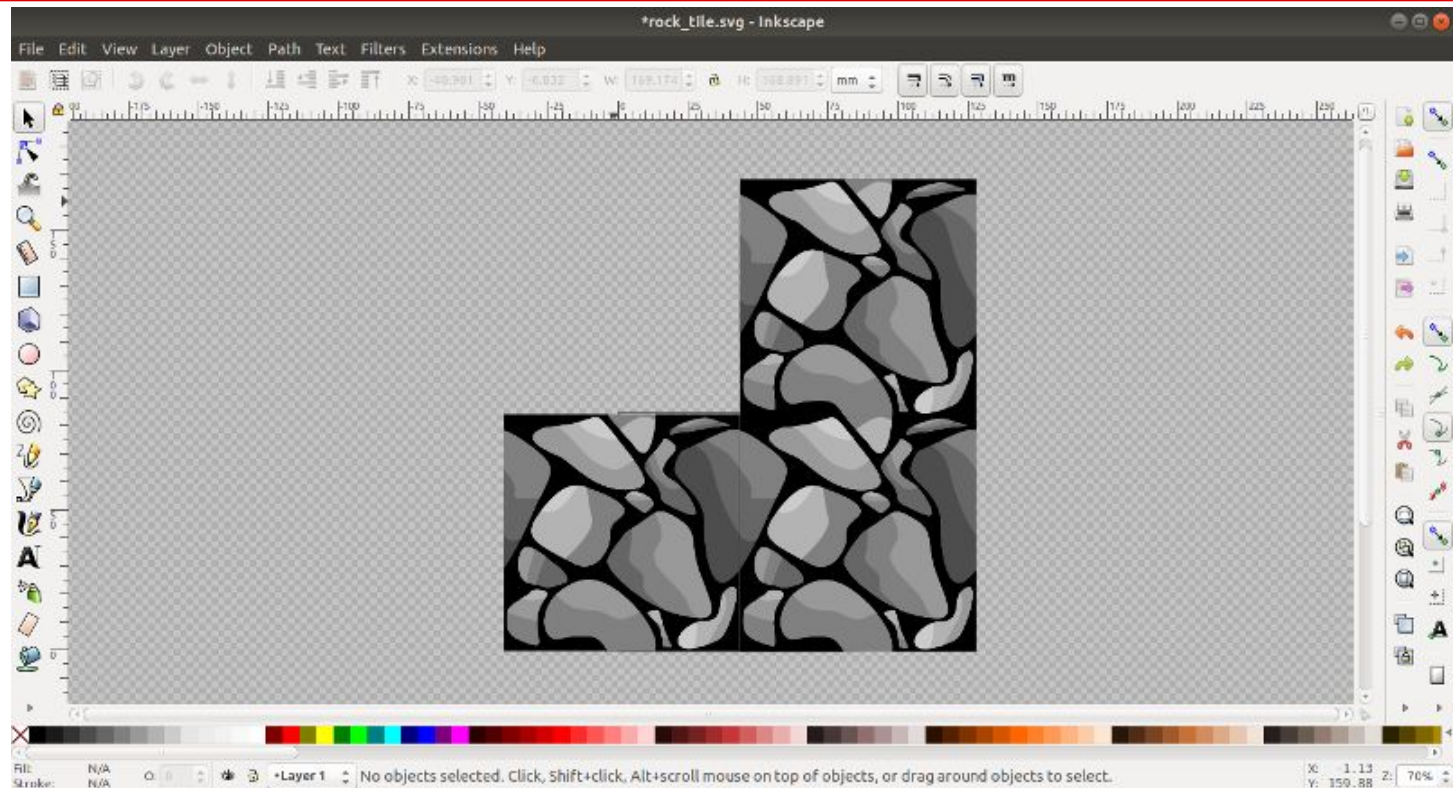
Egg tile



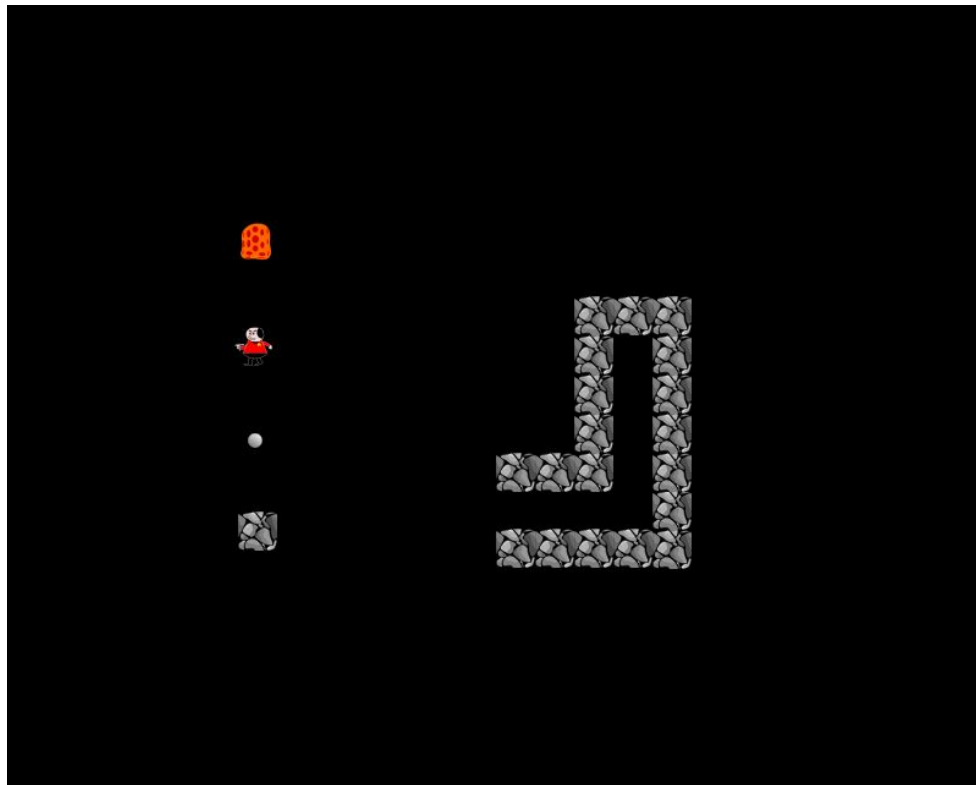
Rock tile



Seamless repeating tiles



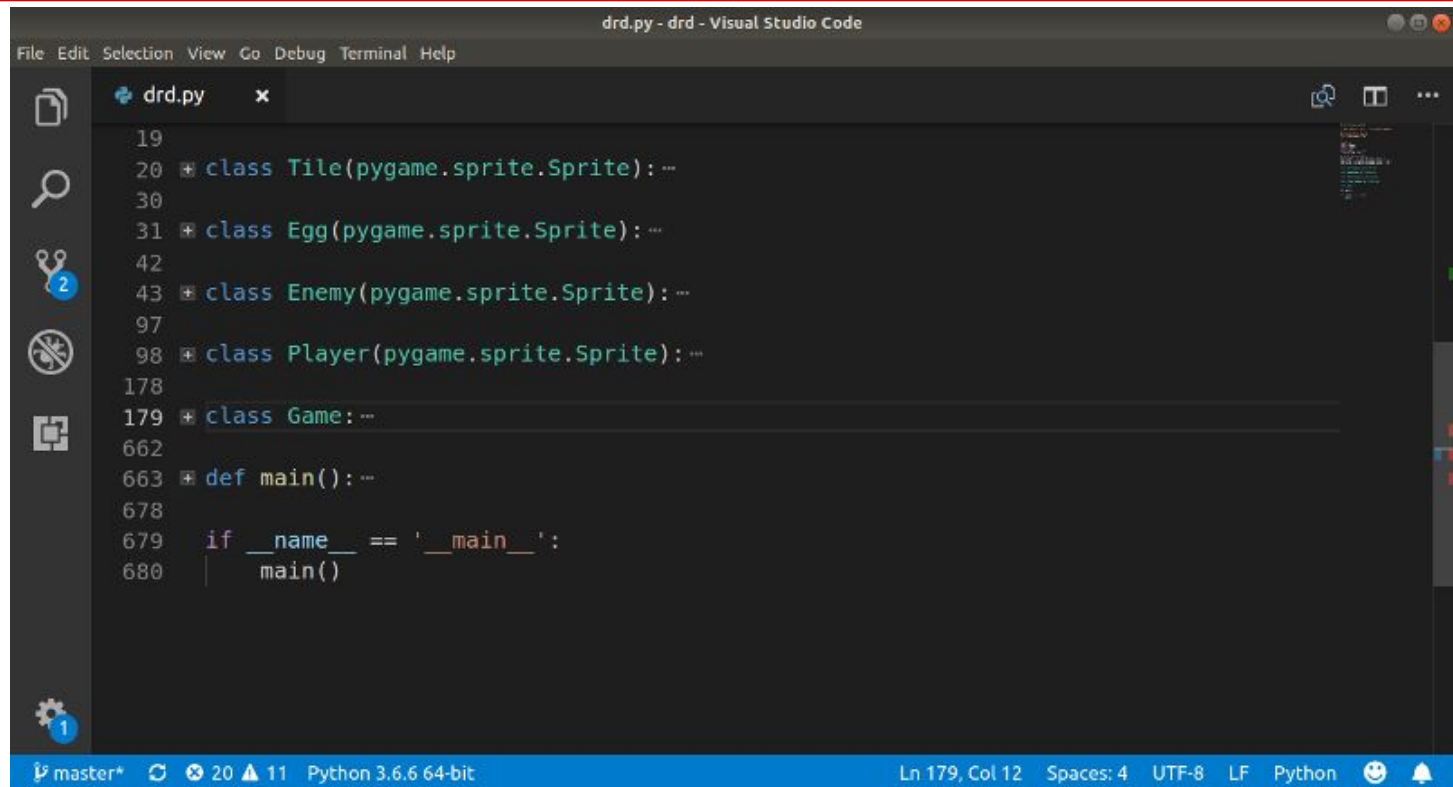
Tiles test



Manage tiles with Pygame sprites

- Sprite - a two dimensional image
 - Game objects can be sprite objects
 - Sprites can draw and update themselves
 - Sprites can be added to groups
 - Sprite groups for drawing and collision detection
-

Pygame sprites

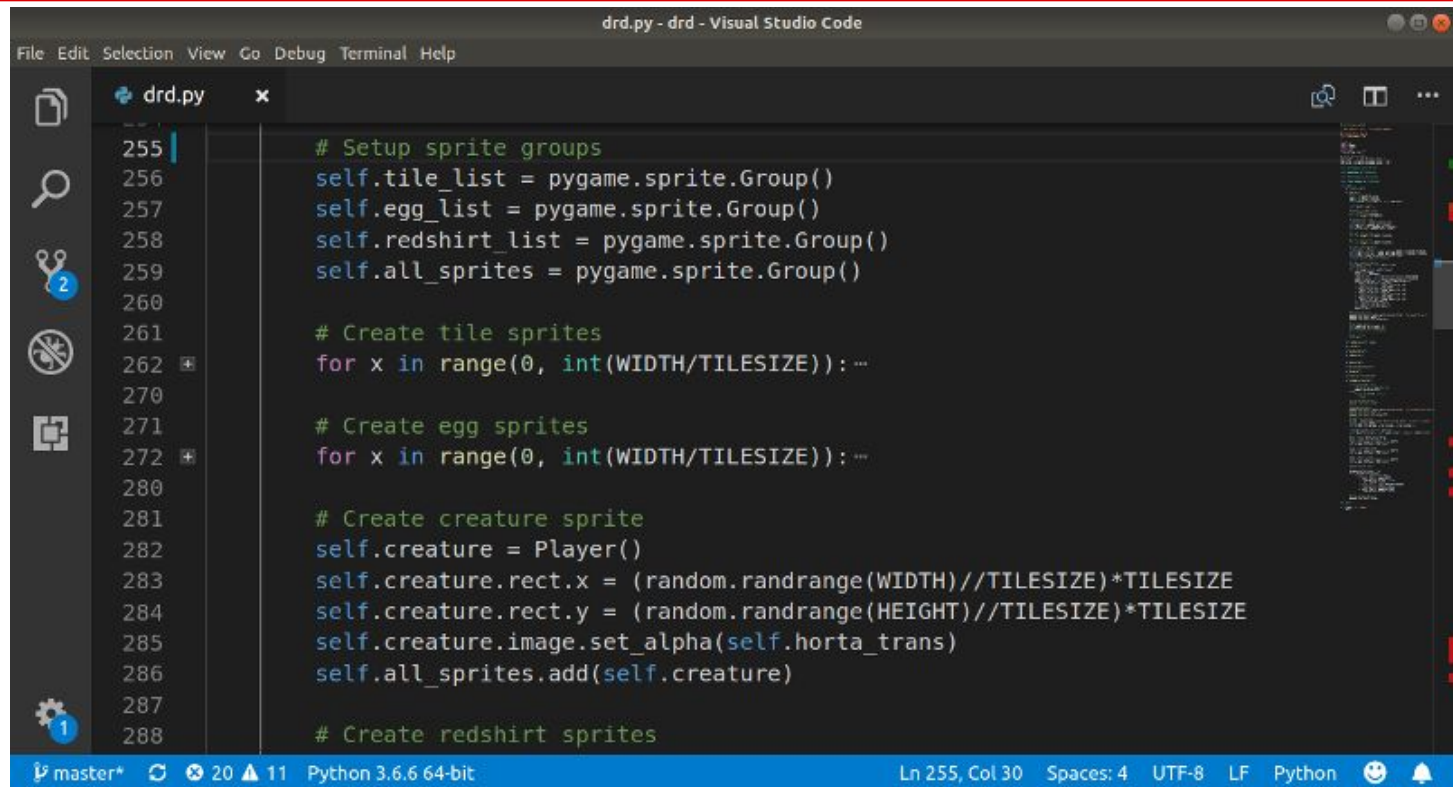


```
drd.py - drd - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

19
20 class Tile(pygame.sprite.Sprite): ...
30
31 class Egg(pygame.sprite.Sprite): ...
42
43 class Enemy(pygame.sprite.Sprite): ...
97
98 class Player(pygame.sprite.Sprite): ...
178
179 class Game: ...
662
663 def main(): ...
678
679 if __name__ == '__main__':
680     main()

Python 3.6.6 64-bit
```

Pygame sprite groups

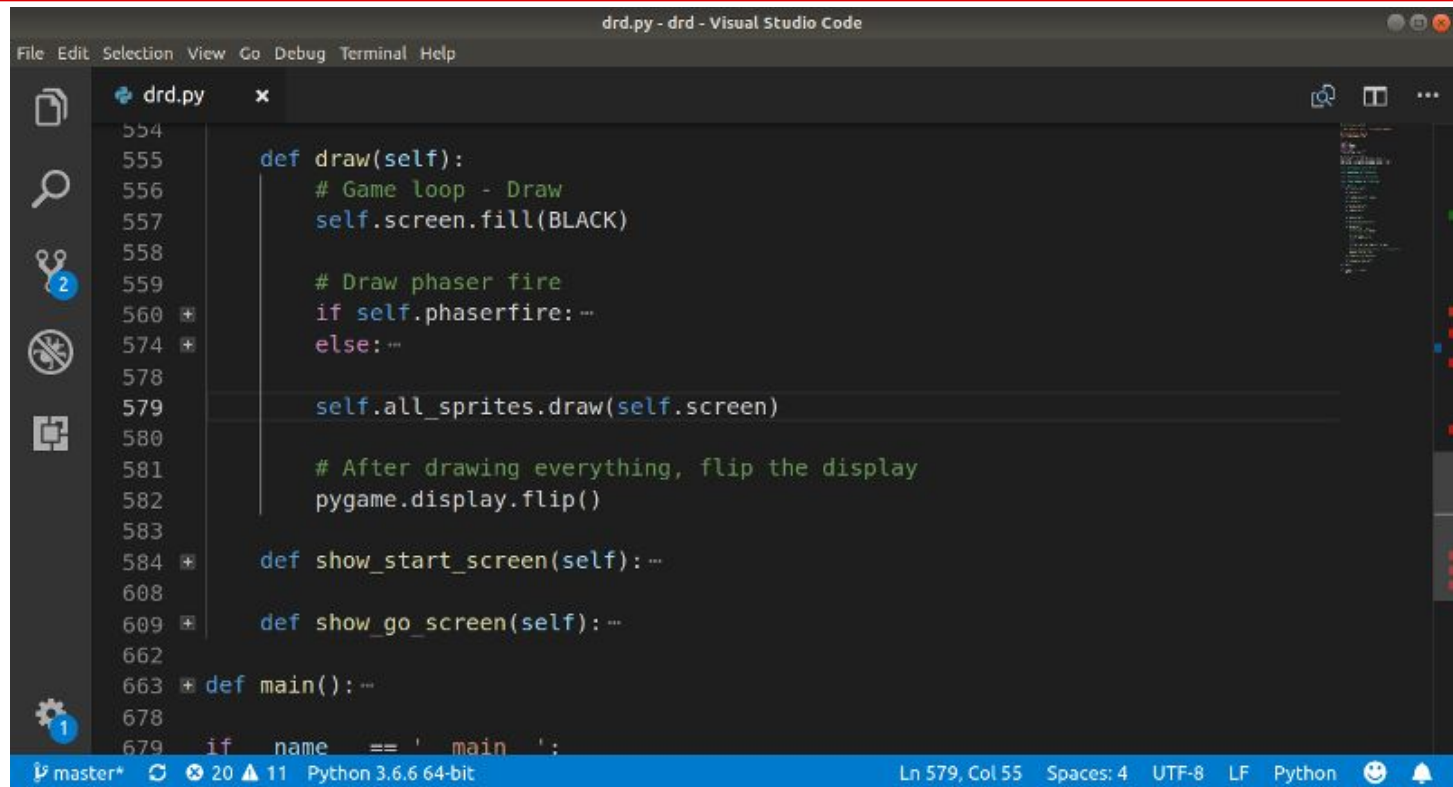


```
drd.py - drd - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

drd.py x
255 | # Setup sprite groups
256 | self.tile_list = pygame.sprite.Group()
257 | self.egg_list = pygame.sprite.Group()
258 | self.redshirt_list = pygame.sprite.Group()
259 | self.all_sprites = pygame.sprite.Group()
260 |
261 | # Create tile sprites
262 | for x in range(0, int(WIDTH/TILESIZE)): ...
270 |
271 | # Create egg sprites
272 | for x in range(0, int(WIDTH/TILESIZE)): ...
280 |
281 | # Create creature sprite
282 | self.creature = Player()
283 | self.creature.rect.x = (random.randrange(WIDTH)//TILESIZE)*TILESIZE
284 | self.creature.rect.y = (random.randrange(HEIGHT)//TILESIZE)*TILESIZE
285 | self.creature.image.set_alpha(self.horta_trans)
286 | self.all_sprites.add(self.creature)
287 |
288 | # Create redshirt sprites
```

master* 20 11 Python 3.6.6 64-bit Ln 255, Col 30 Spaces: 4 UTF-8 LF Python

Sprites group draw

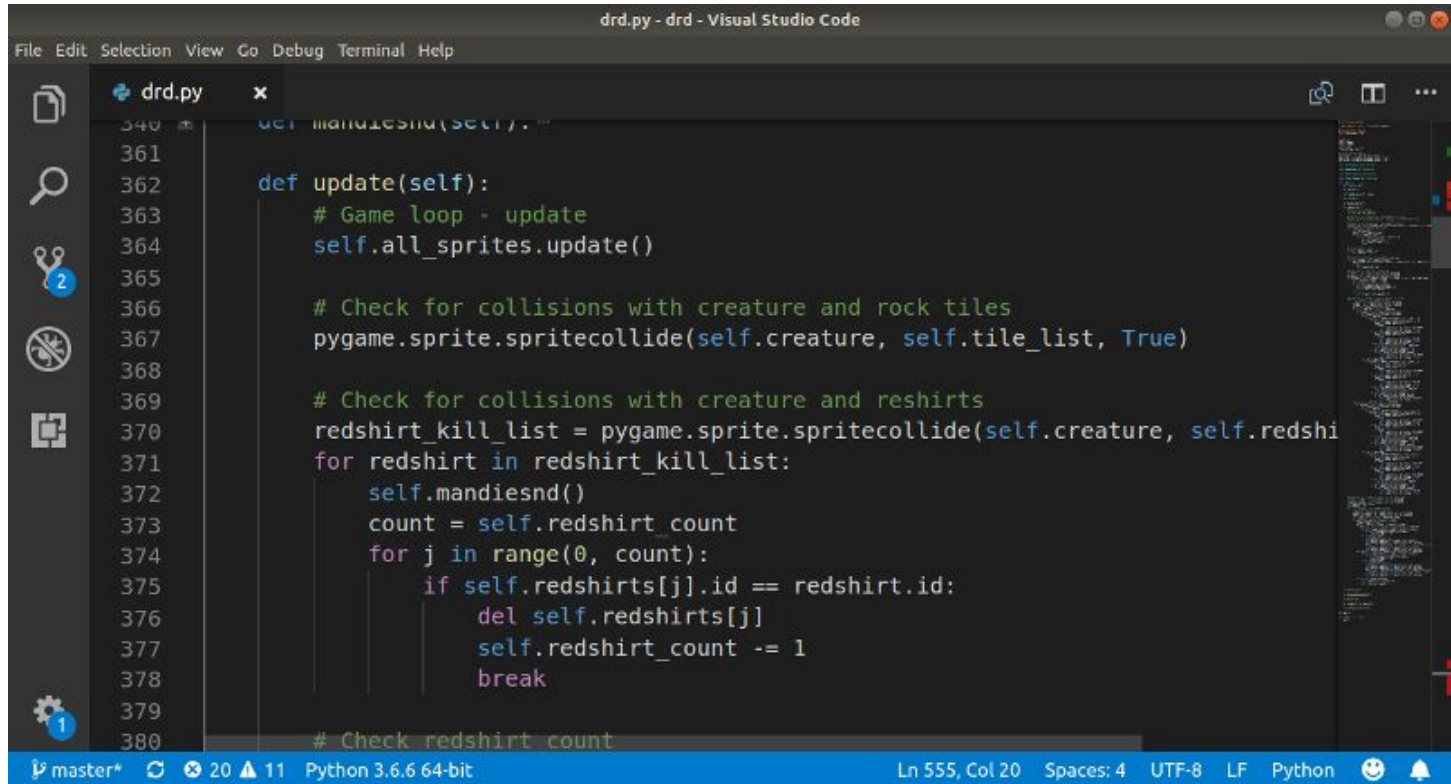


```
drd.py - drd - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

drd.py x
554
555     def draw(self):
556         # Game loop - Draw
557         self.screen.fill(BLACK)
558
559         # Draw phaser fire
560         if self.phaserfire: ...
574         else: ...
578
579         self.all_sprites.draw(self.screen)
580
581         # After drawing everything, flip the display
582         pygame.display.flip()
583
584     def show_start_screen(self): ...
608
609     def show_go_screen(self): ...
662
663     def main(): ...
678
679     if name == 'main':
```

Ln 579, Col 55 Spaces: 4 UTF-8 LF Python

Sprites group collision detection



The screenshot shows a Visual Studio Code editor window titled "drd.py - drd - Visual Studio Code". The editor displays a Python script with the following code:

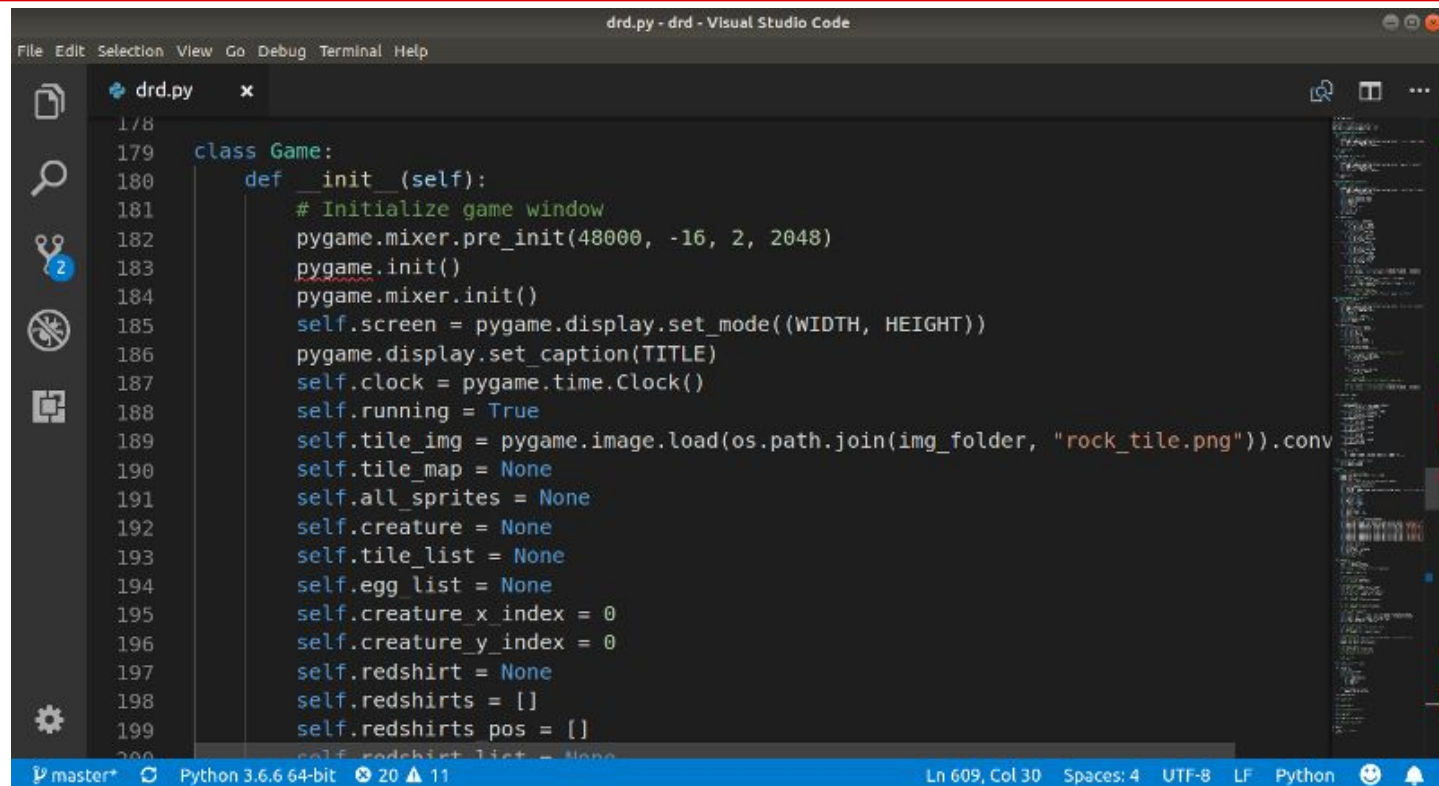
```
340 def mandiesnd(self):
361
362 def update(self):
363     # Game loop - update
364     self.all_sprites.update()
365
366     # Check for collisions with creature and rock tiles
367     pygame.sprite.spritecollide(self.creature, self.tile_list, True)
368
369     # Check for collisions with creature and reshirts
370     redshirt_kill_list = pygame.sprite.spritecollide(self.creature, self.redshirts, True)
371     for redshirt in redshirt_kill_list:
372         self.mandiesnd()
373         count = self.redshirt_count
374         for j in range(0, count):
375             if self.redshirts[j].id == redshirt.id:
376                 del self.redshirts[j]
377                 self.redshirt_count -= 1
378                 break
379
380     # Check redshirt count
```

The status bar at the bottom indicates the file is "master*", the cursor is at line 20, column 11, and the file is encoded in UTF-8 with LF line endings. The editor is running Python 3.6.6 64-bit.

Game sounds

- Background sounds
 - Game start screen
 - Game run screen
 - Game over screen
 - Sound effects
 - Eggs squished
 - Phaser fire
 - Redshirts scream
-

Pygame mixer

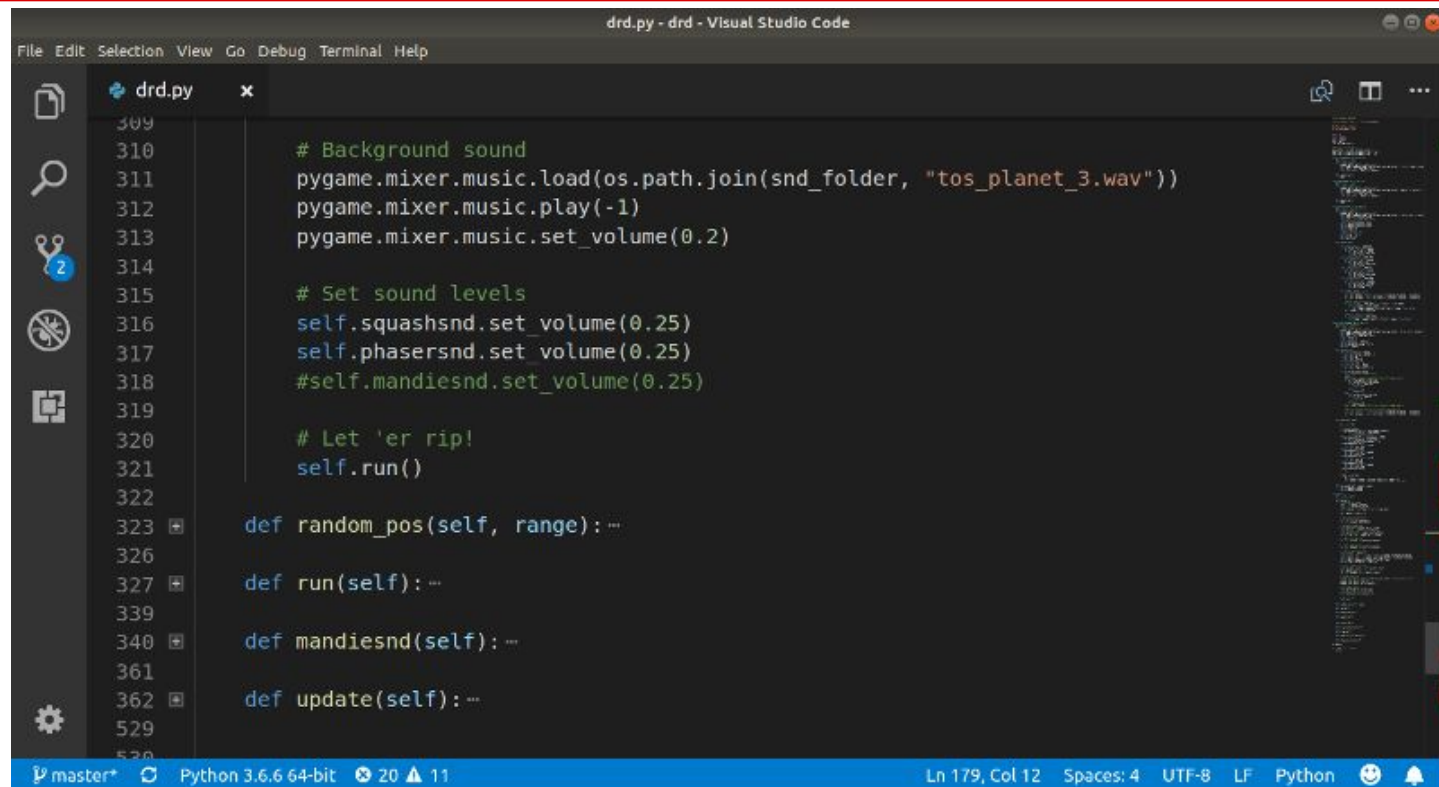


```
drd.py - drd - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

drd.py x
178
179 class Game:
180     def __init__(self):
181         # Initialize game window
182         pygame.mixer.pre_init(48000, -16, 2, 2048)
183         pygame.init()
184         pygame.mixer.init()
185         self.screen = pygame.display.set_mode((WIDTH, HEIGHT))
186         pygame.display.set_caption(TITLE)
187         self.clock = pygame.time.Clock()
188         self.running = True
189         self.tile_img = pygame.image.load(os.path.join(img_folder, "rock_tile.png")).conv
190         self.tile_map = None
191         self.all_sprites = None
192         self.creature = None
193         self.tile_list = None
194         self.egg_list = None
195         self.creature_x_index = 0
196         self.creature_y_index = 0
197         self.redshirt = None
198         self.redshirts = []
199         self.redshirts_pos = []
200         self.redshirt_list = None
```

Python 3.6.6 64-bit 20 11 Ln 609, Col 30 Spaces: 4 UTF-8 LF Python

Pygame background sounds

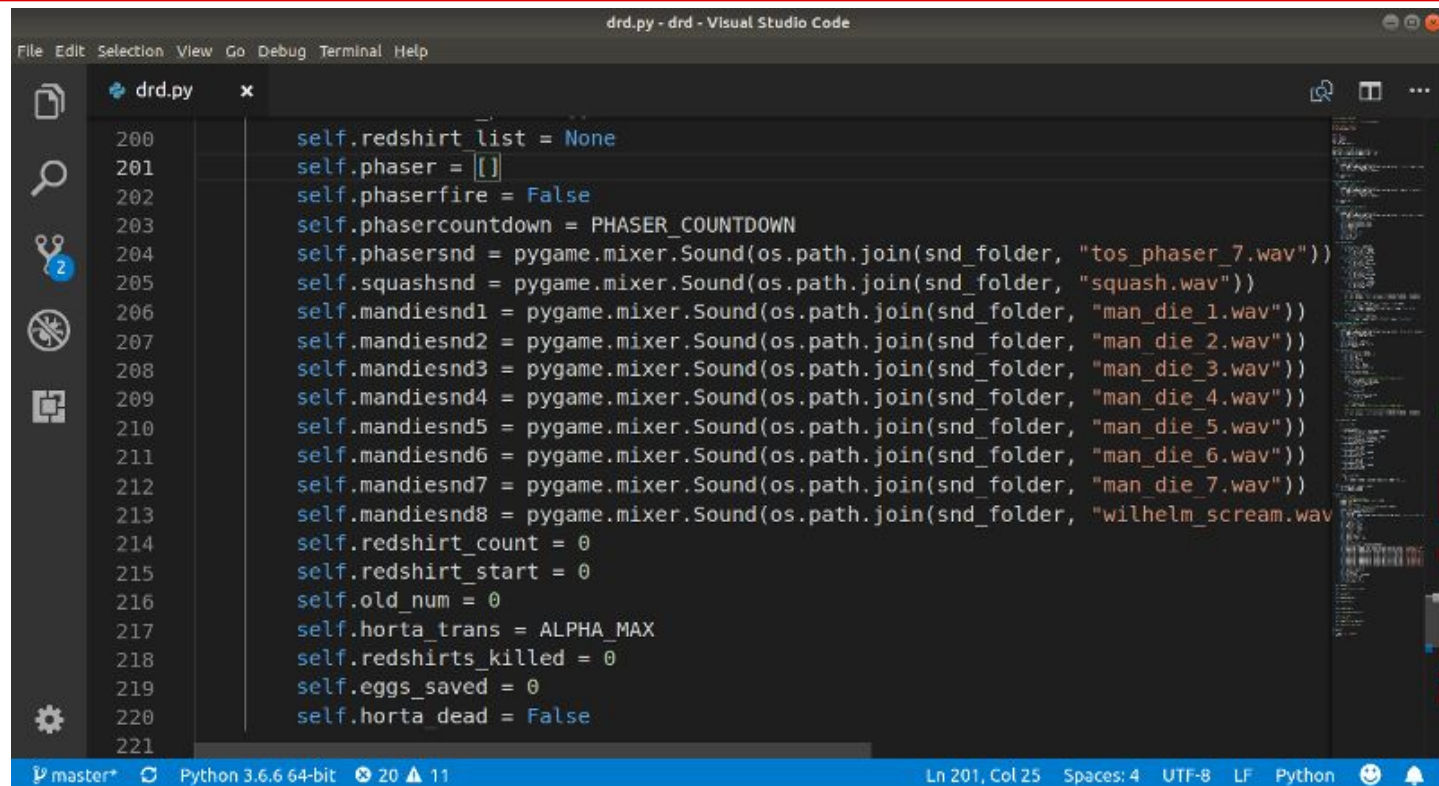


The screenshot shows a Visual Studio Code editor window titled "drd.py - drd - Visual Studio Code". The editor is displaying a Python script with the following code:

```
309
310     # Background sound
311     pygame.mixer.music.load(os.path.join(snd_folder, "tos_planet_3.wav"))
312     pygame.mixer.music.play(-1)
313     pygame.mixer.music.set_volume(0.2)
314
315     # Set sound levels
316     self.squashsnd.set_volume(0.25)
317     self.phasersnd.set_volume(0.25)
318     #self.mandiesnd.set_volume(0.25)
319
320     # Let 'er rip!
321     self.run()
322
323     def random_pos(self, range): ...
326
327     def run(self): ...
339
340     def mandiesnd(self): ...
361
362     def update(self): ...
529
530
```

The status bar at the bottom indicates the file is on the "master" branch, using Python 3.6.6 64-bit, with 20 lines and 11 columns. The cursor is at line 179, column 12. The editor also shows the file explorer on the left and the search bar on the right.

Pygame sound effects - load

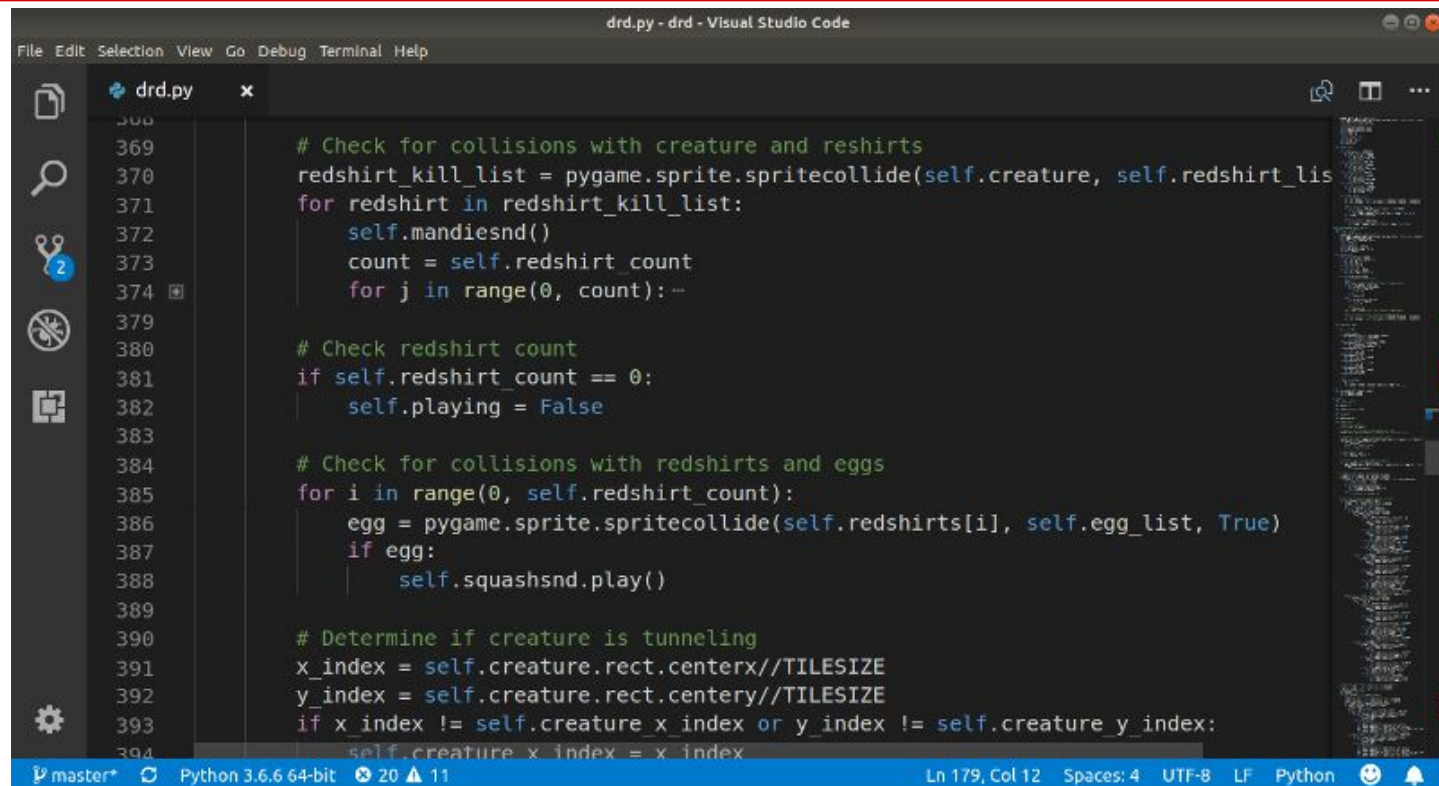


```
drd.py - drd - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

drd.py x
200 self.redshirt_list = None
201 self.phaser = []
202 self.phaserfire = False
203 self.phasercountdown = PHASER_COUNTDOWN
204 self.phasersnd = pygame.mixer.Sound(os.path.join(snd_folder, "tos_phaser_7.wav"))
205 self.squashesnd = pygame.mixer.Sound(os.path.join(snd_folder, "squash.wav"))
206 self.mandiesnd1 = pygame.mixer.Sound(os.path.join(snd_folder, "man_die_1.wav"))
207 self.mandiesnd2 = pygame.mixer.Sound(os.path.join(snd_folder, "man_die_2.wav"))
208 self.mandiesnd3 = pygame.mixer.Sound(os.path.join(snd_folder, "man_die_3.wav"))
209 self.mandiesnd4 = pygame.mixer.Sound(os.path.join(snd_folder, "man_die_4.wav"))
210 self.mandiesnd5 = pygame.mixer.Sound(os.path.join(snd_folder, "man_die_5.wav"))
211 self.mandiesnd6 = pygame.mixer.Sound(os.path.join(snd_folder, "man_die_6.wav"))
212 self.mandiesnd7 = pygame.mixer.Sound(os.path.join(snd_folder, "man_die_7.wav"))
213 self.mandiesnd8 = pygame.mixer.Sound(os.path.join(snd_folder, "wilhelm_scream.wav"))
214 self.redshirt_count = 0
215 self.redshirt_start = 0
216 self.old_num = 0
217 self.horta_trans = ALPHA_MAX
218 self.redshirts_killed = 0
219 self.eggs_saved = 0
220 self.horta_dead = False
221
```

master Python 3.6.6 64-bit 20 11 Ln 201, Col 25 Spaces: 4 UTF-8 LF Python

Pygame sound effects - play



```
drd.py - drd - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

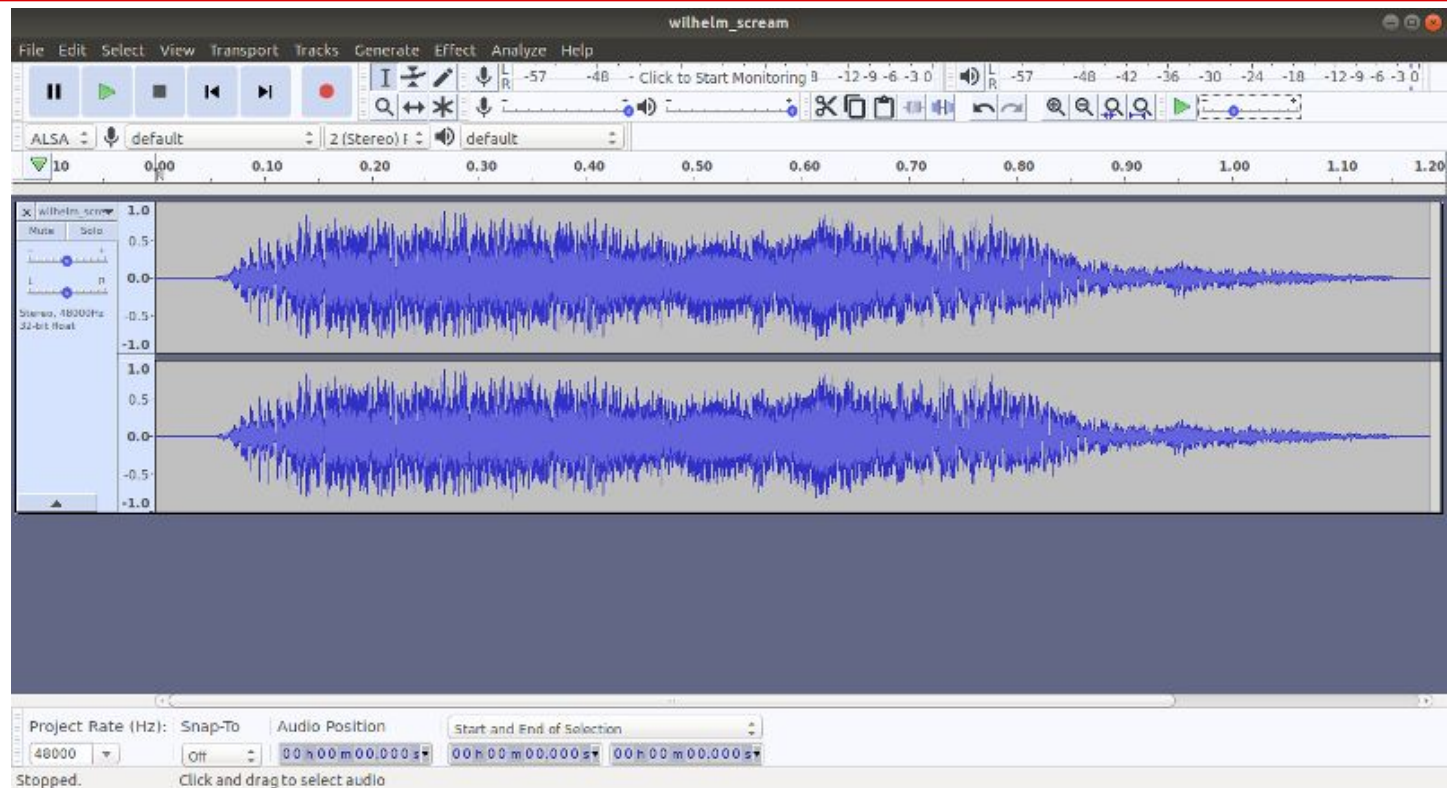
drd.py x
368
369     # Check for collisions with creature and reshirts
370     redshirt_kill_list = pygame.sprite.spritecollide(self.creature, self.redshirt_list, True)
371     for redshirt in redshirt_kill_list:
372         self.mandiesnd()
373         count = self.redshirt_count
374         for j in range(0, count):
375
376
377     # Check redshirt count
378     if self.redshirt_count == 0:
379         self.playing = False
380
381     # Check for collisions with redshirts and eggs
382     for i in range(0, self.redshirt_count):
383         egg = pygame.sprite.spritecollide(self.redshirts[i], self.egg_list, True)
384         if egg:
385             self.squashsnd.play()
386
387     # Determine if creature is tunneling
388     x_index = self.creature.rect.centerx//TILESIZE
389     y_index = self.creature.rect.centery//TILESIZE
390     if x_index != self.creature_x_index or y_index != self.creature_y_index:
391         self.creature_x_index = x_index
392         self.creature_y_index = y_index
```

master* Python 3.6.6 64-bit 20 11 Ln 179, Col 12 Spaces: 4 UTF-8 LF Python

Sounds come from?

- trekcore.com
 - freesound.org
 - Pygame mixer expects a specific file format
 - Pygame mixes sets the frequency, bit depth
 - Sound files were all different formats, frequencies, bit depth, etc
-

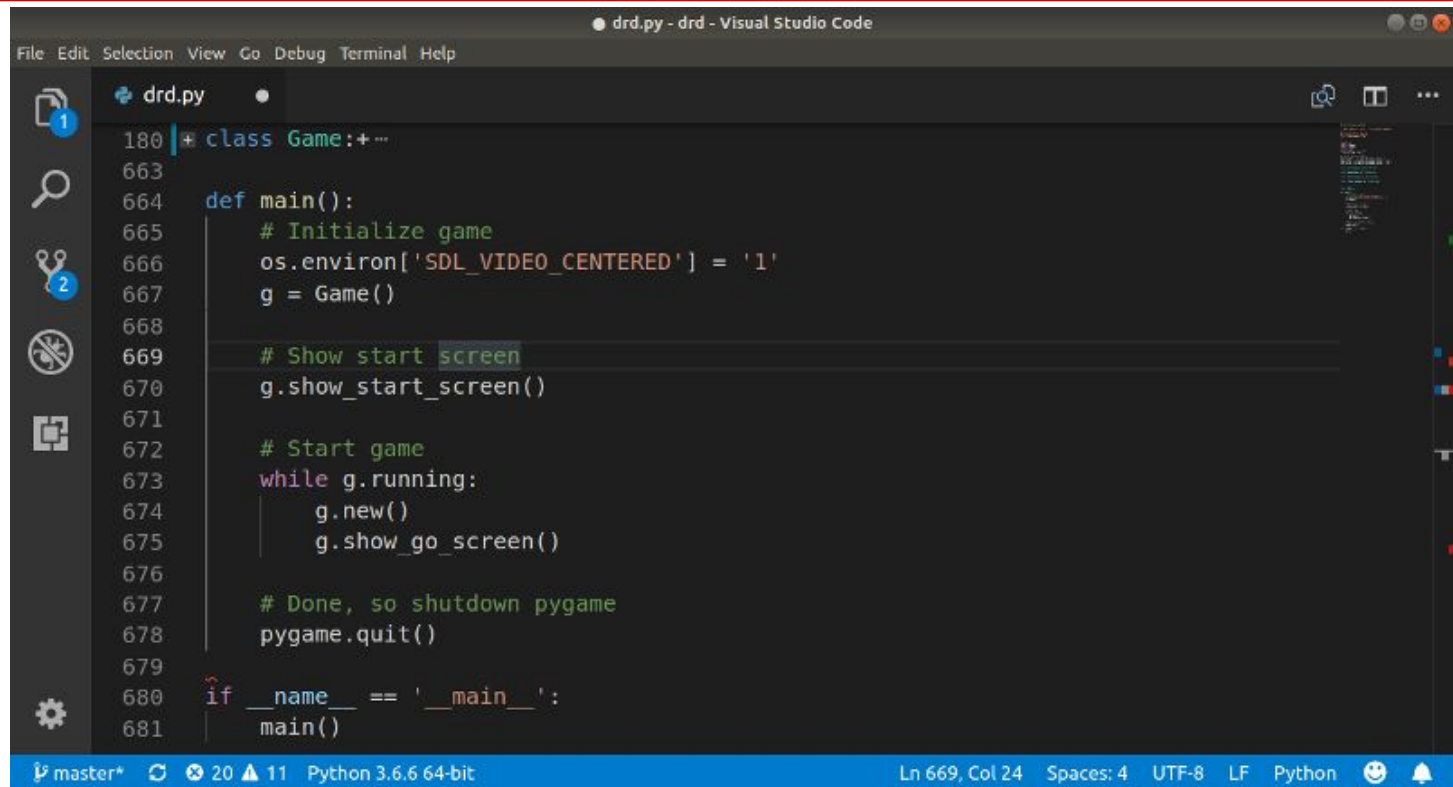
Audacity to the rescue



General program flow

- Initialize game library
 - Load game elements
 - Run game loop
 - Manage frame rate
 - Check events
 - Update logic
 - Draw screen
-

Game start

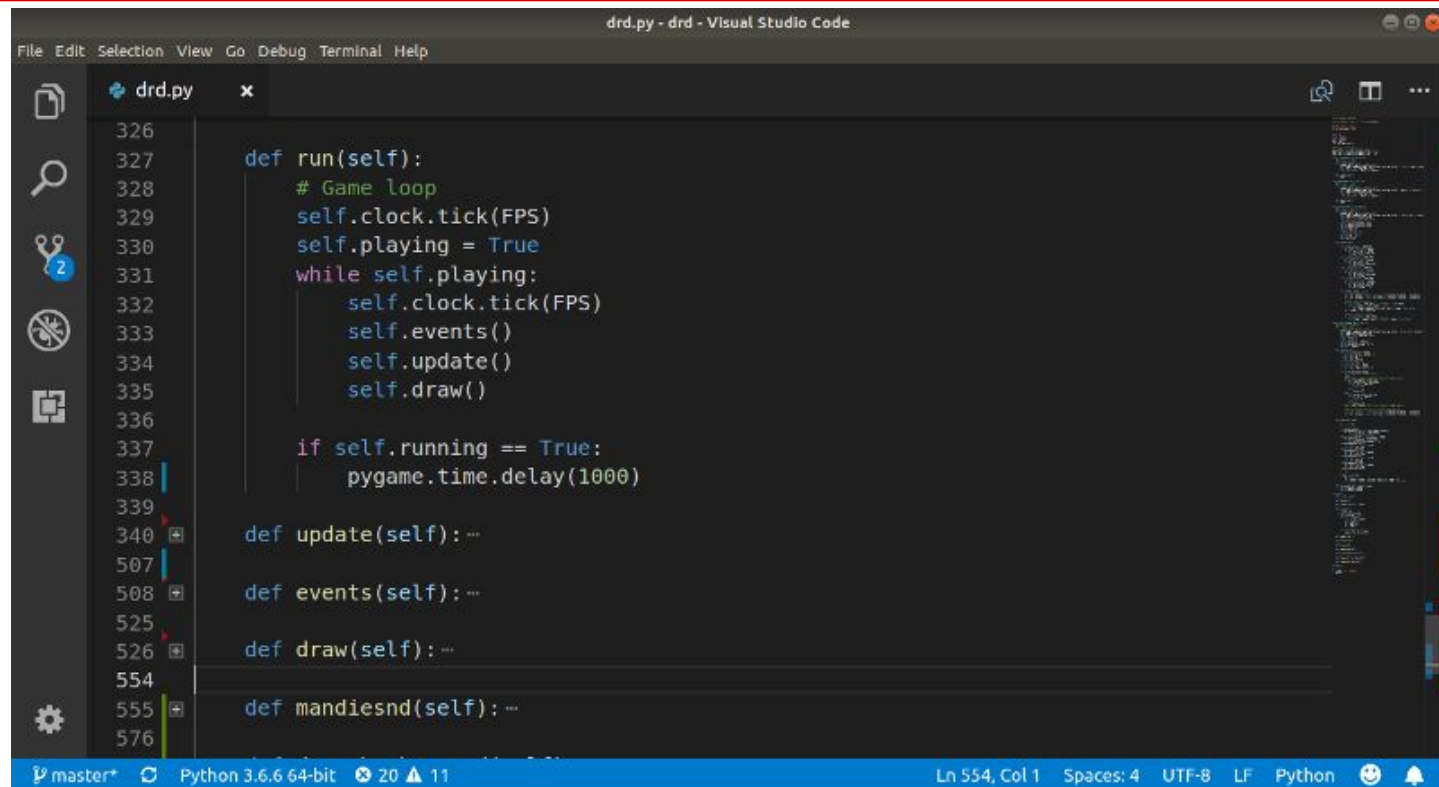


```
drd.py - drd - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

drd.py
180 class Game:
663
664 def main():
665     # Initialize game
666     os.environ['SDL_VIDEO_CENTERED'] = '1'
667     g = Game()
668
669     # Show start screen
670     g.show_start_screen()
671
672     # Start game
673     while g.running:
674         g.new()
675         g.show_go_screen()
676
677     # Done, so shutdown pygame
678     pygame.quit()
679
680 if __name__ == '__main__':
681     main()
```

master* 20 11 Python 3.6.6 64-bit Ln 669, Col 24 Spaces: 4 UTF-8 LF Python

Run loop

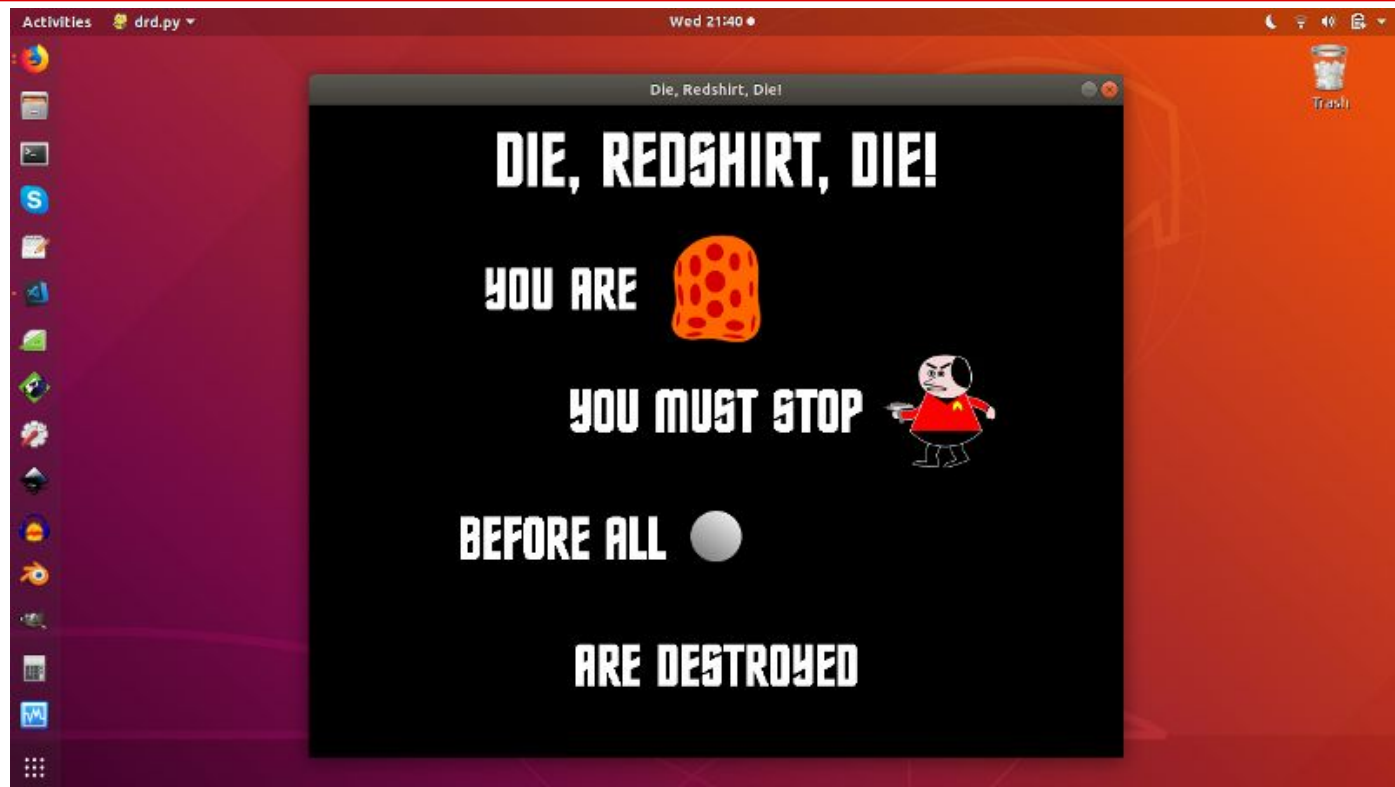


```
drd.py - drd - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

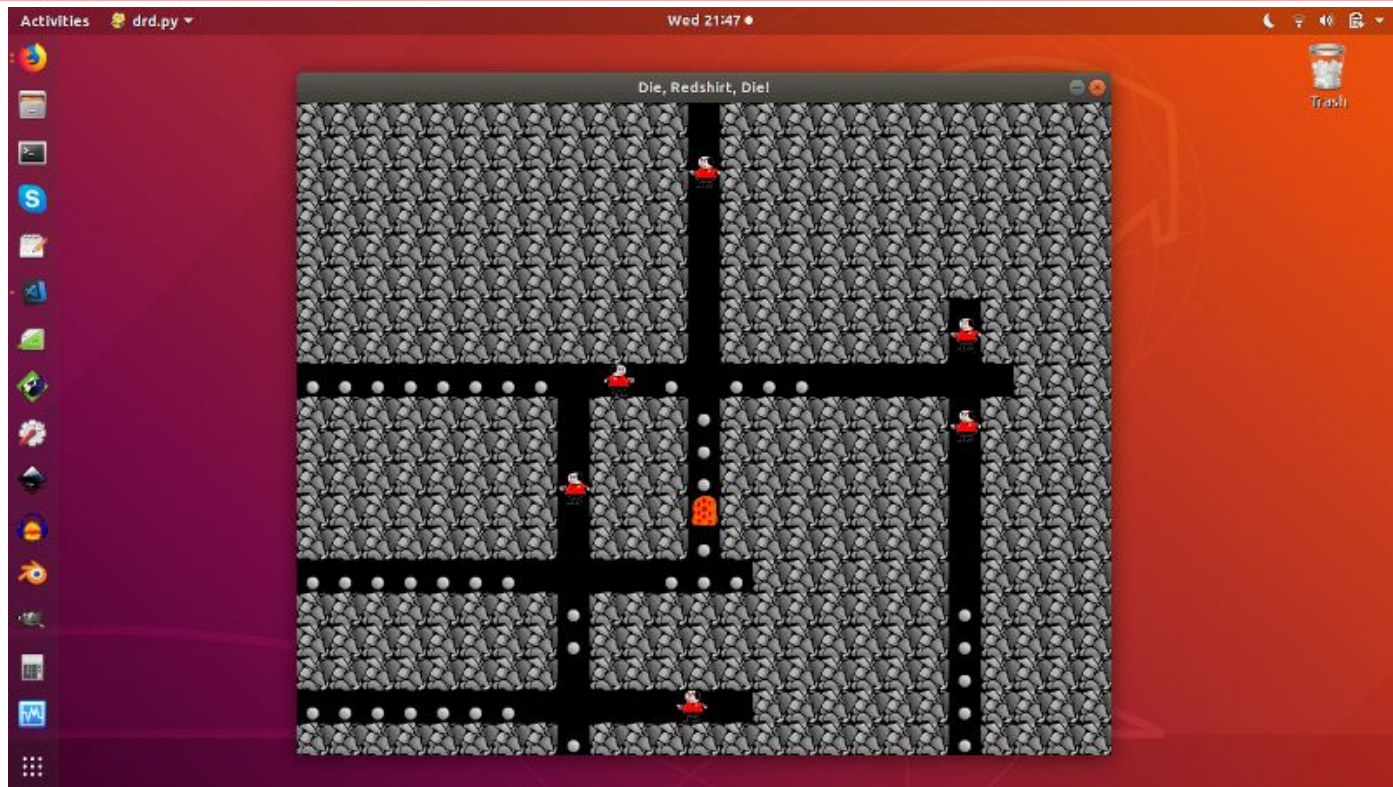
drd.py x
326
327     def run(self):
328         # Game loop
329         self.clock.tick(FPS)
330         self.playing = True
331         while self.playing:
332             self.clock.tick(FPS)
333             self.events()
334             self.update()
335             self.draw()
336
337         if self.running == True:
338             pygame.time.delay(1000)
339
340     def update(self): ...
507
508     def events(self): ...
525
526     def draw(self): ...
554
555     def mandiesnd(self): ...
576
```

Ln 554, Col 1 Spaces: 4 UTF-8 LF Python

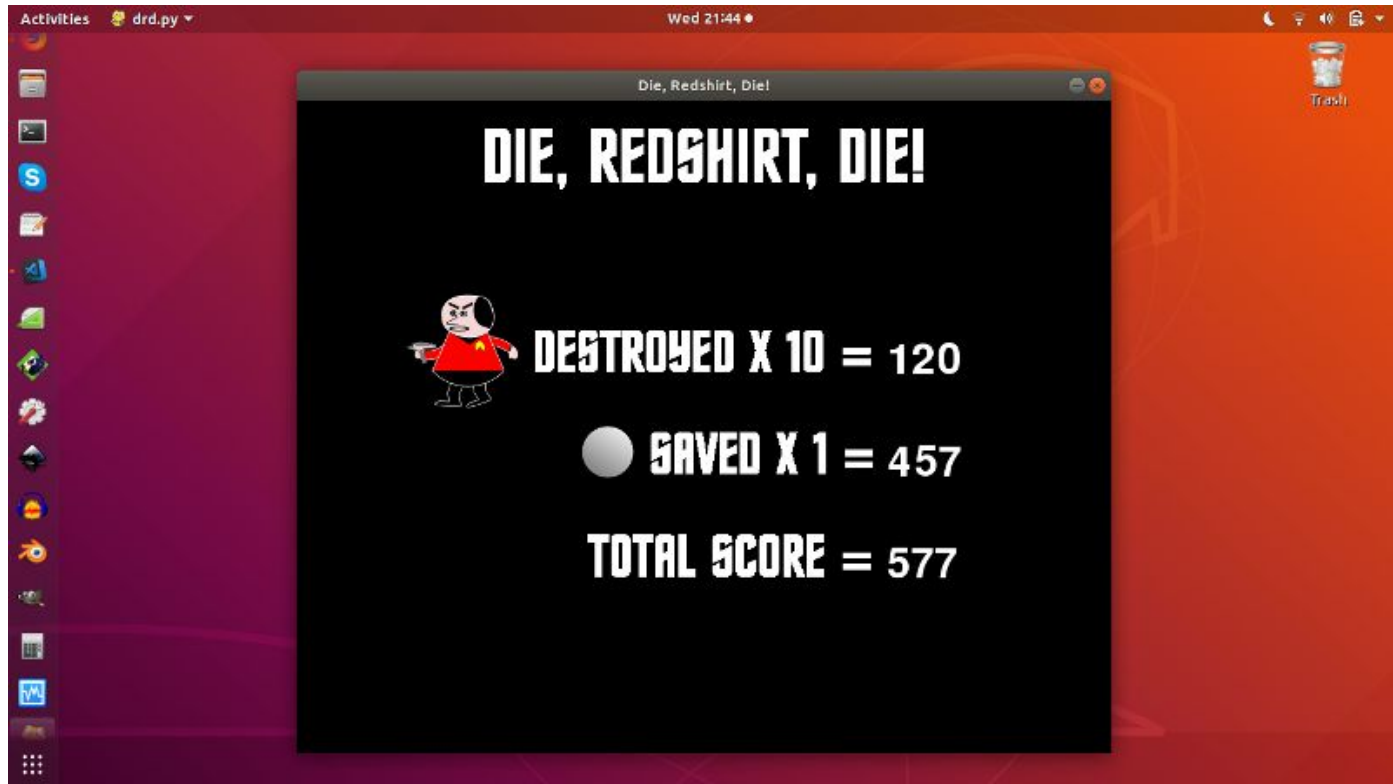
Start game screen



Run game screen



Game over screen



Questions and/or Comments

My First Pygame

Terry Howald

terry.howald@gmail.com

<https://github.com/terryhowald/drd>
