

Final Integration Project: Bugs War

Bugs War is a timed interactive mouse-click game that possesses an Arduino board functionality. This project integrates various sound, graphics, and hardware skills accumulated in the course throughout the semester. Javascript (P5.js and Tone.js), HTML, and Arduino Hardware programming provide the foundation of this project. Here graphical elements using P5 and audio elements using Tone.js are implemented and integrated with Arduino hardware.

The game's goal is for users to eliminate (squish) a set number of green bugs using mouse clicks before the allocated time ends. Each bug squish gradually raises the speed of other non-squished bugs thereby raising the difficulty as you play. Missed squishes lead to the penalty of enemy reinforcements and users need to take care to avoid them. Other bugs exist but require certain conditions to be spawned. When squished, the blue bug slows other bugs, the red bug restores three seconds to your allocated time, and the purple spawns a remote controllable drone that squishes bugs on contact.

User activity triggers in-game events and sounds. In-game events are conveyed to the Arduino and trigger responses on the board. One instance is the game's perks functionality which under certain conditions determined by user activity are activated. The screen when playing possesses three black-filled ellipses at the top that represent meters for these perks. The ellipses change color when their associated perk is active. For example, getting a number of kills that is a multiple of five triggers an in-game event that fills the leftmost perk meter with the color blue. Getting a number of kills that is a multiple of eight fills the center perk meter with the color red. The rightmost perk meter turns purple randomly rather than through user activity. These are all conveyed to the Arduino board.

The Arduino board circuit combines two buttons and three standard LEDs. The blue LED corresponds to the slow perk acquired from getting a multiple of 5 kills, the red LED is for the reverse perk gotten from squishing a multiple of 8 bugs, and the white LED represents time warnings. The plain white LED blinks repeatedly when the allocated game time is almost up. One button triggers the perk action of the nearest colored LED, so a lit-up blue LED means the associated nearest button spawns a slow bug (blue bug) on the game screen. Both LEDs blinking at intervals means a drone perk is active and corresponds with the purple ellipse perk meter. Both buttons must be clicked to activate this perk action.

Each perk action spawns the bug with the same color as their perk meter on the game screen. Squishing these bugs alters the game similar to power-ups in other games. The ellipse (drone) spawned from squishing the purple bug can be controlled using the two buttons. This 'drone' moves in only one direction on both the x-axis and y-axis, however hitting the screen's boundaries can reverse the directions. One button moves the ellipse along the x-axis while the other does the y-axis.

A demo showcasing most of these features is located on the Github pages index. It depicts two playthroughs that involve the use of the drone bug and the slow bug as well as overall gameplay. Fig 1 is a sketch of the Arduino board, here the Yellow LED represents the plain white LED.

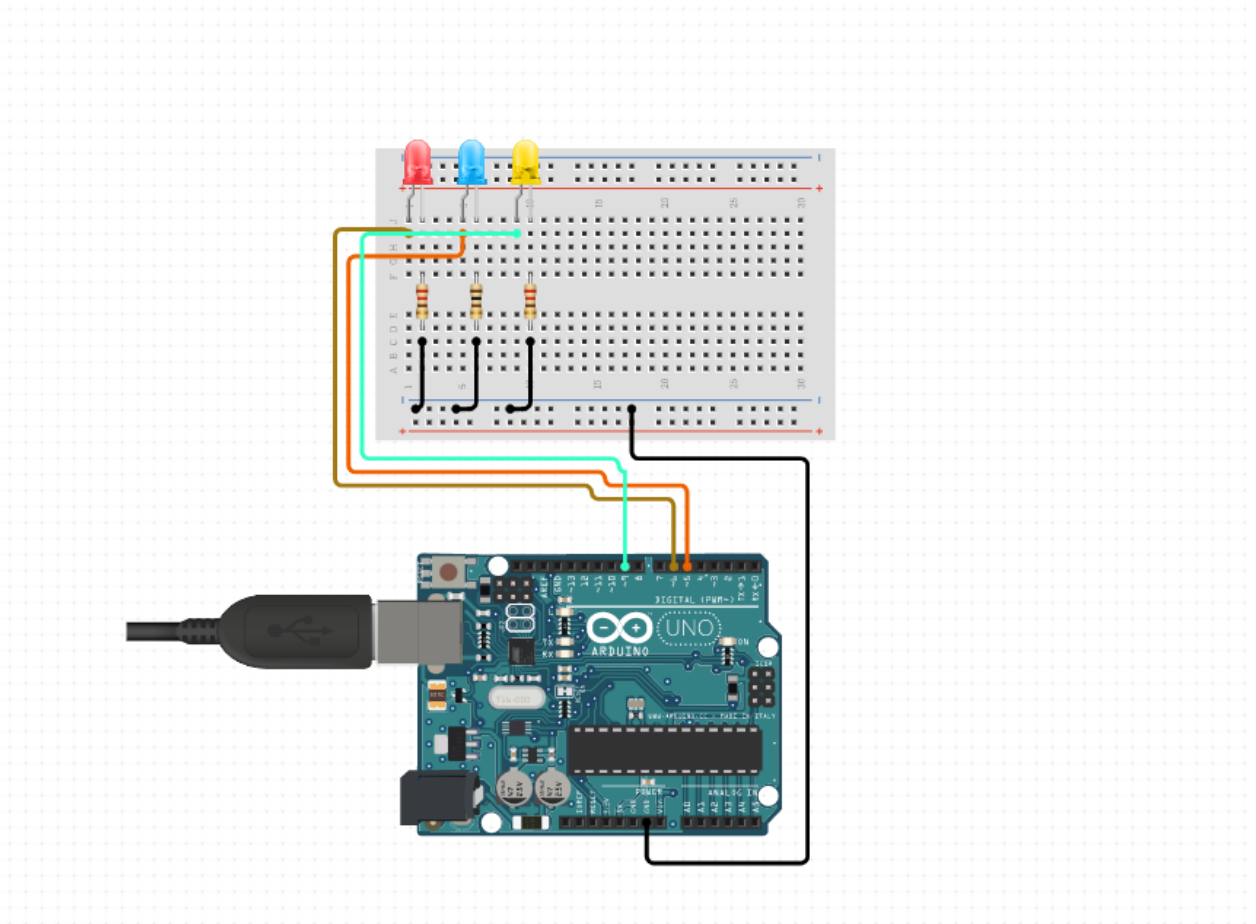


Fig 1

