Spider Combo system.

Two players, presented with button combinations, timed to input buttons, wrong key results in failure, faster combinations result in more points.

As player gets ahead they get more complicated combinations and the female faces the other more.

Once the point disparity is wide enough the female eats the loser and loves the winner.

Button combinations, List of keys, generated based on difficulty value.

List of enums as available keys,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Very Low*** | ***Low*** | ***Medium*** | ***High*** | ***Very High*** |
| Left | Left | Left | Left | Left |
| Right | Right | Right | Right | Right |
| Up | Up | Up | Up | Up |
| Down | Down | Down | Down | Down |
|  | A | A | A | A |
|  | B | B | B | B |
|  |  | X | X | X |
|  |  | Y | Y | Y |
|  |  |  | RB | RB |
|  |  |  | LB | LB |
|  |  |  |  | RT |
|  |  |  |  | LT |

4 4 4 5 5

**Generate Combo()**

New List, sequence of Length,

Included buttons range by difficulty

For each list element add random button within range

DisplayCombo()

**Display Combo()**

Draw sequence of UI elements from combo

Set current step to 0 and timer to 0

Start dancing (hide behind a bool in the update?)

**Dance()**

Start timer

Compare current step to next key input, if they don’t match, FAIL, else next step.

If there’s no next step, combo completed, tally a score, run the animation, CheckWinLose.

**CheckWinLose()**

Compare player scores, if difference > x, run WinLose, else rotate female to represent difference and GenerateCombo()

**WinLose(loser)**

Female tackles loser, play sounds, female approaches male, play particles, show UI elements