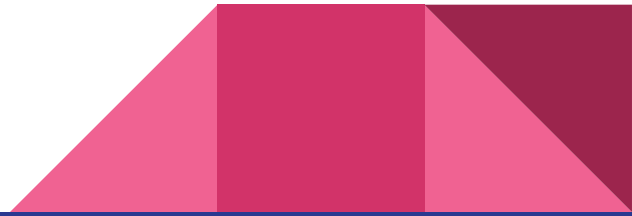
Project 1

Alien Invasion

12. A Ship that Fires Bullets

12-1. Blue Sky: Make a Pygame window with a blue background.

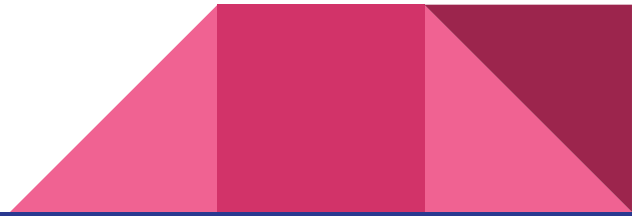
12-2. Game Character: Find a bitmap image of a game character you like or convert an image to a bitmap. Make a class that draws the character at the center of the screen and match the background color of the image to the background color of the screen, or vice versa.



12. A Ship that Fires Bullets

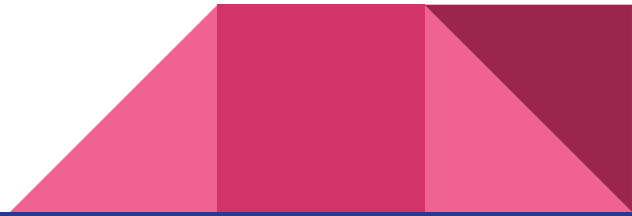
12-3. Rocket: Make a game that begins with a rocket in the center of the screen. Allow the player to move the rocket up, down, left, or right using the four arrow keys. Make sure the rocket never moves beyond any edge of the screen.

12-4. Keys: Make a Pygame file that creates an empty screen. In the event loop, print the event.key attribute whenever a `pygame.KEYDOWN` event is detected. Run the program and press various keys to see how Pygame responds.



12. A Ship that Fires Bullets

12-5. Sideways Shooter: Write a game that places a ship on the left side of the screen and allows the player to move the ship up and down. Make the ship fire a bullet that travels right across the screen when the player presses the spacebar. Make sure bullets are deleted once they disappear off the screen.



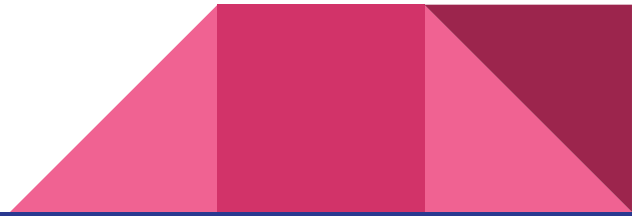
13. Aliens!

13-1. Stars: Find an image of a star. Make a grid of stars appear on the screen.

13-2. Better Stars: You can make a more realistic star pattern by introducing randomness when you place each star. Recall that you can get a random number like this:

```
from random import randint
random_number = randint(-10,10)
```

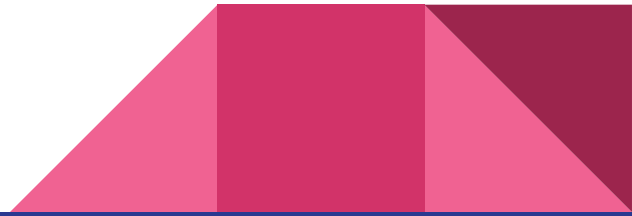
This code returns a random integer between −10 and 10. Using your code in Exercise 13-1, adjust each star's position by a random amount.



13. Aliens!

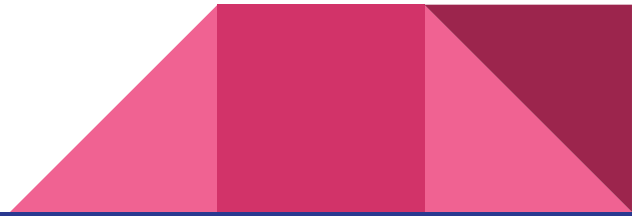
13-3. Raindrops: Find an image of a raindrop and create a grid of raindrops. Make the raindrops fall toward the bottom of the screen until they disappear.

13-4. Steady Rain: Modify your code in Exercise 13-3 so that when a row of raindrops disappears off the bottom of the screen, a new row appears at the top of the screen and begins to fall.



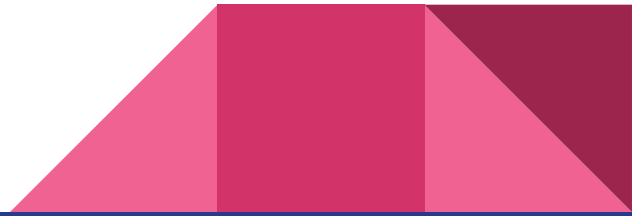
13. Aliens!

13-5. Catch: Create a game that places a character that you can move left and right at the bottom of the screen. Make a ball appear at a random position at the top of the screen and fall down the screen at a steady rate. If your character “catches” the ball by colliding with it, make the ball disappear. Make a new ball each time your character catches the ball or whenever the ball disappears off the bottom of the screen.



13. Aliens!

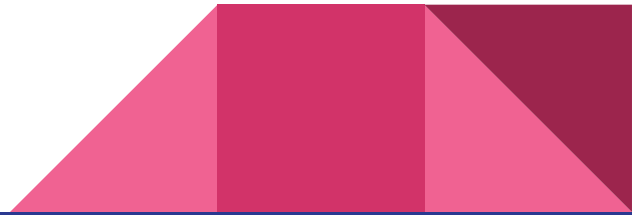
13-6. Game Over: Using your code from Exercise 13-5 (page 284), keep track of the number of times the player misses the ball. When they've missed the ball three times, end the game.



14. Scoring

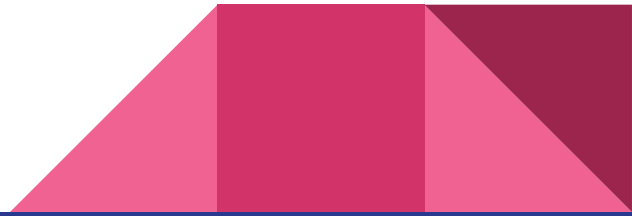
14-1. Press P to Play: Because Alien Invasion uses keyboard input to control the ship, it's best to start the game with a keypress. Add code that lets the player press P to start. It may help to move some code from `check_play_button()` to a `start_game()` function that can be called from both `check_play_button()` and `check_keydown_events()`.

14-2. Target Practice: Create a rectangle at the right edge of the screen that moves up and down at a steady rate. Then have a ship appear on the left side of the screen that the player can move up and down while firing bullets at the moving, rectangular target. Add a Play button that starts the game, and when the player misses the target three times, end the game and make the Play button reappear. Let the player restart the game with this Play button.



14. Scoring

14-3. Challenging Target Practice: Start with your work from Exercise 14-2 (page 298). Make the target move faster as the game progresses, and restart at the original speed when the player clicks Play.



14. Scoring

14-4. All-Time High Score: The high score is reset every time a player closes and restarts Alien Invasion. Fix this by writing the high score to a file before calling `sys.exit()` and reading the high score in when initializing its value in GameStats.

14-5. Refactoring: Look for functions and methods that are doing more than one task, and refactor them to keep your code organized and efficient. For example, move some of the code in `check_bullet_alien_collisions()`, which starts a new level when the fleet of aliens has been destroyed, to a function called `start_new_level()`. Also, move the four separate method calls in the `__init__()` method in Scoreboard to a method called `prep_images()` to shorten `__init__()`. The `prep_images()` method could also help `check_play_button()` or `start_game()` if you've already refactored `check_play_button()`.

Note: Before attempting to refactor the project, see Appendix D to learn how to restore the project to a working state if you introduce bugs while refactoring.

14-6. Expanding Alien Invasion: Think of a way to expand Alien Invasion. For example, you could program the aliens to shoot bullets down at the ship or add shields for your ship to hide behind, which can be destroyed by bullets from either side. Or use something like the pygame.mixer module to add sound effects like explosions and shooting sounds.