```python
 1 """
 2 A really simple domino game.
 3 """
 4
 5 import boneyard as yard
 6 # boneyard must have these functions: create, tiles_remaining
   , and draw
 7
 8 import domino as doms
 9 # domino must have these functions: create, as_str, get_left,
   get_right
10
11
12 the_yard = yard.create()
13 game_over = False
14
15 while not game_over:
16     if yard.tiles_remaining(the_yard) == 0:
17         print('Ran out of dominoes')
18         game_over = True
19     else:
20         input('Press return to continue')
21         tile = yard.draw(the_yard)
22         print('Got tile %s' % (doms.as_str(tile)))
23         if doms.get_left(tile) == 6 or doms.get_right(tile
   ) == 6:
24             print('Got a SIX!!!')
25             game_over = True
26
27 print("Game Over.")
28
29 """
30 I affirm that I have carried out the attached academic
   endeavors with full academic honesty,
31 in accordance with the Union College Honor Code and the course
    syllabus.
32 """
33
```

```python
1  """
2  A module for main file that creates the [ | ] format domino
   with left and right values
3  """
4
5  def create(left, right):
6      """
7      Create a representation for a domino with the given left
   and right values
8      :param left: an integer for the left value
9      :param right: an integer for the right value
10     :return: a tuple of left and right values
11     """
12     domino = (left, right)
13
14     return domino
15
16  def get_left(domino):
17     """
18     Get the left value of a given domino
19     :param domino: a tuple of left and right values
20     :return: an integer for the left value
21     """
22     left = domino[0]
23
24     return left
25
26  def get_right(domino):
27     """
28     Get the right value of a given domino
29     :param domino: a tuple of left and right value
30     :return: an integer for right value
31     """
32     right = domino[1]
33
34     return right
35
36  def as_str(domino):
37     """
38     Get the [ | ] format domino with left and right values
39     :param domino: a tuple of left and right values
```

```
40        :return: a string with left and right values
41        """
42        return "[%d | %d]" % (get_left(domino), get_right(domino))
```

```python
"""
Models a boneyard -- a pile of dominoes.
"""

import domino as d
import random

def create():
    """
    returns a pile of dominoes containing
    one copy of every possible domino
    """
    yard = []
    for i in range(0,7):
        for j in range(0, 7):
            tile = d.create(i, j)
            yard.append(tile)
    return yard

def draw(boneyard):
    """
    removes a random domino from the boneyard
    and returns it
    """
    n = random.randint(0, len(boneyard)-1)
    return boneyard.pop(n)

def tiles_remaining(boneyard):
    """returns the number of tiles left in the yard"""
    return len(boneyard)
```