

```

def roll_dice(probability_of_six=1/6):
    """
    Return a random number on a dice based on a given probability of rolling 6.
    If the die is rigged, the probability of the remaining rolls is uniform.
    """
    if random() < probability_of_six:
        return 6
    return randint(1, 5)

def game(board_size, print_state=True, probability_of_six=1/6):
    """
    Play a game that simulates the movement of a piece on a game plan of size
    <board_size>. If <print_state> is True, print the roll and state of the
    piece after each round. Otherwise, return the number of rounds needed to
    reach the goal. You can also set the <probability of rolling 6>.
    """
    game_round = 0
    position = 0
    while position < board_size:
        game_round += 1
        if print_state:
            print("Round", game_round, "-- Roll:", end=" ")

        move_forward = 0
        while move_forward % 6 == 0:
            roll = roll_dice(probability_of_six)
            if print_state:
                print(roll, end=" ")
            move_forward += roll
            if move_forward == 18:
                # Unlucky roll 666
                position = 0
                move_forward = 0
                break
            if move_forward % 6 == 0 and position + move_forward > board_size:
                # Don't throw again if you are already out of bounds after 6
                break

        if position + move_forward <= board_size:
            position += move_forward
        if print_state:
            print("\nI am on position", position)

    if print_state:
        print("Game finished in round", game_round)
    else:
        return game_round

```