



ACADEMIC YEAR: 2020-2021

ASSIGNMENT 2 (2D Game)

Assignment Posted: October 02, 2020**Assignment Due: October 22, 2020 before 11.59pm****Final Deadline with 20% flat Penalty: October 25, 2020 before 11.59pm**

Overview:

Duck Hunt (a light gun shooter video game developed by Nintendo for its NES video game console) was a popular video game during the 1980s. The game had a light/beam gun as an accessory to aim and shoot ducks as well as moving objects appearing on screen. This was a first-person perspective shooter game with an objective to shoot the moving targets midair before they hit the ground or disappear. At any given point in time, there used to be a few targets available on screen. Player used to get a few attempts to aim and shoot the targets down. On successfully shooting down the minimum required targets, a player used to advance to the next level. In case of not shooting the required number of targets, a game over screen used to show up. More importantly, the nameless canine who used to laugh at the payer every time he/she missed a target has haunted the players for ages. The game had ninety-nine levels and the game used to reset after the last level.

Your first Unity programming assignment is to develop a 2D variation of Duck Hunt. In this variation rather than being in beautiful sunlight, the game will take place in a scary starry night. With Halloween just around the corner, you will be creating this game with a lot of spooky elements. Flying objects will be replaced by pumpkins or ghosts moving in a random fashion. The player will get a fixed amount of time to shoot down these targets, after which they will disappear. In case the player misses a shot, the famous dog should make an appearance. Each level will have a fixed number of targets to be shot in a fixed amount of time. Moreover, a witch on a broom should appear occasionally to offer bonus earning opportunity to the player. The witch will require three shots to get killed and will be available only for a short period. The goal of the player is to get a high score and, also to complete as many levels as possible.



Basic Game Play:

Points are awarded for successfully shooting the targets and the value is based on number of successful shots and attempts taken to shoot them:

- Every successfully shot target will yield 3 points and every missed target will penalize the player with 1 point. All targets will spawn in random place, will follow random trajectories and will be available for a short period of time on screen.
- To add to the challenge, the moving targets should increase in speed every time there is a change of level.
- The player will be able to shoot two targets at a time, if they overlap and will get a bonus of 5 points for a successful shot.
- The witch will appear at least twice in each level at random times.
- The player will be able to take only one shot at a time, continuous shooting is not allowed.

Game Versions:

You have to develop two versions of the game as follows:

- **Version-Normal** – In this version, the basic version of the game is implemented as above. The unending game ends when the player fails to shoot a specified number of targets.
- **Variant-Special** - In this version you can get a continuous shooting gun to shoot down as many targets as you can. In this special mode, the number of targets will also increase. You will use a special key to toggle this mode for a limited time (say, 3 seconds). Missed targets will cost 2 points and successful shots will yield 1 point each.

User interface and gameplay parameters:

- Design your own user interface using keys for the aim and shoot mechanism.
- The current score should be displayed on the screen.
- You will have to suitably adjust your gameplay parameters, such as number and types of targets, how frequently are they spawned, where do they spawn, are of what kind, speed and speed change factors, *etc.*, so as to yield a playable game.

Submission:

Assignment must be submitted only through Moodle. No other form of submission will be considered. Please create a zip file containing your source code, all the assets, data files, a README (.txt) file explaining how to read the code, compile, run, and play the game on a PC with Unity. You must also record and submit a short video of the game play in the same zip file. The zip file should be named Assignment#_YourStudentID.

Evaluation Procedure:

You MUST demonstrate your program to the lab instructor during lab hours, right after the due date. You must run your submitted code, demonstrate its full functionality and answer questions about the Unity programming aspects of your game. Major marking is done on the spot during the demo. Your code will be further checked for structure, non-plagiarism, *etc.* However, ONLY demonstrated submissions will receive marks.

Evaluation Scheme:

1. Working implementation of Normal version with all game play: (40%)
2. Working implementation of Variant Special with all game play: (15%)
3. Setting and playability (User Interface, Game world, ...): (10%)
4. Aesthetics and overall impression: (20%)
5. Q &A (to demonstrate understanding of Unity programming): (15%)

Ground Rules:

You are welcome to discuss high-level implementation issues with your classmates and others, but you should avoid actually looking at other students' code as whole, and under no circumstances should you be copying any portion of another student's code or copying complete code from the Internet. However, seeking help from other students for debugging some portion of your code is reasonable. Basically, these "ground rules" are intended to prevent a student from "freeloading" off another student or from the Internet, even accidentally.