

0 references

```
public void check_player_will_start_running()
{
    CharacterInputController C = new CharacterInputController();

    C.StartRunning();

    Assert.IsTrue(C.m_IsRunning);
}
```

0 references

```
public void check_player_will_stop_running_after_stopMove()
{
    CharacterInputController C = new CharacterInputController();

    C.StopMoving();

    Assert.IsFalse(C.m_IsRunning);
}
```

0 references

```
public void check_player_is_sliding_while_running()
{
    CharacterInputController C = new CharacterInputController();
    C.m_IsRunning = true;
    C.m_Sliding = false;
    C.Slide();
    Assert.IsTrue(C.m_Sliding);
}
```

0 references

```
public void check_player_is_sliding_while_running_and_sliding()
{
    CharacterInputController C = new CharacterInputController();
    C.m_IsRunning = true;
    C.m_Sliding = true;

    C.Slide();
    Assert.IsTrue(C.isSliding);
}
```

0 references

```
public void check_player_is_NOT_still_jumping_after_slide()
{
    CharacterInputController C = new CharacterInputController();
    C.m_IsRunning = true;
    C.m_Sliding = false;
    C.m_Jumping = true;

    C.Slide();
    Assert.IsFalse(C.m_Jumping);
}
```

0 references

```
public void check_player_is_jumping_while_running()
{
    CharacterInputController C = new CharacterInputController();
    C.m_IsRunning = true;
    C.m_Sliding = false;
    C.m_Jumping = false;

    C.Jump();
    Assert.IsTrue(C.m_Jumping);
}
```

0 references

```
public void check_lane_change_right()
{
    CharacterInputController C = new CharacterInputController();

    C.m_IsRunning = true;
    C.m_CurrentLane = 1;

    C.ChangeLane(1);

    Assert.AreEqual(m_CurrentLane, 2);
}
```

0 references

```
public void check_lane_change_left()
{
    CharacterInputController C = new CharacterInputController();

    C.m_IsRunning = true;
    C.m_CurrentLane = 1;

    C.ChangeLane(-1);

    Assert.AreEqual(m_CurrentLane, 0);
}
```

0 references

```
public void check_lane_change_right_out_of_bounds()
{
    CharacterInputController C = new CharacterInputController();

    C.m_IsRunning = true;
    C.m_CurrentLane = 0;

    C.ChangeLane(-1);

    Assert.AreEqual(m_CurrentLane, 0);
}
```

0 references

```
public void check_lane_change_left_out_of_bounds()
{
    CharacterInputController C = new CharacterInputController();

    C.m_IsRunning = true;
    C.m_CurrentLane = 2;

    C.ChangeLane(1);

    Assert.AreEqual(m_CurrentLane, 2);
}
```

0 references

```
public void check_player_is_NOT_still_sliding_after_jump()  
{  
    CharacterInputController C = new CharacterInputController();  
    C.m_IsRunning = true;  
    C.m_Sliding = true;  
    C.m_Jumping = false;  
  
    C.Jump();  
    Assert.IsFalse(C.m_Sliding);  
}
```