How the ”average Joe” can upgrade and modernize old classic games, and the electronics required

Johannes Holstad
Institute of Informatics,
University of Oslo,
Gaustadallen 23B 0373 Oslo,
Norway

***Abstract*—Since children and young adults today play with iPad’s and electronics from a young age, classic games like Tetris®, chess, battleship and so on, lose their temptation for generation Z. The goal of this experiment was to see if we could relatively fast and easy improve an old classic game, so that kids and young adults today again could enjoy the games we enjoyed so much. We based our experiment on battleship and improved this game over only three weeks. The same techniques and principles we used here can be applied to other old classic games like chess or ludo as well.**

***Index Terms*—DIY, game improvement, arcade games, Arduino**

I. Introduction

The ability to build your own constructions have long been available to only the rich and well fortunate. Expensive machinery like 3D printers and laser cutters are being made available at more places then just huge companies and factories. Schools, universities and publicly open makerspace areas are making them available for whoever finds it interesting. We propose that it is fairly simple to build a new and improved version of a classic like battleship, and with this show that most people can do the same. This will be done by only using the materials and machines we found in our university. The whole construction is build from scratch, the only pre-built component we use is an Arduino used to program the game console.

II. The problem

Old board games that used to gather the young and made then socialize are dying out because of their simplicity and lack of electronics. Kids these days are more interested in online gaming and often spend several hours a day in their room. We would like to show that everybody can take an old retro game that connects people and improve and upgrade it into something new and exciting.

III. The idea

The idea was to improve the old classic game battleship. We wanted a modern version that included lights, more interaction with the player then the old version and add new features. And we wanted to do it quick and with the materials that was already readily available in our university. By doing this we prove that upgrading and making personalised retro games is available to a lot more people in the modern day with the use of new machinery and technology then it was before.

IV. Background

Battleship is a two player game where each player have their own board separated by a wall between them, the players each get a number of boats of pieces that they place on their board, see figure 1. The board is a grid like a chessboard of varying size, but the standard is a ten by ten board. the pieces can also vary in size and length but the standard configuration [1] is shown in table I. This is the most played Hasbro configuration, but there are countless variations and you can play with whatever setup you want. The goal of the game is to guess where your opponent have places his boats and sink all of them. The players take turns and say out loud a square on the opponents board where he think a boat is placed. Once a player have gotten all of his boats sunk he loses the game.

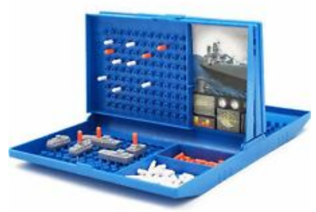

Fig. 1: The classic battleships

Numb. per player	class of ship	Size
1	Carrier	5
2	Battleship	4
3	Submarine	3
4	Destroyer	3
5	patrol boat	2

TABLE I: Hasbro game setup

V. Method

A. Construction

The console was manly built using 3D CAD/CAM designing software and a laser cutter. The build process can be found in detail in [2].

B. Electronics

The main goal of the is the make the game desirable and eye-catching. In order to achieve this we saw no other way to do this then to make it electronic with an upgradeable system.

Power source: The game is powered by a 7.5V power supply connected to the microcontroller. Which pulls it down to 5V and supply all LEDs and switches.

LED: We decided that we wanted to use RGB LED lights to display ships, water, bombs (hit and miss) and over all create an arcade experience. The game would then require a lot of them and in order to fit them all to one microcontroller we found the simplest way would be to create LED matrices out of 5V addressable LED strips as shown in figure 2. The strips only have three wires (Vin, Din/Dout and GND). This means that we can control single LEDs in the matrix using one pin connected to the microcontroller. Resulting in a simpler layout and making it easier to program the console. We also included a chain of LEDs around the wall to upgrade the experience. All LED sections (wall front/back, board (front/back), around wall) are connected to 5V, GND and their own pin on the microcontroller through a 390Ω resistor.

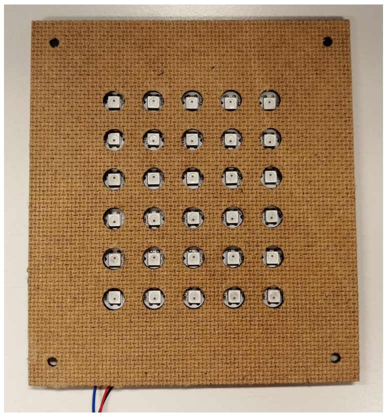
Fig. 2: LED matrix for wall

Control Box: The game is played using self-made control boxes, see figure 3. Each of them have 9 wires connected to the joystick, buttons, flip switches and one shared GND. The boxes are plugged in to the wall of the game. From there they are all connected to GND, a pin on the microcontroller and to 5V through a 1.2 kΩ resistor to reduce noise. When a switch/button is pushed it will then pull the pin to GND and register an "LOW" in the code.

Unmanned Aerial Vehicle (UAV): The fun thing about DIY (do it yourself) projects is that you can put your own mark on it and include whatever you want to. One thing we did was to include a UAV or drone that the player can use once. When used the "drone" searches for enemy ships and reports back to the player. On the players wall one column or row will flicker between red and white for 2 seconds. The player will then know that somewhere on that row or column there is some part of a ship. They will not know the orientation of the ship.

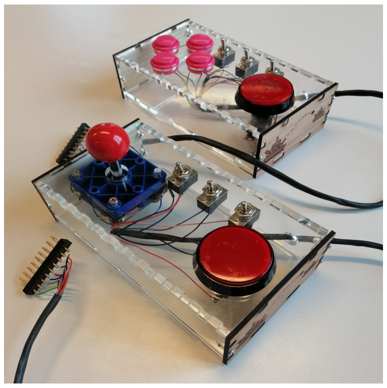
Fig. 3: Control boxes used by players

RESET: We also included a way to reset the game without unplugging it, by connecting an output pin and the reset pin on the Arduino. It is now programmed to reset if both players press the UAV button at the same time.

Components
WS2812B LED Strips (Adressable)
WS2811 LED Chain (Adressable)
Zippyy Arcade Joystick
Push Buttons
Flip Switches

Everything is connected to the microcontroller board Arduino Due[3] as shown in figure 4. The final result can be seen in figure 5.

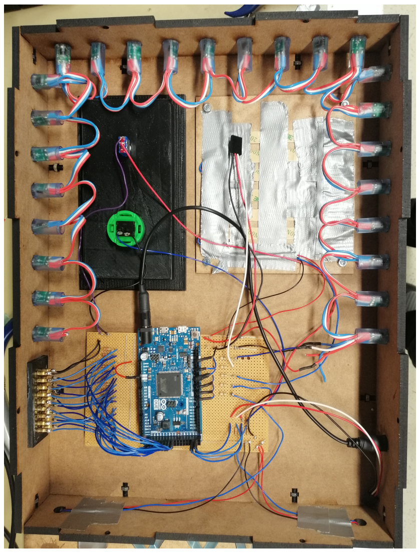
Fig. 4: Inside the wall

C. Code

All the code was programmed in the Arduino IDE. The code for controlling the LED strips is based on the FastLED library [4] [5]. It lets you individually control LEDs in strips such as WS2812B [6], WS2811 and more. The game code is manly based on matrices, such as the board layout for both player (if there's a ship or not), that wall layout (hit/miss) and LED indices. The most amusing part of coding the game yourself is that your imagination is the limitation. You can include whatever you want to, and if you would like to upgrade the game at a later time that is no problem. The code will be made available at GitHub [7]. The gist of the code is to set up all pins (input/output), and then run a start sequence where the players can place ship on the board. After that the game begins one player at a time. Upgrades we have included is the UAV, as explained in the electronics section, safety switches meaning that if the three switches are not in the correct direction (all down) the game will give a warning the first time you fire (red flashing lights all over the board). The next time you fire you fire at your own board. The players turn does not change when the switches are in the wrong direction. We also included a winning sequence where the winners side and the wall lights randomly change color, while the losers side is constantly red. It is fairly easy to program an Arduino and once you are familiar with the setup anyone can do it.

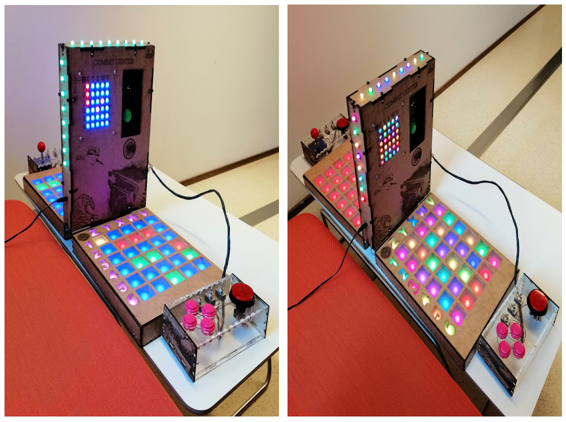
(a) During the game (b) When someone wins

Fig. 5: Game play

VI. Related Work

In the latest years we have seen a rise in DIY culture. People are interested in building their own games and object. Therefore anyone who wants to start a build, but has no to little experience have a vast amount of information they can use, even if people haven't built the exact same thing. In our case we found no one who had upgraded battleship to the extent we have, but we did for instance find guides on how to control LED strips[8].

VII. Conclusions and Future Work

The completion and success of this experiment shows that with the modern availability to design tools like fusion and easy access to laser cutters and 3d printers, the making and upgrading of old classic are possible for the average person. With just three weeks and the technology and tools readily available at our public university we managed to take battleship to a new level and bring in into the modern day and age.

Acknowledgment

This work could not be done without the support and collaboration of Pål Primstad and the guidance of Mats Erling Hvin and Vegard Syseth from the ROBIN group at the University of Oslo as part of the project in in5590.

References

[1] M. B. Company, "Battleship instructions," 2016. [Online]. Available: https://www.hasbro.com/common/instruct/Battleship.PDF
[2] P. Primstad, "Can old classic games be upgraded and made modern by the everyday man, or are they old classics for a reason?" 2019. [Online]. Available: http://www.robotikk.com/in5590/devil/g9/p8/Battleship_article_paal.pdf
[3] Arduino, "Arduino due," 2019. [Online]. Available: https://store.arduino.cc/due
[4] D. Garcia and M. Kriegsman, "Fastled animation library," 2019. [Online]. Available: http://fastled.io/
[5] ——, "Fastled animation library," 2019. [Online]. Available: https://github.com/FastLED/FastLED
[6] Adafruit, "Ws2812 intelligent control led integrated light source." [Online]. Available: https://cdn-shop.adafruit.com/datasheets/WS2812.pdf
[7] J. Holstad and P. Primstad, "Battleship project," 2019. [Online]. Available: https://github.uio.no/johahols?tab=repositories
[8] D. Nedelkovski, "How to control ws2812b individually addressable leds using arduino," 2018. [Online]. Available: https://howtomechatronics.com/tutorials/arduino/how-to-control-ws2812b-individually-addressable-leds-using-arduino/
[9] Arduino, "Arduino ide," 2019. [Online]. Available: https://www.arduino.cc/en/Main/Software
[10] Adafruit, "Ws2811, signal line 256 gray level 3 channal, constant current led drive ic." [Online]. Available: https://cdn-shop.adafruit.com/datasheets/WS2811.pdf