**Few important points to be considered whilst drafting a test strategy for game DUCK Hunt.**

1. It starts with requirements workshop. A thorough understanding of Storyboard, Game Architecture, Game Features, Characters, Rules and Game stages/levels helps US to create an effective game testing strategy.
   1. In this case there are no clear requirements about the hardware configuration
   2. No information about the customer deployments(the real time customer installation details)
   3. The compatibility of the system across various platforms
   4. No details about the screen resolution
   5. The user level permissions are not clearly defined whilst passing through webservices

2.       Game test strategy preparation is the next phase. The Test Strategy document must include sections like Timeline, Testers, Number of Testing Cycles, In-scope, Out-scope, Types of Testing to be conducted, Risk based Test analysis, Service Level Agreements, Risks and Mitigation, Defect Logging Process and Reporting Process. For a typical action game,

* + In-scope tests could be Functionality, Performance, Compatibility, etc.
  + Validation of hardware components such as Light gun, Console, other devices etc.

3.       In third phase, testers should design test cases. Testers should create positive as well as negative test cases.  Following are some effective test case design techniques:

* + Critical Path Test is a technique to identify the critical paths in the game.
  + In contrast to Happy Path testing, Exception Path Testing helps to document alternate paths and exceptions path.
  + Also, boundary value analysis, equivalence partitioning, and error guessing are the traditional techniques of black-box testing.

4.       Executing game test cases in the identified environment with different software and hardware configuration reveal more defects.

* + Alpha, Beta Testing and respective age group testing helps to improve the quality of the game further.
  + Content Testing plays a significant role in game testing. It validates the following aspects of the game:

1. Is the content appropriate for the gamers?

2. How easy it is to understand the content?

5.       Test results should be recorded as a movie or as a series of screenshots. This helps the development team to review the behavior effectively.

6.       All the defects should be logged, prioritized, categorized and tracked effectively. Defect Logs help developers for the next title work.

**Answer ONE:**

At very high level I would draft a test strategy as below for the set of product requirements given:

1. Black Box Testing considering the following points:
   1. Test the GUI and Usability features of Game
   2. Familiar with the game rules and test the game play against these rules.
   3. Test loading/saving from a game save device using different hardware (e.g. hard drive or a memory card) and ensure the correct messages are displayed on the screen.
   4. Test for the multi-player mode
   5. Test for localization
2. Look and Feel
   1. Verify the GUI components(Ex: Ducks, Trees, Grass, etc)
   2. Verify the Usability of the components defined on the screen(Ex: Background colour, etc)
   3. Test for clipping
   4. Examine the overlap (where a semi-transparent object shown on top of another solid colour object and/or the background), and check if the overlap is appropriate in terms of size, placement, the purpose and the information that is provided).
   5. Screen resolution on different devices
      1. Must examine the entire screen and not just a small part of it.
   6. Scope of the testing: The testable items for this requirement are GUI components, Screen Resolution and look & feel
3. Functionality of the light- gun
   1. System testing: Verify whether the functionality of the gun is working properly
   2. System integration testing : Verify whether the expected behaviour on the screen is in sync with the light gun functionality
   3. Scope of the testing: The testable items for this requirement are light gun and the screen response
4. Functionality of the sensor
   1. System testing: Verify whether the functionality of the sensor is working properly
   2. System integration testing : Verify whether the expected behaviour on the screen is in sync with the light gun functionality (i.e registration of the hit)
   3. Scope of the testing: The testable items for this requirement are light gun and the screen response
      1. Points 1,2 and 3 can be derived based on the functional testing and they can be achieved by equivalence partition and boundary value testing design techniques
      2. Point 4,5,6,7 and 8 can be achieved by System testing

**Answer TWO:**

Not enough information to log a bug as more the information we provide the easier way to get it fixed saving money/time:

I would definitely question/query more on the information I received from the customer service representative before I log bug.

Following are the Bug steps:

**Bug subject:** UI discrepancy and user cannot hit the ducks on the screen

**Pre conditions**: The light-gun, sensor and screen should be in working condition, Power socket should be working properly.

**Environment details**: Production

**Steps to reproduce:**

1. Pull the light trigger pointing at the ducks on the screen
2. etc..

**Expected result:** The ducks should be hit displaying the text ‘Nice shooting!’

**Actual result:** The user is getting the white box flash around the ducks and the dog laughs

**Attachments:** GUI Screenshots, server log console, etc

**QA Observations:**

1. The actual issue seems to be missing the black colour when the user pulls the trigger (As when the user pulls the trigger the screen should display as black with white rectangles in the position of the ducks).
2. The usability of the application needs to be reviewed as its hurting the customer s feelings.

**Answer THREE:**

Assuming there are 3 users with permissions as below:

1. Read, write and execute
2. Read, write
3. Read only

Status 200: (With user A- RWE)

1. Enter valid name, my app and domain name and a valid permissions
2. Verify whether the user can do CRUD operations without any issues

Status 400: (With User A, B and C)

1. Enter valid name , Invalid app name/domain name and valid permissions
2. Enter
3. Enter invalid name, invalid app name and invalid permissions
4. Enter valid name, valid app name and invalid permissions
5. Enter invalid name, valid app name and invalid permissions

Status 403: (With User B and C)

1. Verify with a valid authorization header
2. Verify with an invalid authorization header
3. Verify with an available user
4. Verify whether the ownership of the user is valid

Status 404(With User C)

1. Verify with an invalid App ID
2. Verify with invalid data
3. Verify with invalid user name and password

Status 500(Verify with User A , B and C)

1. Verify when the server is down or when the code in web services has been changed

**Testing the below factors as part of Performance testing:**

1. Test loading/saving from a game save device (e.g. hard drive or a memory card) and ensure the correct messages are displayed on the screen.
2. Ensure a “game load” or “data load” message and an appropriate “loading” indication displayed on the Loading Screen.
3. Ensure the loading time is acceptable
4. Test the game for memory leak or memory overload by leaving the game on/running.
5. Load a game without a MEMORY card, or pulling out the MEMORY card during game loading.