**Pong V2 Reflection Activity**

Q1 To bounce the pong ball off the paddle, we can use collidepoint() to check if the center of the ball has intersected the paddle. For this question, analyze this Python statement:

paddle.collidepoint(center)

For each token used in the statement, identify the token kind and, if the token is an identifier, identify the type of object the identifier is bound to. If the type is function, indicate whether it is a user-defined function, user-defined method, library function or library method.

|  |  |  |
| --- | --- | --- |
| **Token** | **Token Kind** | **Type** |
| paddle | identifier | pygame.Rect |
| . | delimiter |  |
| collidepoint | identifier | Library method |
| ( | delimiter |  |
| center | identifier | list |
| ) | delimiter |  |

What type of object must collidepoint() be applied to?

|  |
| --- |
| pygame.Rect |

Q2 Which of the following are valid calls to collidepoint()? Mark them Y for yes valid or N for not valid. For each invalid answer, choose from the following options to indicate the reason it is invalid:

1. Incorrect arguments used for collidepoint().
2. Incorrect arguments used for pygame.Rect().
3. collidepoint() applied to an object with the wrong type.

|  |  |  |
| --- | --- | --- |
| **Object** | **Y/N** | **Reason #** |
| rect = pygame.Rect(90, 90, 60, 80)  rect.collidepoint() | N | 1 |
| rect = pygame.Rect(90, 90, 60, 80)  rect.collidepoint([20, 20]) | Y |  |
| rect = [100, 100, 50, 60]  rect.collidepoint([20, 20]) | N | 3 |
| pygame.Rect(25, 90, 3, 4).collidepoint([20, 20]) | Y |  |
| rect = pygame.Rect(120, 450, 300, 70)  rect.collidepoint(20, 20) | Y |  |
| pygame.Rect.collidepoint([20, 20]) | N | 2 |
| rect = pygame.Rect((70, 40), (100, 130))  rect.collidepoint(20, 20) | Y |  |