

# CHECKLIST OF BBM104 SPRING 2023 PROJECT ASSIGNMENT 4

Please try the following test cases and drop your notes under each bullet point in **BLUE** color. You can see some examples below:

- Check that functionality x works.  
**This functionality works like a charm.**
- Check that functionality y works.  
**This functionality has some flaws, the system crashes when it is tried to do operation namely t.**
- Check that functionality z works.  
**This functionality does not work at all.**

You can add extra test cases, please insert them at where you tried at the video, but please insert them as in **RED** color, your notes are still in **BLUE**. This paper is also for noting which parts of your system does not work well, so you can create hypothetical test cases that makes your system does not response as it has to do and explain the problem.

My suggestion to you is firstly completing this document and then starting to record demo video, with this kind of approach, you can benefit from your notes that you written to this document, and your demo goes in parallel with your demo which is more consistent, but you are free to fill the checklist and record the demo in desired order, but note that your checklist and demo has to be parallel to each other.

## Test Cases

- Contents of the assets folder must be same as the one provided at Piazza.  
**The directories under the assets folder must have the same name as the assets folder provided at Piazza. The sprites can be changed, for example the game can run with different backgrounds, crosshairs, foregrounds etc. (not adding a foreground while adding a background may cause index out of bounds exception) Although sprites can be changed, changing the duck sprites might not fit the game. Additionally, the game only recognizes the first image it encounters on the directories assets/favicon and assets/welcome. Lastly, putting no sprites in a directory crashes the program.**
- 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)
  - javac -version (must be resulted same as "javac 1.8.0\_xxx")
  - java -version (must be resulted same as "java version "1.8.0\_xxx"")
- Compile your code and then run it.  
**The code can be compiled and run.**

First and Last Name : Barış Yıldız

Student ID : 2210356107

Link of the Demo Video : <https://youtu.be/8eA8bbVUXAk>

## CHECKLIST OF BBM104 SPRING 2023 PROJECT ASSIGNMENT 4

- Show that the game opens without any crash. Your system must also read the contents of the assets folder by itself.  
[The functionality works like a charm.](#)
- Narrate and show the scrolling functionality during your narration, you can mention it during the gameplay, you do not have to spend any extra spare time for it. If you did not implement this functionality, please state below; if you have done it, please drop your note as mentioned at beginning of the checklist.  
[The scrolling functionality is not implemented.](#)
- Show the title and icon of your game.
- Show that music is playing in loop at title screen.  
[The functionality works like a charm.](#)
- Show that user can exit with ESC key and open the game again.  
[The functionality works like a charm.](#)
- 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.  
[The 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.  
[The functionality works like a charm.](#)
- Show that music is still playing in loop where it has been left during background selection screen.  
[The functionality works like a charm.](#)
- Show that arrow keys can navigate between backgrounds and crosshairs.  
[The functionality works like a charm. I have also shown which key I was pressing to change the background and crosshairs.](#)
- Select one of the crosshair and background and then proceed with ENTER key.  
[The functionality works like a charm.](#)
- Show that your game will not start until intro effect finishes.  
[The 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.  
[The functionality works like a charm.](#)

First and Last Name : Barış Yıldız

Student ID : 2210356107

Link of the Demo Video : <https://youtu.be/8eA8bbVUXAk>

## CHECKLIST OF BBM104 SPRING 2023 PROJECT ASSIGNMENT 4

- Narrate and play each level until to the finish and start again from first level by pressing the enter key.  
[The game does start from the first level, keeping the desired background and crosshair. I've shown the different type of ducks and the 6 different directions. I've shown that the ducks cannot be hit during the endgame screens.](#)
- Show that playing again from the end of the game does not plays the sound effect again.  
[The functionality works like a charm.](#)
- Play each level again and press the ESC key to return to the title screen.  
[The 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).  
[The functionality works like a charm. The game can be continued with different selections without being relaunched.](#)
- Play the game and lose one of the levels, then press ENTER to play again.  
[The functionality works like a charm.](#)
- Play the game and lose one of the levels, then press ESC to return to the title screen.  
[The functionality works like a charm.](#)
- Show that your game cannot shoot after each situation (end of level by success or fail, end of the final level), you do not have to show it separately, it is enough to show it during the game play at the steps above, it is also enough to show it during the game play for the following steps. (Until (1) finishes)  
[The functionality works like a charm. I've shown it in the narration of gameplay and in the losing the game test case.](#)
- Show the sound effects of the game (rifle, game over, game win, level complete, duck is hit and falling etc).  
[The functionality works like a charm. I've shown it in the narration of gameplay.](#)
- Show that ducks are reflecting while they hit to the edges (or corners), you must show each reflection for each duck.  
[The functionality works like a charm. I've shown it in the narration of gameplay.](#)
- Show that ducks are getting in between background and foreground objects.  
[The functionality works like a charm. I've shown it in the narration of gameplay.](#)
- Show that one rifle can hit more than one duck.

First and Last Name : Barış Yıldız

Student ID : 2210356107

Link of the Demo Video : <https://youtu.be/8eA8bbVUXAk>

## CHECKLIST OF BBM104 SPRING 2023 PROJECT ASSIGNMENT 4

This functionality doesn't work, only one duck can be hit with one shot. The reason is that I placed the mouse click event handlers on duck objects instead of the scene itself which might have made it easier to implement.

- Show that continuing to the next level (or starting again, exiting etc.) stops the sound effect at the end of the level immediately.

The functionality works like a charm.

- Show the texts during game play (flashing and steady ones).

The functionality works like a charm.

- Show necessary keyboard button interactions if there is something not mentioned.

- Show the flying animation of all the ducks in all orientation.

The functionality works like a charm. I've shown it in the narration of gameplay.

- Show the falling animations of all the ducks. ((1) finishes here)

The functionality works like a charm. I've shown it in the narration of gameplay. I've also shown the directions where ducks look when they are falling.

- 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.

The 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.

The 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)))

The 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.

The code contains a driver class DuckHunt which also has the configuration instance variables SCALE and VOLUME. The classes Sprites and Effects store the images and sound effects respectively. The other classes are scene classes that create the scenes in the game. These are TitleScreen, BackgroundSelectionScreen and Level. These classes each have "initializer methods" that initialize that scene's elements such as text, background etc. The Duck class creates the ducks and moves them. The helper classes DuckColor and FlyingDirection help initializing ducks.

First and Last Name : Barış Yıldız

Student ID : 2210356107

Link of the Demo Video : <https://youtu.be/8eA8bbVUXAk>

# CHECKLIST OF BBM104 SPRING 2023 PROJECT ASSIGNMENT 4

## INCOMPLETE PARTS

- Two or more ducks cannot be killed at the same time. A shot can only kill one duck. The reason for this is that the check for “clicks on ducks” are controlled by duck objects and ducks are not at the same level, they are all between the foreground and the background, but not at the same layer.
- The crosshair doesn't get resized when the game gets resized.
- Sometimes, the ducks get a little bit out of screen (only their beak) before they reflect. (in the collision with the game screen edges)

First and Last Name : Barış Yıldız

Student ID : 2210356107

Link of the Demo Video : <https://youtu.be/8eA8bbVUXAk>

# CHECKLIST OF BBM104 SPRING 2023 PROJECT ASSIGNMENT 4

## RESTRICTIONS

- Your demo **must contain sound**, you can use Zoom's video recorder for this purpose, be sure about that you are sharing your **whole screen, not just the application**.
- Your demo video **cannot be longer than five minutes, four minutes for the GUI and one minute for code**, (3-4 minutes for the game and 1-2 minutes for the code is also OK but the total length of your video cannot be longer than five minutes) it cannot exceed five minutes even a little second, and your video must be in real speed, please do not speed up or down your videos.
- Everything about your **demo video must be in English** (you do not have to set your system language to English for sure). Your video must be fluent. It is not assumed that you are a native speaker, but it is known that your English level is at least B1+ (according to prep class), so, it is requested from you to speak at least as a B1+ speaker, little mistakes about your speech are going to be discarded but avoid mistakes as much as you can.
- **Do not do these operations too fast or too slow**, make them at the speed that a normal user (**who is not very familiar with your system and tries to read texts**) does.
- **Show the things that make your system crash or things that you couldn't done in demo properly** but show them after the requirements of the demo finishes, you can skip some of the requirements by saying that you did not implement it, so that your demo video becomes more fluid, less erroneous, but note that you must show your erroneous parts at the end.
- If your game fails one of the steps, please open the game where it must be according to failed step's result and continue from there.
- **Preferred screen resolution and scale for Windows 10 or later is 1920x1080 with scale of 1.25**, you can select the same for the Linux systems too, for the MacOS, you can use the preferred specifications of your system, but it would be good if you can use your MacBook's own screen instead of any external screens.
- You must both narrate your GUI and code.
- **It is your own responsibility to show all the functionality of your system**, so if you have not mentioned about a functionality that is a request of this project or part of your extra feature, your code may be graded as low as zero for that part as there is no working example. So, please do not feel limited with given conditions, show your all work to get full credit, that means, if you have not shown a concept at your demo video, you will not have any right to say that **"But it worked on my machine!"**, because this demo video is a chance for you to show that your code works on your machine well, but note that you can still have right to objection for the parts that are not mentioned at PDF, Piazza Notes, Q&As; say that there is a bug at a functionality and it is neither mentioned at PDF, Piazza, Q&A, nor mentioned during your implementation and you did not notice that it does not work on other machines, then for sure you can object for it by clipping a video that shows it works on your machine.
- **It is your own responsibility to follow all things about this homework, PDF, Piazza, Q&A etc. So, you assumed that aware of everything mentioned at there.**

First and Last Name : Barış Yıldız

Student ID : 2210356107

Link of the Demo Video : <https://youtu.be/8eA8bbVUXAk>