

CHECKLIST OF PROJECT

Test Cases

- Open the terminal at your main folder (the folder that your code needs to be compiled and run)
- Run the following commands:
 - `ls -lR` (Unix (MacOS, Linux etc.) or `dir -S` (NT (Windows))) (Must show contents of your assets folder, if not then try to show contents of your assets folder by yourself)
[This functionality works like a charm.](#)
 - `javac -version` (must be resulted same as “javac 1.8.0_xxx”)
[This functionality works like a charm.](#)
 - `java -version` (must be resulted same as “java version “1.8.0_xxx””)
[This functionality works like a charm.](#)
- Compile your code and then run it.
[This functionality works like a charm](#)
- Show that the game opens without any crash. Your system must also read the contents of the assets folder by itself.
[This functionality works like a charm.](#)
- Show the title and icon of your game.
[This functionality works like a charm.](#)
- Show that music is playing in loop at title screen.
[This functionality works like a charm.](#)
- Show that user can exit with ESC key and open the game again.
[This functionality works like a charm.](#)

CHECKLIST OF PROJECT

- Show that user can go to the background selection screen by pressing the ENTER key, change background and crosshair, then press ESC key to go back to the title screen.
[This functionality works like a charm.](#)
- If your code can go from background selection screen to title screen, press ENTER again to go background selection screen again, and show that background screen resets itself.
[This functionality works like a charm.](#)
- Show that music is still playing in loop where it has been left during background selection screen.
[This functionality has some bugs.](#)
- Show that arrow keys can navigate between backgrounds and crosshairs.
[This functionality works like a charm.](#)
- Select one of the crosshair and background and then proceed with ENTER key.
[This functionality works like a charm.](#)
- Show that your game will not start until intro effect finishes.
[This functionality works like a charm.](#)
- Show that cursor is changing with crosshair, and it returns normal after moving the mouse out of the window and returning to crosshair when it is in the game window.
[This functionality works like a charm.](#)
- Show that playing again from the end of the game does not plays the sound effect again.
[This functionality works like a charm.](#)
- Play each level again and press the ESC key to return to the title screen.
[This functionality works like a charm.](#)
- Start the game with different crosshair and background (if your game could not be succeeded from the step above, just close the game and open it again).
[This functionality works like a charm.](#)

CHECKLIST OF PROJECT

- Play the game and lose one of the levels, then press ENTER to play again.
[This functionality works like a charm.](#)
- Play the game and lose one of the levels, then press ESC to return to the title screen.
[This functionality works like a charm.](#)
- Show the sound effects of the game (rifle, game over, game win, level complete, duck is hit and falling etc).
[This functionality works like a charm.](#)
- Show that ducks are reflecting while they hit to the edges (or corners), you must show each reflection for each duck.
[This functionality works like a charm.](#)
- Show that ducks are getting in between background and foreground objects.
[This functionality works like a charm.](#)
- Show that one rifle can hit more than one duck.
[This functionality does not work at all.](#)
- Show that continuing to the next level (or starting again, exiting etc.) stops the sound effect at the end of the level immediately.
[This functionality works like a charm.](#)
- Show the texts during game play (flashing and steady ones).
[This functionality works like a charm.](#)
- Show necessary keyboard button interactions if there is something not mentioned.
[This functionality works like a charm.](#)
- Show the flying animation of all the ducks in all orientation.
[This functionality works like a charm.](#)
- Show the falling animations of all the ducks. ((1) finishes here)
[This functionality works like a charm.](#)

CHECKLIST OF PROJECT

- Show the scaling works by scaling up and down. Play the game for a while to show that every part of the game is scaled.
[This functionality works like a charm.](#)
- Show that adjusting volume changes volume of the game (I know that it is nearly impossible, but it is enough to narrate and record it, so, it is just enough to show it as I will test it again at my locale), you can do this at the same time while you are showing the scale factor for the sake of saving the time.
[This functionality works like a charm.](#)
- Close the game and show that contents are still the same by ls -lR (Unix (MacOS, Linux etc.) or dir -S (NT (Windows))
[This functionality works like a charm.](#)
- Moreover, please narrate the critical parts of your code (design), please do not narrate all of it, just narrate it for 1-2 minutes as it is just for understanding the code by your own perspective.
[This functionality works like a charm.](#)

CHECKLIST OF PROJECT

INCOMPLETE PARTS