Part 5 Space Mission Directions

- 1. Navigate out to the Google Classroom for this class.
- 2. Locate the Space Mission Part 5 assignment.
- 3. We are now ready to start adding code to our file. Using your Windows button menu, find and launch your IDLE program.



IDLE is the integrated development environment associated with Python. It is made up of a code editor where you type your code along with other helpful tools that allow you to write, save, and test run programs.

IDLE is designed to recognize Python code, compile Python code, and provide basic debugging tips to programmers if there are problems with their code.

4. Your IDLE window should look something like this once it has launched.:



On Startup, IDLE will display the Python Shell, which can be used to give commands to the computer's operating system. Since we are viewing the shell through IDLE and not the actual command prompt window, the commands that we type into the Shell will not communicate directly with our operating system. However, you can type similar commands in the Python Shell directly from the Python program (not through IDLE) and, if you have permission to access the operating system's commands, you can communicate with the computer's operating system that way.

In IDLE, the shell is mainly used as a launching screen for other activities that we will do, like writing code for our game or debugging a file.

5. Go to File > Open and then browse in the Starting Files folder I gave you to find the escape python file that we have been working on.

🗹 📴 escape	11/22/2021 8:34 AM	Python File	0 KB

- 6. Your escape.py file will open up.
- 7. Click at the end of Line 54.

```
"down": [images.spacesuit_front, images.spacesuit_front_1,
45
46
                images.spacesuit_front_2, images.spacesuit_front_3,
47
                images.spacesuit front 4
48
                1
       }
49
50
51 player direction = "down"
52 player frame = 0
53 player_image = PLAYER[player_direction][player_frame]
54 player_offset_x, player_offset_y = 0, 0
55
56
57 ################
58 ##
        MAP
               ##
59 ################
```

9. Type the code you see on Lines 56 – 73 of the screenshot below. Ensure your indentation and capitalization match what is shown in the screenshot.

```
51 player direction = "down"
52 player frame = 0
53 player image = PLAYER[player direction][player frame]
54 player offset x, player offset y = 0, 0
55
56 PLAYER SHADOW = {
       "left": [images.spacesuit left shadow, images.spacesuit left 1 shadow,
57
                images.spacesuit left 2 shadow, images.spacesuit left 3 shadow,
58
                images.spacesuit left_3_shadow
59
60
                1,
61
       "right": [images.spacesuit right shadow, images.spacesuit right 1 shadow,
62
                 images.spacesuit right 2 shadow,
63
                 images.spacesuit right 3 shadow, images.spacesuit right 3 shadow
64
                 1,
65
       "up": [images.spacesuit back shadow, images.spacesuit back 1 shadow,
              images.spacesuit back 2 shadow, images.spacesuit back 3 shadow,
66
67
              images.spacesuit back 3 shadow
68
              1.
69
       "down": [images.spacesuit_front_shadow, images.spacesuit_front_l_shadow,
70
                images.spacesuit_front_2_shadow, images.spacesuit_front_3_shadow,
71
                images.spacesuit front 3 shadow
72
                1
73
       }
74
75
76 ###############
```

Line 56 will create another dictionary in your game called PLAYER_SHADOW. This is similar to the player dictionary. It contains animation frames for the astronaut's shadow on the floor. As the astronaut moves, the shadow also changes shape.

Lines 57 – 73 contains dictionary keys ("left", "right", "up", and "down") and the images for each shadow position that the animation will iterate through.

11. Type the code you see on Line 75 of the screenshot below.

```
56 PLAYER SHADOW = {
57
       "left": [images.spacesuit left shadow, images.spacesuit left 1 shadow,
                images.spacesuit left 2 shadow, images.spacesuit left 3 shadow,
58
                images.spacesuit left 3 shadow
59
60
               1,
      "right": [images.spacesuit right shadow, images.spacesuit right 1 shadow,
61
62
                 images.spacesuit_right_2_shadow,
63
                 images.spacesuit right 3 shadow, images.spacesuit right 3 shadow
64
                 1,
       "up": [images.spacesuit back shadow, images.spacesuit back 1 shadow,
65
              images.spacesuit back 2 shadow, images.spacesuit back 3 shadow,
66
              images.spacesuit back 3 shadow
67
68
             1,
69
      "down": [images.spacesuit front shadow, images.spacesuit front 1 shadow,
70
               images.spacesuit front 2 shadow, images.spacesuit front 3 shadow,
71
               images.spacesuit front 3 shadow
72
               1
73
       }
74
75 player image shadow = PLAYER SHADOW["down"][0]
76
77
78 #################
79 ## MAP ##
```

Line 75 creates a new variable called player_image_shadow. Its default value is equal to the first item for the "down" entry in the PLAYER_SHADOW dictionary (the item with the index value of 0). This variable will store the astronaut's current shadow, like the player_image variable that we created earlier store's the astronaut's current image.

13. Type the code you see on Lines 77 – 80 of the screenshot below. Ensure your punctuation, indentation, and capitalization match what is shown in the screenshot.

```
56 PLAYER SHADOW = {
57
       "left": [images.spacesuit left_shadow, images.spacesuit_left_l_shadow,
58
                images.spacesuit left 2 shadow, images.spacesuit left 3 shadow,
59
                images.spacesuit left 3 shadow
60
                1,
       "right": [images.spacesuit right shadow, images.spacesuit right 1 shadow,
61
62
                 images.spacesuit right 2 shadow,
63
                 images.spacesuit right 3 shadow, images.spacesuit right 3 shadow
64
                 1,
65
       "up": [images.spacesuit_back_shadow, images.spacesuit_back_1_shadow,
66
              images.spacesuit back 2 shadow, images.spacesuit back 3 shadow,
67
              images.spacesuit back 3 shadow
68
             1,
       "down": [images.spacesuit_front_shadow, images.spacesuit_front_1_shadow,
69
70
                images.spacesuit front 2 shadow, images.spacesuit front 3 shadow,
                images.spacesuit front 3 shadow
71
72
                1
73
       ł
74
75 player image shadow = PLAYER SHADOW["down"][0]
76
77 PILLARS = [
78
       images.pillar, images.pillar 95, images.pillar 80,
79
       images.pillar 60, images.pillar 50
80
       1
81
82
84 ##
        MAP
               ##
85 #################
```

Later in the chapter, we will add animation that fades out the front wall when you walk behind it so you can still see the astronaut. Line 77 creates the PILLARS list that lists different image frames the animation can move through (Lines 78 – 79) when we do this.

15. Type the code you see on Line 82 of the screenshot below.

```
51 player_direction = "down"
52 player frame = 0
53 player image = PLAYER[player direction][player frame]
54 player offset x, player offset y = 0, 0
55
56 PLAYER SHADOW = {
57
     "left": [images.spacesuit left shadow, images.spacesuit left 1 shadow,
                images.spacesuit left 2 shadow, images.spacesuit left 3 shadow,
58
59
                images.spacesuit left 3 shadow
60
                1.
61
      "right": [images.spacesuit right shadow, images.spacesuit right 1 shadow,
                 images.spacesuit_right_2_shadow,
62
                 images.spacesuit_right_3_shadow, images.spacesuit_right_3_shadow
63
64
                 1,
     "up": [images.spacesuit back shadow, images.spacesuit back 1 shadow,
65
              images.spacesuit back 2 shadow, images.spacesuit back 3 shadow,
66
67
              images.spacesuit back 3 shadow
68
             ],
69
     "down": [images.spacesuit front shadow, images.spacesuit front 1 shadow,
70
               images.spacesuit front 2 shadow, images.spacesuit front 3 shadow,
71
                images.spacesuit front 3 shadow
72
                1
73
       ŀ
74
75 player image shadow = PLAYER SHADOW["down"][0]
76
77 PILLARS = [
78
       images.pillar, images.pillar 95, images.pillar 80,
       images.pillar_60, images.pillar_50
79
80
       1
81
82 wall_transparency frame = 0
83
84
85 #################
86 ## MAP ##
87 ################
```

Line 82 creates a new variable called wall_transparency_frame. This variable will remember the PILLARS image frame that is currently being displayed. It is set to the index value of 0, which will be the first item in the PILLARS list (in this case, the images.pillar).

17. Type the code you see on Lines 84 – 89 of the screenshot below.

```
77 PILLARS = [
78
       images.pillar, images.pillar 95, images.pillar 80,
79
       images.pillar 60, images.pillar 50
80
       1
81
82 wall transparency frame = 0
83
84 BLACK = (0, 0, 0)
85 | BLUE = (0, 155, 255)
86 YELLOW = (255, 255, 0)
87 WHITE = (255, 255, 255)
88 GREEN = (0, 255, 0)
89 \text{ RED} = (128, 0, 0)
90
91
92 ################
93 ## MAP
              ##
94 #################
```

Lines 84 – 89 will set up different variables named BLACK, BLUE, YELLOW, WHITE, GREEN, and RED. The numbers after the variables indicate the RGB color values for each variable. This sets the specific shade of each color you would like to use in your game.

Colors in Pygame are stored as tuples. A tuple is like a list whose content you can't change, and it uses parentheses instead of square brackets. You've seen tuples used for coordinates when drawing on the screen. Colors are stored as three numbers that specify the amount of red, green, and blue in the color, in that order. The scale for each color ranges from 0 to 255.

- 18. Ensure that your "MAP" comment runs on Lines 92 94 of your code. You may have to add or delete blank lines to make this happen.
- 19. Scroll and click at the end of Line 455.

443	if current_room in scenery:
444	<pre>for this_scenery in scenery[current_room]:</pre>
445	<pre>scenery_number = this_scenery[0]</pre>
446	<pre>scenery_y = this_scenery[1]</pre>
447	<pre>scenery_x = this_scenery[2]</pre>
448	room_map[scenery_y][scenery_x] = scenery_number
449	
450	<pre>image_here = objects[scenery_number][0]</pre>
451	<pre>image_width = image_here.get_width()</pre>
452	image_width_in_tiles = int(image_width / TILE_SIZE)
453	
454	<pre>for tile_number in range(l, image_width_in_tiles):</pre>
455	room_map[scenery_y][scenery_x + tile_number] = 255
456	
457	*******
458	## GAME LOOP ##
459	*******

- 20. Press ENTER twice.
- 21. Type the code you see on Lines 457 462 of the screenshot below. Ensure your indentation matches what is shown in the screenshot.

```
443
        if current room in scenery:
444
            for this scenery in scenery[current room]:
445
                scenery number = this scenery[0]
                scenery y = this scenery[1]
446
447
                scenery x = this scenery[2]
448
                room map[scenery y][scenery x] = scenery number
449
450
                image here = objects[scenery number][0]
451
                image width = image here.get width()
452
               image width in tiles = int(image width / TILE SIZE)
453
454
                for tile_number in range(1, image_width_in_tiles):
455
                    room_map[scenery_y][scenery_x + tile_number] = 255
456
457
       center y = int(HEIGHT / 2) # Center of game window
       center x = int(WIDTH / 2)
458
459
      room pixel width = room width * TILE SIZE # Size of room in pixels
      room pixel height = room height * TILE SIZE
460
461
       top left x = center x - 0.5 * room pixel width
462
        top left y = (center y - 0.5 * room pixel height) + 110
463
464
465 #################
466 ## GAME LOOP ##
467 ################
```

We now need to position the room on the screen.

The code you just typed starts by working out where the middle of the window in a room is. The HEIGHT and WIDTH variables store the window's size in pixels. Dividing them by 2 gives us the coordinates of the center of the window. We store these coordinates in the center_y and center_x variables (Lines 457 - 458).

This program also works out how wide or tall the image of the room is in pixels (Lines 459 – 460). This will be the width or height of the room in tiles multiplied by the size of a tile. The result is stored in the room_pixel_width and room_pixel_height variables.

To put the room image in the middle of the room, we want half the room to be to the left of the center line and half to the right. So, we subtract half the room width in pixels from the center line and start drawing the room there (Line 461). A similar calculation is made for the top_left_y variable, except we add 110 to the result because our final screen layout will use an area at the top of the screen as an information panel (Line 462).

22. Click at the end of Line 467.

24. Type the code you see on Lines 469 – 470 of the screenshot below. Ensure your indentation matches what is shown in the screenshot.

```
465 ###############
466 ## GAME LOOP ##
467 #################
468
469 def start room():
470
      show text("You are here: " + room name, 0)
471
472 def game loop():
       global player_x, player_y, current_room
473
474
      global from_player_x, from_player_y
475
      global player image, player image shadow
476
      global selected_item, item_carrying, energy
    global player_offset_x, player_offset_y
global player_frame, player_direction
477
478
```

Line 469 creates a new function called start_room in the GAME LOOP section of your code.

Line 470 will run the show_text method to display the text, "You are here: " along with the room_name, taken from the GAME_MAP list. The 0 at the end of the line of code indicates the line number. We haven't created the show_text method in our code yet, so this will make more sense later.

25. Scroll and select Lines 583 – 633 (the entire "EXPLORER" section of your code).



26. Press BACKSPACE to delete this entire section. We won't need it anymore.

```
572
        if player direction == "right" and player frame > 0:
573
            player offset x = -1 + (0.25 * player frame)
574
        if player direction == "left" and player frame > 0:
            player offset x = 1 - (0.25 * player frame)
575
576
       if player direction == "up" and player frame > 0:
577
            player offset y = 1 - (0.25 * player frame)
       if player direction == "down" and player frame > 0:
578
579
            player_offset_y = -1 + (0.25 * player_frame)
580
581
582
583
584 ##################
585 ## START ##
586 #################
587
588 generate map()
589 clock.schedule interval(game loop, 0.03)
```

27. Type the code you see on Lines 583 – 592 of the screenshot below. Ensure your indentation matches what is shown in the screenshot below.

```
584 ## DISPLAY ##
585 #################
586
587 def draw_image(image, y, x):
588
      screen.blit(
589
          image,
590
          (top left x + (x * TILE SIZE),
591
           top left y + (y * TILE SIZE) - image.get height())
          )
592
594 ## START ##
595 ################
596
597 generate map()
598 clock.schedule interval(game loop, 0.03)
```

Lines 583 – 585 create a new section of code, called DISPLAU.

Line 587 creates a new method called draw_image. This method will require the image we want to draw and the and the y and x tile positions of the object in the room whenever it is called. The function will work out where on the screen to draw the image (the pixel position) based on the tile position in the room.

Line 588 - 592 will utilize the screen.blit command to blit the image at the location specified.

29. Type the code you see on Lines 594 – 599 of the screenshot below. Ensure your indentation matches what is shown in the screenshot.

```
584 ## DISPLAY ##
585 #################
586
587 def draw image(image, y, x):
588
     screen.blit(
589
          image,
          (top left x + (x * TILE SIZE),
590
591
          top left y + (y * TILE SIZE) - image.get height())
592
          )
593
594 def draw shadow(image, y, x):
     screen.blit(
595
596
          image,
597
          (top_left_x + (x * TILE_SIZE),
598
          top left y + (y * TILE SIZE))
599
          )
601 ##
      START ##
```

Line 594 creates a new function called draw_shadow. This function will also require the image you want to use and the x and y tile positions when it is called.

Lines 595 – 599 will utilize the screen.blit command to blit the image at the location specified.

The draw_shadow() function is very similar to the function for drawing an image, except that the image's height is not subtracted when calculating its onscreen position. This is what places the shadow below the main image.

31. Type the code you see on Lines 601 – 607 of the screenshot below. Ensure your indentation matches what is shown in the screenshot.

```
584 ## DISPLAY ##
586
587 def draw_image(image, y, x):
588 screen.blit(
589
        image,
         (top left x + (x * TILE SIZE),
590
          top left_y + (y * TILE_SIZE) - image.get_height())
591
592
          )
593
594 def draw shadow(image, y, x):
595
     screen.blit(
596
          image,
          (top_left_x + (x * TILE SIZE),
597
          top_left_y + (y * TILE SIZE))
598
599
          )
600
601 def draw player():
602
    player image = PLAYER[player direction][player frame]
603
      draw image(player image, player y + player offset y,
604
                player x + player offset x)
     player_image_shadow = PLAYER_SHADOW[player_direction][player_frame]
605
606
      draw shadow(player image shadow, player y + player offset y,
     player_x + player_offset_x)
607
```

Line 601 creates another new function called draw_player. This function will draw the astronaut.

First, it puts the correct animation frame into the player_image variable (Line 602).

Lines 603 - 604 then uses the draw_image function that we created on Line 587 to draw the astronaut's image. The draw_image function requires the following arguments:

- The variable player_image, which contains the image to draw.
- The result after adding the global variables for player_y and player_offset_y. This is the y position in tiles, which might include a decimal part.
- The result after adding player_x and player_offset_x for the x position in tiles.

Lines 605 – 607 use a similar code to draw the player's shadow: the correct animation frame from the PLAYER_SHADOW dictionary is put into the player_image_shadow variable. Then, the draw_shadow function is used to draw it. The draw_shadow function uses the same tile positions as the draw_image function.

33. Type the code you see on Lines 609 – 625 of the screenshot below. Ensure your indentation, punctuation, and line spacing matches what is shown in the screenshot.

```
601 def draw player():
602
       player image = PLAYER[player direction][player frame]
603
       draw image(player image, player y + player offset y,
604
                  player x + player offset x)
     player image shadow = PLAYER SHADOW[player direction][player frame]
605
       draw shadow(player image shadow, player y + player offset y,
606
607
                   player x + player offset x)
608
609 def draw():
610
     if game over:
611
          return
612
613
      # Clear the game arena area.
614
      box = Rect((0, 150), (800, 600))
615
      screen.draw.filled rect(box, RED)
616
      box = Rect ((0, 0), (800, top_left_y + (room_height - 1)*30))
617
      screen.surface.set clip(box)
618
      floor_type = get_floor_type()
619
620
     for y in range(room_height): # Lay down floor tiles, then items on floor.
621
          for x in range(room width):
               draw image(objects[floor type][0], y, x)
622
623
               # Next line enables shadows to fall on top of objects on floor
62.4
               if room map[y][x] in items player may stand on:
625
                   draw image(objects[room map[y][x]][0], y, x)
626 #################
627 ## START ##
628 #################
```

Line 609 creates a new function called draw.

Line 610 will check to see if the game_over variable is equal to True. If so, then Line 611 contains a return statement that will exit out of this particular block of code and skip down to the next function. This block of code doesn't need to run if the game is over.

Line 613 contains a comment.

Line 614 begins the process of clearing the game arena, where the space station will be drawn. It does this by drawing a big red rectangle, wiping out the previous screen display. The areas as the top and the bottom that give the player information are separate, so they are not changed.

There are two steps for putting a rectangle on the screen. First, you create the shape using a Pygame object called Rect. On Line 614, we create a rect object called box. The object is placed at the x and y coordinate location of 0 and 150 and is 800 pixels wide by 600 pixels tall.

Line 615 will draw the filled rect object named box on the screen, and will fill the rectangle with the color numbers saved in the RED variable. These numbers are the RGB color values we specified earlier.

Line 616 creates another rect object called box. This object is created after the previous rect object is drawn so that we overwrite the coordinates of the first rect object with the coordinates of the second. The second rect box object will have an x and y location of 0, 0. It will be 800 pixels wide and whatever height is appropriate for the current room the player is in.

You can also use the Rect shape to create a clipping area. This is like an invisible window through which you view the screen. If the program draws something outside the window, it can't be seen. Line 617 sets up a clipping area that's the height of the room to stop the player's shadow from spilling out of the bottom of the game when they're in the front doorway.

Line 618 creates a variable called floor_type. This line will run the get_floor_type() method and store the result in the floor_type variable on Line 618.

The room is drawn in two stages. First, the program draws the floor tiles and anything that the player can walk on. Drawing them first enables scenery, the player, and shadows to be drawn on top of them. This solves the problem of black holes appearing under scenery, because there will be floor tiles in those spaces before the scenery is drawn.

Lines 620 – 621 create "for" loops that will loop for each y coordinates and x coordinate in the room.

Line 622 executes the draw_image function that will access the objects list based on the floor_type of the room that you calculated on Line 618. This will draw the appropriate floor type image on the screen at the appropriate y and x location (whatever location is currently being iterated through in the loop).

Line 623 contains a comment.

Line 624 contains an if function that will check to see if that particular y and x coordinate in the room_map contains items that the player is allowed to stand on.

If this is true, then Line 625 executes the draw_image function again to access the objects list for that particular y and x coordinate and returns the first item in the list, which is the image of that particular object. It will generate the item at the appropriate y and x position.

```
punctuation, and line spacing match what is shown in the screenshot.
620
        for y in range(room height): # Lay down floor tiles, then items on floor.
621
            for x in range(room_width):
622
                draw_image(objects[floor_type][0], y, x)
623
                # Next line enables shadows to fall on top of objects on floor
624
                if room map[y][x] in items player may stand on:
625
                    draw image(objects[room map[y][x]][0], y, x)
626
627
        # Pressure pad in room 26 is added here, so props can go on top of it.
628
       if current_room == 26:
629
           draw image(objects[39][0], 8, 2)
630
           image on pad = room map[8][2]
631
           if image on pad > 0:
632
                draw_image(objects[image_on_pad][0], 8, 2)
633
634
       for y in range(room height):
635
            for x in range(room width):
636
                item here = room map[y][x]
637
                # Player cannot walk on 255: it marks spaces used by wide objects.
638
                if item_here not in items_player_may_stand_on + [255]:
639
                    image = objects[item here][0]
640
641
                    if (current room in outdoor rooms
642
                        and y == room height - 1
643
                        and room map[y][x] == 1) or \setminus
644
                        (current room not in outdoor rooms
645
                        and y == room height - 1
646
                        and room_map[y][x] == 1
647
                        and \mathbf{x} > 0
648
                        and x < room_width - 1):
649
                        # Add transparent wall image in the front row.
650
                        image = PILLARS[wall transparency frame]
651
652 #################
653 ## START ##
654 #################
```

35. Type the code you see on Lines 627 – 650 of the screenshot below. Ensure your indentation,

```
Line 627 contains a comment.
```

Line 628 contains an if function that checks to see if the value of the current_room variable is equal to room 26.

If it is, Lines 629 - 632 will execute.

Line 629 will execute the draw_image function to draw the floor pad image (item #39 in the objects list, the first item in the list is the image associated with the floor_pad). 8 is the y tile location for the image and 2 is the x tile location.

Line 630 creates a variable called image_on_map and sets it equal to be the y and x tile coordinate of 8 and 2 on the room_map.

Line 631 checks to see if the image_on_pad variable is greater than 0. If it is, it will draw the floor pad image at the proper tile location (Line 632).

Line 634 begins the second stage of drawing the room. The program will add scenery in the room, including shadows, using new loops that begin on Line 634. Because these loops come after the floor for the whole room has been drawn, the shadows will be drawn on top of the floor tiles and items on the floor. The shadows are transparent, so you can still see the object underneath the shadow.

Beginning on Line 641, the program begins drawing the transparent front wall.

When the program is drawing the front row of the room (when the y loop equals room_height – 1), it checks whether it needs to draw a semitransparent wall instead of the solid wall object taken from the room map. The semitransparent wall is used if the player is standing behind it.

On the planet surface, the program makes the whole wall transparent. Inside the space station, a transparent wall panel is used only if its not in one of the bottom former positions. The corners always use a solid wall panel. The reason is that it loops odd if you see the solid edge wall start in the second from from the bottom.

Later one, we will add the code to animate the transparency on the wall by changing the number in wall_transparency_frame. You won't see the semi-transparent wall yet in the game.

37. Type the code you see on Lines 652 – 669 of the screenshot below. Ensure your indentation, punctuation, and line spacing match what is shown in the screenshot.

```
641
                     if (current room in outdoor rooms
642
                         and y == room height - 1
643
                         and room map[y][x] == 1) or \setminus
644
                         (current room not in outdoor rooms
645
                         and y == room height - 1
646
                         and room map[y][x] == 1
647
                         and \mathbf{x} > 0
648
                         and x < room width - 1):
649
                         # Add transparent wall image in the front row.
650
                         image = PILLARS[wall transparency frame]
651
652
                    draw image(image, y, x)
653
654
                     if objects[item here][1] is not None: # If object has a shadow
655
                         shadow image = objects[item here][1]
656
                         # if shadow might need horizontal tiling
657
                         if shadow image in [images.half shadow,
658
                                              images.full shadow]:
659
                             shadow width = int(image.get width() / TILE SIZE)
660
                             # Use shadow across width of object.
661
                             for z in range(0, shadow width):
662
                                 draw shadow(shadow image, y, x+z)
663
                         else:
664
                             draw shadow(shadow image, y, x)
665
            if (player_y == y):
666
667
                    draw_player()
668
669
        screen.surface.set clip(None)
670 ################
671 ## START ##
672 #################
```

Line 652 draws the player on top of the floor.

Line 654 will check to see if the current object being drawn has a shadow.

If it does, Line 655 creates a variable called shadow_image and sets its value to be equal to the second item in the objects list for the particular item being drawn.

Line 656 contains a comment.

Lines 657 – 658 check to see if the object has a half_shadow or full_shadow, which would fill half a tile or a whole tile, respectively. These two standard shadows are used with block items (like electrical units and walls) that don't need a distinctive shadow outline. The program checks whether the shadow_image is in a list that contains those two standard images.

If the shadow is one of the standard images, the program then works out how wide the shadow should be in tiles. That is calculated by taking the width of the object casting the shadow and dividing it by the width of a tile (30 pixels).

The program then creates a loop to draw the standard shadow images, using the variable z. It starts at 0 and runs until the width of the shadow minus 1. That's because a range leaves out the last item. The z values are added to the x position from the main loop and are used to draw the shadow tiles.

If the shadow image is not one of the standard shadow images, the program will execute the draw_shadow function using the shadow_image from the objects dictionary and the current u and x tile location.

Line 666 will check to see if the player_y coordinate is equal to the y location that is currently being iterated through in the "for" loop that starts on Line 634. If this is true, the draw_player() function will execute (Line 667) to draw the player image. By drawing the player image after all the floor tiles and scenery, the player image will be placed on top.

Line 669 turns off the clipping area that was set earlier.

- 38. Press ENTER twice.
- 39. Type the code you see on Lines 671 677 of the screenshot below. Ensure your indentation and line spacing matches what is shown in the screenshot below.

```
663
                        else:
664
                            draw shadow(shadow image, y, x)
665
666
           if (player y == y):
667
                   draw player()
668
669
      screen.surface.set clip(None)
670
671 def adjust_wall_transparency():
672 global wall transparency frame
673
674
     if (player_y == room height - 2
675
           and room map[room height - 1][player_x] == 1
676
           and wall transparency frame < 4):
           wall transparency frame += 1 # Fade wall out.
677
678 ##################
679 ## START
                 ##
680 ################
```

Line 671 creates another function called adjust_wall_transparency.

Line 672 converts the wall_transparency_frame to a global variable so that this function is able to access and modify the value of the wall_transparency_frame variable.

If the player is standing behind the wall, the flowing statements are true:

- Their u position will be equal to room_height-2.
- There is a piece of wall in the bottom row of the room that is in line with the player's x position.

If the player is behind the wall (Lines 674 – 675) and the wall transparency is not set to maximum (Line 676), the wall transparency is increased by 1, making the wall more transparent (Line 677)

- 40. Press ENTER twice.
- 41. Type the code you see on Lines 679 682 of the screenshot below. Ensure your indentation matches what is shown in the screenshot.

```
669
       screen.surface.set_clip(None)
670
671 def adjust wall transparency():
672
      global wall transparency frame
673
674
       if (player y == room height - 2
675
           and room map[room height - 1][player x] == 1
676
           and wall transparency frame < 4):
677
          wall transparency frame += 1 # Fade wall out.
678
679
     if ((player y < room height - 2</pre>
680
              or room map[room height - 1][player x] != 1)
681
              and wall transparency frame > 0):
           wall transparency frame -= 1 # Fade wall in.
682
684 ##
               ##
        START
```

If either of the following is true, it means the player isn't hidden by the wall:

- Their y position is less than room_height 2. The player can be seen, at least in part, if they're farther back in the room.
- There is not a piece of wall in the bottom row of the room in line with their x position.

In these cases, if the wall transparency is set to more than the minimum, it's reduced by one.

43. Type the code you see on Lines 684 – 691 of the screenshot below. Ensure your indentation, punctuation, and line spacing match what is shown in the screenshot.

```
671 def adjust wall transparency():
672
       global wall transparency frame
673
674
       if (player y == room height - 2
675
           and room map[room height - 1][player x] == 1
676
           and wall transparency frame < 4):
677
           wall transparency frame += 1 # Fade wall out.
678
679
      if ((player y < room height - 2</pre>
680
               or room map[room height - 1][player x] != 1)
               and wall transparency frame > 0):
681
682
           wall transparency frame -= 1 # Fade wall in.
683
684 def show text(text_to_show, line_number):
685 if game over:
686
           return
687
      text lines = [15, 50]
     box = Rect((0, text lines[line number]), (800, 35))
688
689
      screen.draw.filled rect(box, BLACK)
690
      screen.draw.text(text to show,
691
                       (20, text lines[line number]), color=GREEN)
692 #################
693 ## START ##
```

Line 684 creates another method called show_text.

Line 685 checks to see if the game_over variable is set to True. If so, Line 686 contains a return statement to exit out of this block of code and skip down to the next function. This code doesn't need to execute if the game is over.

The line number will be either 0 for the top row or 1 for the second row, which is reserved for important messages. When the function is called, the message is put into the variable text_to_show and the row number goes into the line_number position.

We use a list called text_lines to remember the vertical positions (in pixels) of the two lines of text (Line 687).

Line 688 will define a box at the x-coordinate of 0 and the y-coordinate equal to the line_number specified when the function is called. The width of the rectangle will be 800 and the height is 35 pixels.

Line 689 fills the rectangle object with black to clear the row of text before the new message is drawn.

Finally, we use the screen.draw.text() function in Pygame to put the text on the screen. This function takes the text, the text's x and y position, and the text color. The position numbers go inside parentheses.

The x position is 20 pixels from the left and the vertical position is taken from the text_lines list, using the number in line_number as the list index.

```
44. Press ENTER three times. Ensure your "START" comment runs on Lines 695 – 697.
```

```
684 def show text(text to show, line number):
685
     if game over:
686
          return
     text_lines = [15, 50]
687
     box = Rect((0, text lines[line number]), (800, 35))
688
689
      screen.draw.filled rect(box, BLACK)
690 screen.draw.text(text_to_show,
691
                       (20, text lines[line number]), color=GREEN)
692
693
694
696 ## START ##
697 ##################
698
699 generate map()
700 clock.schedule interval(game loop, 0.03)
```

45. Click at the end of Line 700.

46. Press ENTER.

47. Type the code you see on Line 701 of the screenshot below.

Line 701 will run the adjust_wall_transparency function every 0.5 seconds. This makes the wall fade in or out as necessary as the player walks around the room.

48. Go to File > Save Now to save your code.

Final Code:

	2 Escare
	P Longe
-	import time, random, math
3	
1	## VARIABLES ##

3	VIIII - 600 Augustus ales
	HIIGHT = 800
	FILITER VATIABLES
	FRIENDI NAME = "Jack"
3	FRIENDL NAME = "Hatthew"
	current_room = 31 # start room = 31
	top left x = 100
2.3	top_left_y = 150
	DENO OBJECTS = (images,floor, images,cillar, images,scil)
	LANDER_SECTOR = random/randint(1, 24)
	LANDER X = rendom.randint(2, 11)
20	
	TILE_SIZE = 30
	player v. player x = 27 5
10	game_cover = Talam
	"left": [images.spacesuit left, images.spacesuit left],
6	images.spacesait_left_1, images.spacesuit_left_3,
	images.spacesuit_left_4
17	"right": Simages.spacesuit right, images.spacesuit right 1,
18	images.spacesult_right_2, images.spacesult_right_3,
	images.spacesult_right_i
	"im": finaces.specesuit back, imaces.specesuit back 1.
12	images.spacesuit back 2, images.spacesuit back 3,
13	images.spacesuit_back_4
	"ddwn": (images.spacesuit front, images.spacesuit front 1.
ic	inages.spacesuit_front_2, images.spacesuit_front_3,
17	inages.spacesuit_front_4
50	
	player_direction = "down"
	Planter frame a C
52	player_frame = 0 player_image = FLAYER(player_direction)[player_frame]
12 13 14	<pre>player_reame = 0 player_impe = FLAYEN(player_direction)(player_frame) player_offset_m, player_offset_y = 0, 0</pre>
1211	panyer_reame = 0 planyer_manye = ELAURE(planyer_direction)[planyer_frame] planyer_offner_m, planyer_offner_y = 0, 0
	<pre>paywr praws = 0 player profe = AUXER[player_direction][player_frame] player_offser_s, player_offser_y = 0, 0 PLAYES_PRADOR = 1 */ctv* [mass.snacewaitleft shadow, inaper.spacesuitleft] shadow,</pre>
23356 67 6	<pre>player_track = 0 player_track = FLATER[player_direction][player_track] player_direck, player_direct, y = 0, 0 player_direck, player_direct, y = 0, 0 player_direck, player_direct, y = 0, 0 player_direck, player_direck, player_direck, player_direck, player, p</pre>
1233-55 (47 10 10 10 10 10 10 10 10 10 10 10 10 10	<pre>paywe_trame = 0 playme_trame = 0 playme_trame = NUTRE[playme_direction][playme_trame] playme_direction</pre>
527557 (6750 50 51 51 51 51 51 51 51 51 51 51 51 51 51	<pre>player_trace = v player_trace + FLATER[player_direction][player_frame] player_dfare_n_player_dfare(y = v, 0) "NETWO NAMEON = 1 "NETWO NAMEON = 1 "NETWO NAMEON = 1 NETWO NA</pre>
523555 67 10 0 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 0 1	<pre>payse trace = 0 payse trace = table = filled the filled trace = filled trace</pre>
	<pre>player_track = 0 player_track = 0 player_track = FLATER[player_track] player_tracks = FLATER[player_track] player_tracks = FLATER[player_track] "left" [Image: apacents[left_shadow, images.spacents_left_shadow, "slott" [Image: apacents[left_shadow, images.spacents_left_shadow, image: apacents[left_shadow, images.spacents_left_shadow, image.spacents[left_shadow, images.spacents_left_shadow, image.spacents[left_shadow, images.spacents_left_shadow, image.spacents[left_shadow, images.spacents[left_shadow, image.spacents[left_shadow, images.spacents[left_shadow]</pre>
	<pre>playst_trame = 0 playst_trame = FLATE(playst_firection)[playst_frame] playst_trame = FLATE(playst_firection)[playst_frame] playst_fire(reflayst) "lift" [langes.spacewait.lift_shadow, langes.spacewait.lift_shadow, langes.spacewait.lift_shadow, langes.spacewait.lift_shadow, langes.spacewait.lift_shadow, langes.spacewait.lift_shadow, langes.spacewait.lift_shadow, langes.spacewait.lift_shadow, langes.spacewait.lift_shadow, </pre>
	<pre>player_track = v player_track = v player_track = v player_track = v player_track = v player_track = v player_track = v vieter (langes.spaceaut_left_standow, inneges.spaceaut_left_standow,</pre>
	<pre>player_track = 0 player_track = 0 player_track = FLATER[player_direction][player_frame] player_direct_N_player_directy_r = 0, 0 *left" [langes.spacewait_left_shadow, langes.spacewait_left_shadow, langes.spacewait_left_shadow fright"; langes.spacewait_left_shadow, langes.spacewait_left_shadow, langes.spacewait_left_shadow, langes.spacewait_tight_l_shadow, langes.spacewait_right_shadow, langes.spacewait_right_shadow, langes.spacewait_right_shadow, langes.spacewait_right_shadow, langes.spacewait_right_shadow, langes.spacewait_right_shadow 1, *opr:[langes.spacewait_heat_shadow, langes.spacewait_heat_shadow, langes.spacewait_heat_shadow, langes.spacewait_heat_shadow, langes.spacewait_heat_shadowait_heat_shadow, langes.spacewait_heat_shadow, lang</pre>
	<pre>player_track = v player_track = v p</pre>
	<pre>player_trace = 0 player_trace = 0 player_trace = FLATER[player_firestion][player_frame] player_trace, player_effecty = 6, 0 interv(</pre>
	<pre>piper_trame = 0 piper_trame = 0 piper_trame = transformer_trame(piper_trame) piper_trame = transformer_trame(piper_trame) piper_trame= piper_transformer_trame= piper_trame=</pre>
	<pre>player_trace = 0 player_trace = 0 player_trace = FLATER[player_firestion][player_frame] player_traces = f interv() [images.playersit][sft_2 thaddw, images.spacesuit_leftihaddw, images.spacesuit_left_2 thaddw, images.spacesuit_leftihaddw, images.spacesuit_left_2 thaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw inages.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw inages.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw, images.spacesuit_lightihaddw</pre>
	<pre>pipert inse = 0 pipert inse = 0 pipert inse = flattE [pipert direction][piper frame] pipert inse = flattE [pipert direction] [piper frame] pipert insees = pipert direction [piper direction]</pre>
	<pre>player_trace = 0 player_trace = 0 player_trace = FLATER[player_firestion][player_frame] player_trace = FLATER[player_fires] player_trace = flatter= fla</pre>
13355 670 0 0 1 2 3 4 5 6 7 H 9 1 2 3 4 5 6 7 1 9 1 2 3 4 5 6 7 H 9 1 2 3 4 5 6 7 H 9 1 2 3 4 5 6 7 1 9 1	<pre>paper trace = 0 player_trace = 0 target_traceElayer_direction[]slayer_frame] slayer_traceElayer_cffst[_y = 0, 0</pre>
	<pre>player_trace = 0 player_trace = 0 player_trace = FLATER[player_fired:info; player_ffeet_s_life; fired; fired; p = 6, 0 player_ffeet_s_life; fired; fired; fired; fired; fired; fired; player_ffeet_s_life; fired; fired; fired; fired; fired; fired; fired; player_ffeet_s_life; fired; fired; fired; fired; fired; fired; fired; player_ffeet_s_life; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fired; fire</pre>
13355 670 0 0 1 2 3 4 5 67 8 9 1 1 2 2 4 4 5 7 9 9 1	<pre>pipyer_trame = v pipyer_trame = v pipyer_trame = v pipyer_trame = v pipyer_trame = v pipyer_trame = v view = v view</pre>
	<pre>player_track = 0 player_track = 0 p</pre>
	<pre>pipyer_trame = 0 pipyer_trame = 0 pipyer_trame = transmission [pipyer_frame] pipyer_trame = transmission [pipyer_frame] pipyer_trame= pipyer_transmission [pipyer_frame]</pre>
13355 670 A0 L23 4 2 67 H 9 1 L23 4 4 5 7 7 7 7 7 7 7 1 1 2 3 4	<pre>player_trace = 0 player_trace = 0 p</pre>
动动运转 转行组织的 红脸的 转达线行 群岛的 红豆含有黄属含有黄的 计超过转位 能	<pre>player_inse_= 0 player_inse_= (insetion player_inset) player_inset = FLUTER[player_insetion player_inset] player_inset = FLUTER[player_insetion player_inset] player_inset = FLUTER[player_insetion player_insetion player_insetion </pre>
动动运转 稀纤维 地口 建固己酸盐 稀纤醋 煤油 计字语 医神经外殖 神经 计超级 网络维尔	<pre>pipipit.trass = 0 pipipit.trass = 0 pipit.trass = 0 pipit</pre>
结结结结 经计通过通过通过转转转行 链球机 社经法 体性保护 植物的 计超过转线 经行用户	<pre>player_trace = 0 player_trace = 0 player_trace = FUITE[[player_fires!] player_trace = FUITE[[player_fires]] player_trace = for the second of the se</pre>
以过远端 预计组组组组组组织统合行组织加入力分落性预计分值推计自动转转统统计算组组	<pre>piping: Intes = 0 piping: Intes = 0 Piping: Intege = FUITE[[piping: direction][piping: frame] Piping: Cffeet_, piping: direction[[piping: ngacesuit_leftihadow,</pre>
以部分所 预订通知时社经过转转的行用现在社会法有限分分分词计计计计计计计计计计计计	<pre>player_trace = 0 player_trace = (player_trace = (</pre>
以前分析 施行通道的过程过程的 经行销取消过程分布性所行的律师计和目标的施行 间的的复数	<pre>pipert.trass = 0 pipert.trass = 0 pipert.trass = 0 pipert.trass = 0 pipert.trass = 0 pipert.trass = 0 "infer" [Langes.spacesult_left"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow, inges.spacesult_lift"_shadow inges.spacesult_lift"_shadow inges.spacesult_lift"_shadow inges.spacesult_lift"_shadow inges.spacesult_lift"_shadow(inges.spacesult_frontshadow, inges.spacesult_lift"_shadow(inges.spacesult_frontshadow, inges.spacesult_lift"_shadow(inges.spacesult_frontshadow, inges.spacesult_lift"_shadow(inges.spacesult_frontshadow, inges.spacesult_lift"_shadow(inges.spacesult_frontshadow, inges.spacesult_lift"_shadow(inges.spacesult_frontshadow, inges.spacesult_lift_shadow(inges.spacesult_frontshadow, inges.spacesult_front_shadow(inges.spacesult_frontshadow, inges.splitts, inges.splitts_to i ituto_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_state_stat</pre>
经济分析 被罚的复数让起门转给给你错算你让你没有不知道?"他们们起话站给她们们的那里是没道道	<pre>player_trace = 0 player_trace = (player_trace = (</pre>
经济分析 被打道放回 计结晶转位 经存储增加 计空流存在 预计 准律师 计运行神经运行 用的复数复数转移	<pre>piputcrises = 0 piputcrises = 0 piputcrises = 0 piputcrises = 0 piputcrises = 0 piputcrises = 0 "life" [Langet spacerait eft; shadow, images spacerait_liftshadow,</pre>
经济运转 施行通道的过程过程的 施行研究的 过程分析性 经行用部门 计语言转给施行 用的 机计检查转换 计	<pre>player_trace = 0 player_trace = (</pre>
经济运转 施行通道的过程回转的施行研究加入学会在推荐介绍很好自能的特殊施行时间和利益到转转转分 经	<pre>pipert_trans = 0 pipert_trans = 0 p</pre>
经财运转 预订通过回过起回转运转价借取加入设计预推放行力推动的计划计算时运行的间间和计划计算机把预计算的	<pre>player_trace = 0 player_trace = (</pre>
经财务结 使打迫使的 计超出转位 的复数加加加加加加加加加加加加加加加加加加加加加加加加加加加加加加加加加加加加	<pre>pipupt_trans_c = 0 pipupt_trans_trans_c = 0 pipupt_trans_c = 0 pipupt_trans_trans_c = 0 pipupt_trans_c = 0 piputt_trans_c = 0 piputt_transpiputt_trans_c = 0 piputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_transpiputt_trans</pre>
经财务所 预订通过回过通过通过运行研究的社会分析性不行通常的主体计算的研究时间的的社会进程研究的的 法法	<pre>player_trans = 0 player_trans = HUREN[Player_direction[]player_frame] player_trans. = HUREN[Player_direction[]player_frame] player_direct_N_player_direct_N_h, 0 = HUREN_BLANCS = I ***********************************</pre>

for planetsectors in range(1, 26() proms 1 to 25 are generated here GAME_MAP.append(["The dusty planet surface", 13, 13, True, True])[

108 GAME_N 107 100	DF 4=[}{"Theore mease", bacget, width, Top emit?, Right emit?] ["The similod", 13, 5, Ture, Falde], € room 26
105	("The engineering law), 10, 11, False, False), 4 room 37 ("Poolat Heaten Oominit", 9, 10, False, Fause, 4 room 30
111	("The treating pailery", 9, 14, fairs, fairs, fairs), 4 room 20 ("The treats analysis, 5, rolar, fairs, fairs), 4 room 20
113	("The action's estry key", T. 11, Title, Tanle), # gound 31 ("left = allow reads", 57, "Title", False), # gound 32 ("left = allow reads", 57, "Title", False), # gound 32
115	(Phipte schlur 1000*, 7, 18, 2000, Torel, # 1000 83 (The schlares Lab*, 13, 13, 18, Labs, Torel), # 1000 84
117	["The greathcase", i3, i3, True, Feller], 6 room 35 [FLATER HARF + 's steeping quarters', 9, 11, feller, feller], 6 room 36
119	["West Sourcialor", 15, 5, Tane, Taned, 1 yours P
121	("The compute computity yous", 11, 11, True, Falled, # room 39 (Main Hassing Contail", 14, 14, Taine, Falled, # room 40
123	["The migh hey", 12, 7, 7mm, faired, 4 mag 4]
125	(Willings control somt, 5, 5, Baller, True), 5 2000 45
127	("Security permit to Minister Converts", 7, 7, Then Nierd, 4 permit 45 (27070) 1997 - 4 To Alexandro Converts", 7, 11, 1000, Tona 45
125	[TSIDDA GAME + "'s sleeping quarters", 5, 11, True, True, 1, 2 room 47
	(The photometry is a first photometry is not a first photometry is
133	['les inco wriekop', w. 14, frum, falme] # foom 50
136 #minple	and in Check Co Map Boort to Check data entry
137	Ann(ease_par)*1 - Bar_sits, 7000 fist sub boxe_par nyo/y Hatch?
1.23 ++++++	10/010
141 ###################################	RUIS ## ##################################
142 141 objects	
144 Or 155 II	[images.floor, here, "The floor is suity and clear"]/ [images.floor, images.full geodow, "The wall is smooth and coll?"],
198 23 197 31	[Images.soll, Boos, "Light Like a descert. Or moduld that be descert?). [Images.soll.g. or, Images.hild randow, "The wall is matching and color").
181 4: 189 5:	[inaque.html, images.html _ shadow, "At i shad on i conferential is bed"], [inaque.html, images.html _ shadow, "At is mades from strong plashine"],
150 6s 151 7s	Integer, shair left, Ware, "A shair with a sefe ushine").
152 81	[images.booksee tall, umages.tull_dadow, "Doublet" a starbul util the features books".
134 Pc	Images booksnap small, images half ghadow, "Monghebelma," a shared with reference bookset.
156 20	[Integer caller, image: half_shedow, the set1 here. for setce integers, integers, integers, itself.
158 22:	Langer deal computer, imput half plandow.
160 12	[inspec_piont, inspec_piont whole, "A specificary plant, grown here"], [inspec_piont(sol)_inspec_piont] should be a specific plant, grown here"],
162	in the second seco
164	"Excitate prime and for previous the space relation",
186 16	langer.excth, inger.athu findow, thank issues in the cartery of th
187	"A space iscues, a coll lage, out ameling it's growing deter"), [imper.phys.gr. imper.phys.gr. imped.phys.gr. imped.phys.gr. is a space is a sp
176	[inger_pipes], inger_pipes_missions_mission, [inger_pipes], inger_pipes_mission, [inger_pipes], inger_pipes_mission, [inger_pipes], inger_pipes_mission, [inger_pipes], inger_pipes_mission, [inger_pipes], inger_pipes_mission, [inger_pipes_mission, [inger_pipes_mission], inger_pipes_mission, [inger_pipes_mission], inger_pipes_mission, [inger_pipes_mission], inger_pipes_mission, [inger_pipes_mission], inger_pipes_mission, [inger_pipes_mission, [inger_pipes_mission], inger_pipes_mission, [inger_pipes_mission, [inger_pipes_mission], inger_pipes_mission, [inger_pipes_mission, [inger_pipes_mission], inger_pipes_mission, [inger_pipes_mission], inger_pipes_mission, [inger_pipes_mission, [inger_pipes_mission], inger_pipes_miss
171 19 172	[images.pipes]_image_spipes]_shadow, "Flipes for the life support systems"],
173 20: 174 fur ant	[Inseque.door_inseque.door_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inseque.coor_inse
178 21: 176 For sat	[inspection] inspection: inspection: "The willow's door." boy reasons. It requires two persons operations.").
177 221	[Inners.door_inners.door_bindow, *A locked door. It seeds * \ + LAITE_UNAL * ** access fact?],
178 23	[ineque:doc, ineque doc, whetev, X locked down. In nexts " \ + FRING[INE + "S scotes card"].
101 24	[Images.doc; images.doc; mhadow, "A loutes down. In messa " A + TRIENCY IMAE + "A access down"].
103 25:	[Inness.doc, images.doc_mbadow_ "A locked doc. File Sound form Man Diseason Convool"].
199 261 196	[Inages.doc; inages.doc; shadow, "& looked coc is the engineering bay,"],
187 271	[inages.sap, inages.idl]shadow, "The screen says that make this was decourt " \
159	+ striplants SECTOR + * // N : * + str(LANDER X) + \ N // N : * + str(LANDER X) + \ N // N : + str(LANDER X)
151 281	[inspec.ycok]arge: inspec.ycok]arge thadow. "A rock ite mears warfant femialite a determet, "the rock",
103 29: 104	[inspec.cob mail: inspec.rock_mail ishedow, "A mult be havey bleve of Morian spectra,",
150 80 196 81	[Inspes: crace, Noine, "A cluster in the plants surface"). [Inspes: freque, Noine,
107	"A loss para fanos, il sejos protect the statim from dust storms"]. Unaços, contextion, incess, contraction statedos.
199	"One of the scientific experiments. It pently without set"]. Hander, root ann. hander, root ann abadow.
201 292 34	"A follow arm, and for heavy lifting"). Inners holds, inners hold hadow, A markhing them bullet").
202 55 304 36	[Images.sink, Manas, W. make stable and and a stable", "the tags"].
205 37	"A grant globe of the planet. It penning globes from inside"].
207	"A table of representation, analyzing the planet still and there"). Honores resultion machine, here shows
205	"A reality measure it replice a credit.", "the weaking machine",
211	"A present mean to have sure subset of our alone "I, Durgen assess the ball, have a sure subset of the "D present subset".
213 41	[Inser.miston_control_dest, inser.atsiss_control_dest_madow, \
215 42	[images.buttom, images.buttom shadow,
217 217 718 43	Images with the start, images in the start start start and the start s
219 44	[inoges.window] images.fill_shadow.
221 45	"Take standow polymans a view out onto the planet surrated."), Emerger.rebet. implex.rebet.graduot. "A clausing athort, numbed off.").
222 46	Inners.ssoil, insers.ssoil makes, smaller met-up."], "A plane suitain sobol wailing met-up."],
224 471 225 481	[Insept.rocket, images.rocket, baladow, "A contepteron confic in impair"], [Insept.rocket, images.rocket, baladow, "A contepteron confic in impair"], [Insept.rocket, Conte, Mich., "Confic Elcar - context will conf"],

26	491	[images.drons, mos, "A delivery drons"],
	50:	[images.energy_ball, Hene, "An energy ball - desperate!"),
	521	[images.computer, images.computer shadow.
30	1.100	"A computer workstation, for managing space station systems."],
51	53:	[images.clipboard, None,
	141	"A disploard, bomeone has coulded on it.", "the disploard"),
84		"S piece of stiory bubble gum. Spaceberry flavour.", "bubble gum"),
	551	[images.yoys, Rone, "A coy made of fine, strong string and plastic. 5
	Hamb IQ: 56:	<pre>integrav experiments.", FLATER_RAME + "'s yoyo"], limenes.thread. Bing.</pre>
38		"A piece of fine, strong string", "a piece of string"].
39	571	[images.needle, None,
41	55 :	[insper.threaded needle, Sone,
42		"A cartus needle, spearing a length of string", "needle and string").
44	891	"The six conjector has a leak.", "a leaky air conjector"].
59	601	[images.caniater, Nome,
4.8		"It looks like the seal will hold ", "a sealed air canister"],
10		"The mirror throws a circle of light on the walls.", "a mirror"],
49	62.1	[images.bin_empty, Nume,
	631	"A fairy used bin, wate of light plastid", "a bin"), finares, bin full, Hone.
52		"A beavy bin full of water", "a bin fall of water"],
	641	[images.rags, Hone,
	65.1	[isages.hasser, Hone,
10		"A hanner. Haybe good for cracking things open ", "a hanner"],
	67:	[images.spoon, bons, "A large serving spoon", "a spoon"), [images.food pouch, Hone,
65		"A dehydrated food pouch. It needs water.", "a dry food pack"],
	681	[images.food, None,
12	69:	[images.book, Hone, "The book has the woods 'Don't Fanic' on the
6.1	cover 1	<pre>n large, friendly letters*, *s book*],</pre>
11	-701	"An HFS player, with all the latest tunes", "as HFS player"!.
64	71:	[images.lander, Mone, "The Foodle, a small space expidenation cruft. \
27	Its blan	or box has a radio sealed inside.", "the Fordle lander"],
	Foodle"	(inequestion), we lead o communications system, from the c
	791	[images.gps_module, None, "% GES Module", "a GES module"),
	T4:	[images.positioning_system, Hops, "Fart of a positioning system.]
73	751	[images.positioning system, None,
74	701	"A working positioning system", "A positioning computer"],
74	anything	. Can you sharpen them?", "hight scissors"],
	17:	[imeges.scissors, Bone,
	76.1	"Bapor-sherp scissors; Garerni!", "scarpebed scissors"], limages.oredit. Noos.
9.0		"A small coin for the station's vending systems",
11		"a station predict"],
		"This access card belongs to " + FLAYER NAME, "an access card"],
94	801	[isages.access_card, None,
		"This access card belongs to " + FRIENDL NAME, "an access card"],
87		"This access card belongs to " + FRIEND2_NDME, "am access card"]
5.0	32	
90	itens p	laver may carry = list(range(53, 53))
91	# Munices	is below are for floor, pressure pad, soil, toxic floor,
22	STARE D	
94		layer_may_stand_om = items_player_may_carry + [0, 09, 0, 40]
		layer_may_stand_om = items_player_may_carry + [0, 09, 2, 40]
		layer_may_stand_on = items_player_may_carry + [0, 09, 2, 40]
90 97		layer_may_stabd_om + items_player_may_cerry + [0, 85, 8, 40] TERT 10 TERT 10
96 97 98	ALL BOE	layar may orand on * itens player may carry + [0, 36, 2, 40] ******* ******
90 97 98 98	Al SCE	layar_may_erand_on = itens_player_may_corry = [0, 35, 2, 40] ####################################
90 97 98 98 98 01	A Scener A Scener A Scener A Scener	layer_may_stand_om = items_player_may_corry = [0, 58, 2, 40] ************************************
907 3 3 0 0 1 2 2	A Scene A S	leyer_may_scand_om = items_player_may_corry = [0, 35, 2, 40] Herrithe Herrithe Ty describes objects that cannot nows factower comma. mather: [[Goldect mather, y position, # position]] = ([153,6,2]].
90788001289	41 SCEI 41 SCEI 41 SCEI 4 Scene 4 Scene 5 S	<pre>layer_may_prand_om = items_player_may_corry = [0, 95, 2, 40] corr to for the second second</pre>
9978800128980	4 Scene 4 Scene 5 Scene 4 Scene 4 Scene 5 Scene 4 Scene 5 S	<pre>loger_may_stand_om = items_player_may_corry = [0, 95, 1, 40] ####################################</pre>
9973801239900000	41 5000 41 5000 4 50000 4 50000 4 50000 4 50000 2 61 2 71 2 81 2 91	<pre>taysr_may_srand_on = items_player_may_corry = [0, 35, 2, 40] ####################################</pre>
99999001239990000	# 5000 # 5000 # 5000 # 5000 # 5000 # 5000 # 5000 2000 201 201 301	<pre>loger_may_stand_om = items_player_may_corry = [0, 55, 5, 40] ####################################</pre>
99330122599690910	4 Source 4 Source 4 Source 4 Source 4 Source 4 Source 4 Source 4 Source 5 Sour	<pre>loger_may_scand_om = items_player_may_corry = [0, 35, 1, 40] ####################################</pre>
9933001239900000000111	41 SCE 41 SCE	<pre>loger_may_stand_om = items_player_may_coury = [0, 58, 5, 40] ******** **************************</pre>
999990023990000001112	41 SCE 41 SCE	<pre>loger_may_stand_on = items_player_may_corry = [0, 95, 1, 40] ####################################</pre>
99330012399000000111334	4 500 4 500 4 500 5 700 26: 27: 28: 29: 30: 32: 31: 32: 33: 34: 34: 34: 34: 35: 35: 35: 34: 35: 35: 34: 34: 34: 34: 34: 34: 35: 34: 35: 34: 35: 35: 35: 35: 35: 35: 35: 35	<pre>Layer_may_stand_on = items_player_may_coury = [0, 58, 5, 60] ***********************************</pre>
999990023999000000001112345	4 50en 4 50en 4 50en 5 700 26: 27: 28: 29: 30: 32: 32: 34: 35: 34: 35: 34: 35: 35: 35: 35: 35: 35: 35: 35	<pre>loger_may_prond_on = items_player_may_corry = [0, 95, 1, 40] ry describes dijects that connet news intree reme. mader: [[00/950 milder, y position, a position]] [135,531, [135,531, [135,14], [157,531, [17,552], [17,630, [1,63,14], [135,14], [17,140], [17,630, [1,63,14], [135,14], [17,140, [14,140], [14,140], [17,140, [14,140], [14,140], [17,140, [14,140], [14,140], [111,14, [15,140], [14,140], [111,14, [15,140], [14,140], [111,14, [15,140], [14,140], [111,14, [15,140], [14,140], [111,14, [15,140], [14,140], [111,14, [15,140], [14,140], [111,14, [15,140], [14,140], [111,14, [15,140], [14,140], [111,14, [15,140], [14,140], [111,14, [15,140], [14,140], [111,14, [15,140], [14,140], [111,14, [15,140], [14,140], [111,14], [111,14,14], [14,14], [14,14], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141,140], [141</pre>
9999900000000000111111111	4 Source 4 Source 5 Joos 5 Joos 26: 27: 281 291 30: 311 321 33: 341 351	<pre>Light_may_stand_on = items_player_may_coury = [0, 35, 1, 40]</pre>
999990000000000011111111111111	4 Source 4 Source 5 10000 5 10000 26: 27: 381 291 30: 311 321 33: 341 351 361	<pre>loger_may_prod_om = trems_player_may_corry = [0, 95, 1, 40] setup = setu</pre>
99999000000000000111111111110	4 Some 4 Some 4 Some 4 Some 4 Some 2 To 2	<pre>loger_may_prod_om = teems_player_may_corry = [0, 55, 5, 60] for the form of the start content means terms. meter: [000000 mathet, y position, # position]] = { [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.1], [103.6.</pre>
99999000000000000000000000000000000000	4 5000 4 5000 5 2000 2 200 2 200 2 20 2 20	<pre>loger_may_prod_om = trems_player_may_corry = [0, 55, 1, 40] regin = regi</pre>
99999000000000000000000000000000000000	4 5000 4 5000 4 5000 5 0000 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 27 26 26 27 26 20 20 20 20 20 20 20 20 20 20	<pre>loger_may_stand_om = teems_player_may_coury = [0, 55, 5, 60] for the formation of teems terms for the formation of the</pre>
999990000000000001111111111200000000000	4 502 4 502 4 502 5 1000 1 5 10000 1 5 1000 1 5 1000 1 5 1000 1 5 1000 1 5 1000 1 5 10	<pre>loger_may_stand_on = teems_player_may_coury = [0, 58, 5, 40] for the second standard state income functions income functions for the second state income functions income functions for the second state income function income function income functions for the second state income function income func</pre>
· · · · · · · · · · · · · · · · · · ·	4 5000 4 5000 4 5000 4 5000 2 4 2 7 2 8 2 9 2 8 2 9 2 8 2 9 2 8 2 9 2 8 2 9 2 8 2 9 2 8 2 9 3 1 3 0 3 3 3 4 3 55 3 66 3 7 3 55 3 66 3 7 3 7 3 8 3 7 3 7 3 7 3 7 3 7 3 7 3 7 3 7	<pre>loger_may_prod_om = teems_player_may_corry = [0, 55, 5, 60]</pre>
999990000000000011111111111120000000000	4 5000 1 5 5000 1 5 5000 1 5 5000 1 2 6: 2 7: 2 8: 2 9: 3 9: 3 1: 3	<pre>Loger_may_stand_on = teems_player_may_coury = [0, 58, 5, 60] ***********************************</pre>
99990000000000011143111112000000000000000000	4 5000 4 5000 5 1000 5 1000	<pre>loger_may_prod_om = teems_player_may_corry = [0, 55, 5, 60]</pre>
999900000000000111111111112000000000000	4	<pre>Loger_may_stand_on = teems_player_may_coury = [0, 58, 5, 60] ***********************************</pre>
	4 Sector 4 Sector 4 Sector 24	<pre>loger_may_prod_om = trems_player_may_corry = [0, 58, 1, 40] rest = rest</pre>
· · · · · · · · · · · · · · · · · · ·	4 Source 4 S	<pre>[agg_mag_mrad_on = tens_player_mag_enery = [0, 58, 5, 60] ***********************************</pre>
99980012399000000001111111111222222222222222	4 Source 4 S	<pre>loger_may_stand_on = teems_player_may_estry = [0, 55, 1, 40] regstart ry describes dijects that identit mere between imme. ry describes dijects that identit mere between imme. (105, 6, 21, (105, 4, 1, 105, 1, 10, 1, 10, 1, 10, 1, 10, 1, 10, 1, 10, 10</pre>
	4 Section 2015 4 Section 2015 4 Section 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015	<pre>inpre_may_stand_on = terms_player_may_coury = [0, 55, 5, 60] for the second state of the second model and th</pre>
	444 SCE 45 SCENE 4 SCENE 4 SCENE 4 SCENE 264 274 264 274 264 274 264 274 264 274 264 274 264 274 264 274 274 274 274 274 274 274 27	<pre>loger_may_stand_on = teems_player_may_coury = [0, 58, 5, 40] representation dejusts that income news incluses income. representation dejusts that income news incluses income. (103,6,31, 103,6,31, 103,6,31, 101, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31, 103,6,31,</pre>
· · · · · · · · · · · · · · · · · · ·	4 Section 4 Section 4 Section 4 Section 4 Section 4 Section 5 Section	<pre>iquer_may_stand_on = teems_player_may_coury = [0, 55, 5, 60] for the second state of the state of the second state of th</pre>

11 CheckBus = 0
12 CheckBus = 0
13 CheckBus = 0
14 CheckB A HARE NAP /// identification // ide isi generate map(): * This function makes the map for the oursest roles, * This functions makes the map for the oursest roles, states you between the state of the state of the state states on the state of the state of the state of the room_news = room_dates[] room_news = room_dates[] room_news = room_dates[] room_news = room_dates[] floor_type = get_floor_type()
fl current_room in range(1, 11);
botton_edge = 3 faull
side_edge = 2 faull
if current_room in range(21, 26);
botton_edge = 1 foull
side_edge = 2 faull if current_room > 25: bottom_edge = 1 #xel1 side_edge = 1 #wal1 f Create top line of roum map, roam map=[linds_maps] ' room_width] room_maps.room_width] room_maps.ropend[linds_cont_______ idd] bottom_maps.room_maps. * (filds_cont____it) ' room_width - 2) + [aids_adaps]) room_maps.room_maps.room_width) i Add doorways. middle_row = int(room_height / 2) middle_column = int(room_width / 2) middle_couses if room_data[4] + 1 if exit as right of this room
room_papelind.dim row[room_width - 1] = floor_type
room_map[middls_row+1][room_width - 1] = floor_type
room_map[middls_row+1][room_width - 1] = floor_type room_mappingdis_row-lifesem_wints - 1 + fines_type fources_toos NMP HTOE: 1 = 1 fif cond in soc on left of map room_too_left: = GME_TAR(ourrent_room - 1) if foroms too left has to set a right each, add left want in this room if room_to_left(i) room_maphicadle_row + 11(0) = floor_type room_maphicadle_row + 11(0) = floor_type sf soom deta[3]: # If exit at top of this soom room nep[0][naddle_column] = floor_type room_nep[0][naddle_column + 1] = floor_type room_nep[0][naddle_column - 1] = floor_type toom_map(u)[source_course -1] * find=type ff ourtest_toon * MAR SIZE ~ MAW FUTGH + if from 12 book on bolton for ioon boltow = GMAR BAR(ourtest_roomARA #UTGH) 1 fiftoom televour mas too paths, and sets if a boots of Shin one (f room_below[31] room_map(room_below[-1][baldids_column + i] * filor_type room_map(room_below[-1][baldids_column + i] * filor_type if current_from is memory: during accessive schemery(current_room): schemery_wimmer = this schemery(0) schemery_w = this schemery(1) schemery_k = this schemery(1) schemery_k = this schemery(1) schemery_k = this schemery(1) = schemery_number image_here = objects[doenery_number][0] image_width = image_here.get_width() image_width_in_tiles = int(image_width / TILE_STIK)

for tile_number in range(l, image_width in_tiles): room_map(scenery_y)[scenery_x + tile_number) = 255

357	center y = int(HEIGHT / 2) # Center of game window
458	room pixel width = room width + TILE SIZE # Size of room in pixels
460	room pixel height = room height * TILE SIZE
461	top_left_x = center_x - 0.5 * room_pixel_width
461	top_left_Y = (center_Y - 0.5 * room_pimel_height) + 110
365	
4675	**********
900	** GANE LOOP **
468	
-169	def start_room())
478	show_text ["Tou are here: " + room_name, 0}
871	tet ore locally
478	global player x, player y, current room
474	global from player w, from player y
975	ginnal player image, player image snadow
477	rinhal player offset x, player offset y
±70	global player_frame, player_direction
279	TE TARA CONTI
401	return
902	
20.0	if player frame > 0:
385	cime.sleep(0.05)
404	if player_frame -= 1:
487	player_frame = 0
386	player offset v = 0
390	
491	Frame player's current position
49.2	old player y = player y
394	
399	<pre># nove if key is presed</pre>
407	if keyboard.right:
490	from_player_m = player_m
955	from_player_y = player_y
601	player A +- 1 player direction = "right"
502	player_frame = 1
203	slif beyboard.left: #alif stops player making diagonal movements
	from player v = player v
504	player x -= 1
507	player_direction = "left"
500	player frame = 1
51.0	from player x = player x
-111	from_player_y = player_y
212	player_y 1
214	player_frame = 1
-515	ell! keyboard.down)
-516	from_player_x = player_x
318	nister v te 1
31.0	player direction = "down"
63.0	player_frame = 1
600	# check for aniting the room
123	if player a room width: + through door on RIGHT
124	<pre>#clock.unschedule(hazard.move)</pre>
526	generate ward
527	player_x = 0 # enter at left
526	player_y = int(room_height / 2) # enter at door
530	fatart room()
697	zeturn-
532	If player a se all & through door on LETT
224	#clock.unschedule(hazard move)
635	current_room -= 1
335	player x = room width = 1 # enter at right
230	player y = int(room_height / 2) # enter at door
239	player_frame = 0
1450 541	recurn ()
542	
363	
10.00	if playery room height: * through hour at Borrow
	<pre>st player, y == room Meight: # through door at BOTTOM #clock.unschedels(hensti novs) nurrett room == MAP NIDTH</pre>
040 044	<pre>#LE player y ** room >* lot i through door at BOTTOM #clock unstatheding (instant nove) durrent room ** NAP_WIDTH generate_wop()</pre>
五4年 五4年 547	<pre>if player y == room_Maint + through door at BOTTOM felater numericals (meant nove) ourrent room = MAP_RIDTB generate xep () player y = 0 + enter at top</pre>
545 547 547 547	<pre>if player_y == room_paight f through note at DOTTON filter.unterheding (neuronal notes) currents_room NAP_RIDIE neuronal room_paight to p player_y = 1 in stars to p player_y = 1 in through the top player_y = 1 introom_paight (2) i water at door player_y = introom_paight (2) i water at door pl</pre>
549 548 547 348 348 830	<pre>if place</pre>
545 546 547 547 547 547 547 547 547 547 547 547	<pre>if playery = room_paight f theogen nore at DUTON flayery = room_paight f theory processing and the playery = room for the playery = f and the top playery = f intro top playery = introom_width / 2) f enter at door playery = rest = 0 fract(_cost) secure</pre>
540 547 547 547 540 540 880 851 552 552	<pre>if place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_place_plac</pre>
545 547 547 547 547 540 830 830 830 831 553 554	<pre>if playery == room parameter = through more at DUTION protecting year to be attract at the protecting year () playery y = 0 = # state att top playery y = 0 = # state att top playery x = int procey, width / 2) # enter at the player y = 0 = # state att top playery y = 0 = # state att top playery x = int procey, width / 2) # enter at the filtery unarbediate (through door att TOP filtery unar</pre>
545 547 547 547 547 547 547 547 547 547	<pre>if plots_v = isotopic to be a control outcome (v = v = v = v = v = v = v = v = v = v</pre>
545 547 547 547 547 547 547 547 555 555	<pre>if Ploymery == icong Mainter = interact = correct corrected processing and the interact = correct playmery == 0 = enter act top playmery = 0 = enter act top playmery == 0 = enter act top playmery == 0 = enter act top for the enter act to the enter act better playmery = for maintage = 1 = for the act to better playmery = for maintage = 1 = for the act to better playmery = for maintage = 1 = for the enter act better</pre>
949 946 947 947 947 947 947 947 950 951 955 955 956 956 956 957 950	<pre>if Picture; we include the interval in the picture; outcomes(); non = nAR_NINTER picture; n</pre>
949 946 947 947 947 947 940 953 953 953 955 955 955 955 955 955 955	<pre>if Ploymety == icong Mainter = intendent note at Dorton granter = [] yoon == Add X turber granter = [] yoon == Add X turber granter = [] yoon == Add X turber player y = 0 = attract stop player y = 0 = attract stop player y = 0 = attract stop attract = 0 attract = 0 attract = 0 attract = 0 attract = 0 player y = cong state (neared news) opticstop yee () player y = scom state = 1 & attract st bottom player y = scom state - 1 & attract st bottom player y = scom state - 1 & attract st bottom player y = scom state - 1 & attract st bottom player y = scom state - 1 & attract st bottom player y = scom state - 0 player - 0 player y = scom state - 0 player y = s</pre>
040 040 040 040 040 050 050 050 050 050	<pre>if PloyIng 'we icongraduate' i through note at Durice outerest', toos -m ARE XLUTE generating yes 0 i estate at top playery ye 0 i estate at top of state (see a second by 2) i estate at door player frame.</pre>
040 040 040 040 040 040 040 050 050 050	<pre>if Ploysely == icody_states = i decody house at Dorton ourcest, you == house xtuDB yencedocy_mp() playsely = 0 = state at top playsely = 0 = state xtuDB sectors ourcest, you == house top playsely = scale at the state at bottom playsely = scale at th</pre>
040 040 040 040 040 040 040 040 050 050	<pre>if PloyIng == icong_limits = 1 through nois at Duricon outcomest, from == nAge Visition generating ways = 0 = status = 1 to 0 player great = 0 = status = 2) = status = t dont player great = 0 findst from == nAge Visition = t TOP findst from == nAge Visition = t TOP findst unachedist (instatut move) outcomest = non-maked = 1 f enter at bottom player great = 0 findst great = 0</pre>
0+0 0+0 0+0 0+0 0+0 0+0 0+0 0+0	<pre>if ployer_j == icod_planet = i decode note at Dorton outreet, you == ndg that the icod planet = icod planet planet, y = 0 = ndts = to to planet, y = 0 = ndts = to to planet, y = 0 = ndts = to to planet, y = 0 = ndts = to to planet, y = ndt = ndts = to to planet, y = ndt = ndts = to to planet, y = ndt = ndts = to to planet, y = ndt = ndts = to to planet, y = ndt = ndts = to to planet, y = ndt = ndts = to to planet, y = ndt = ndts = to to planet, y = ndt = ndts = to to planet, y = ndt = ndts = to to planet, y = ndt = ndts = to to planet, y = ndt = ndt = ndts = to to planet, y = ndt = ndt = ndts = to to planet, y = ndt = ndt = ndts = to to planet, y = ndt = ndt = ndts = to to planet, y = ndt = ndt = ndts = to to planet, y = ndt = ndt = ndts = to to planet, y = ndt = ndt = ndt = ndt planet, y = ndt = ndt planet, y = ndt planet, y</pre>
	<pre>if PloyIng = icongraduate ; through more at Dorton outcome; you = naky through some at Dorton generate you = internation / 2) f enter at done player (rese = 0) f clark unschediel (hased more) currents (rese = 0 player y = enternation / 2) f enter at done player y = come maky HIDTB player y = enternation / 2) f enter at done player y = come maky HIDTB player (rese = 0 f for the player is standing somewhere they shoulds't, more them back. f f come maky flayer y([flayer x] ms is litere_player pay_stand_on 3) </pre>
	<pre>if ploysty == non_plants: = interapt none at Dorton outrest, non == NAR_NINTS gureestor.pop() playsty == 0 = state st top secure fit playsty == 0 = st 4 through does at TOE secure gureestor.pon = NAR_NINTS guree</pre>
	<pre>if Ploying' = icongraduate / through none at Dorton outcome / too = nAR / LUTRY generating and the icongraduate / too player / 0 { state at top player / 0 = nAR / LUTRY currents / toos = nAR / LUTRY player / * = non package = 1 f enter at better player / toos = 0 f state / too f state / too f</pre>
A 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1	<pre>if Player_1 == icod_player_1 = isocapi none at Dorton outcomerty (non == hak_t toTR) gurcestor_map() player_2 = (= state = st top player_2 = (= state = state = st top player_2 = (= state = state = st top player_2 = (= state = state = st top st top player_2 = (= state = state = st top player_2 = (= state = state = st top player_2 = (= state = state = state = st top player_2 = (= state = state = st top player_2 = (= state = state = st top player_2 = (= state = state = st top player_2 = (= state = state = st top player_2 = (= state = state = st top player_2 = (= state = state = st top player_2 = (= state = state = st top player_2 = (= state</pre>

if plays_direction == "cipit" edu plays_drame > 0: plays_drame_s = -1 = (0.25 * plays_drame) plays_drame_s = -1 = (0.25 * plays_drame) plays_drame_s = -1 = (0.25 * plays_drame) if plays_direction = "up" edu plays_frame > 0 plays_drame_s = - "up" edu plays_frame > 0 plays_drame_s = - "dime" and plays_frame > 0 plays_drame=s = "dime" and plays_frame > 0 plays_drame=s = "dime" and plays_frame > 0 ** DISPLAY ** def draw_inage(ixage, y, x): sciten.blit((top_left_x + (x * TILE_ST2E), top_left_x + (y * TILE_ST2E), _ image.get_baight()) draw_shadow(image, y, w);
screen.blit(image, (top_left_x + |x * TILE_STIE), (top_left_y + |y + TILE_STIE))) st draw_playse(); playse_inage = FLATE [playse_direction] [playse_frame] draw_inage playser_inage, playse_y + playse_offset_y, playse_inage andow = fuller_setMod([playse_direction][playse_frame) draw_bhadwe(playse_int = playse_offset_w) draw_bhadwe(playse_int = playse_offset_w) f draw() | if game_over: i(line the generation state to an = fact(0, 55); (00), (00)) serven-desc.filed_res(how, RED box = fact(0, 0), (80; com_left_y + (room_height - 1)*30)) screen.surface.set_clin(box) foor_the screen.surface.set_clin(box) F Preserve pair in cosh 24 is added here, so propa can go in top of it. as carrent none (19)(10), 5, 21 Lange (none) = cosh (19)(10), 5, 21 Lange (none) = cosh (19)(12) (1 insec (none) = 0) discost (none) = (1 ange (none) = (1 ange (none)) discost (none) = (1 ange (none)) = (1 ange (none)) discost (none) = (1 ange (none)) = (1 ange String in support in them plays may be a set by wide objects. if a in support of them plays in them plays may read on y wide objects. if them, here one in them plays may read on + 1231; in any e object (them_max)[0] draw_image(image, y, w) slats draw_shadow(shadow_image, y, w) if (player_y == y): draw_player() screen.surface.set_olsp(None) if (player y == room height - 2 and room map(room height - 1][player m] == 1 and wall transparency frame < 611 wall_transparency_frame += 1 + fade wall out. retroit templine = (15, 50]
box = Boxt((0, text)line[sing_number]), (000, 35))
screen.csr.(110d_sec(box, b.COM)
screen.csr.(110d_sec(box, b.COM)
screen.csr.(110d_sec(box, b.COM)), (20, 5est_lines(line_number)), color=GREEN) AT START DE doi generate map])
700 glock.acheoule_interval(game_icop, 0.08)
701 clock.acheoule_interval(adjust_wall_transparency, 0.06)