Homework Turnin

Name:	Jack T Venberg
Account:	jvenberg (jvenberg@uw.edu)
Student ID:	1764621
Section:	AH
Course:	CSE 154 18sp
Assignment:	cp8
Receipt ID:	72eaa5241ef72acb9b684bfea932c3db

Uploading to Webster

Uploaded to
https://webster.cs.washington.edu/remote_render/data/turnin/72eaa5241ef72acb9b684bfea932c3db/

- cp8.zip

Turnin Successful!

The following file(s) were received:

UNO_deck.svg (170457 bytes)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0" y="0"
  <g id="Layer_1">
    <g>
      <g>
        <path d="M284,1 L444,1 C466.091,1 484,18.909 484,41 L484,321 C484,343.091 466.091,361 444,361 L284,36
        <path d="M284,1 L444,1 C466.091,1 484,18.909 484,41 L484,321 C484,343.091 466.091,361 444,361 L284,36
      </g>
      <path d="M284,21 L444,21 C455.046,21 464,29.954 464,41 L464,321 C464,332.046 455.046,341 444,341 L284,3
      <path d="M424,81 C335.636,81 264,152.636 264,241 C264,263.092 281.908,281 304,281 C392.364,281 464,209.
      <path d="M356,121 L336,141 L336,165 L356,145 L356,241 L376,241 L376,121 z" fill="#FF5555"/>
      <path d="M284,31 L274,41 L274,53 L284,43 L284,91 L294,91 L294,31 z M444,331 L454,321 L454,309 L444,319
    </g>
    <g>
      <g>
        <path d="M41,1 L201,1 C223.091,1 241,18.909 241,41 L241,321 C241,343.091 223.091,361 201,361 L41,361
        <path d="M41,1 L201,1 C223.091,1 241,18.909 241,41 L241,321 C241,343.091 223.091,361 201,361 L41,361
      </g>
      <path d="M41,21 L201,21 C212.046,21 221,29.954 221,41 L221,321 C221,332.046 212.046,341 201,341 L41,341
      <path d="M181,81 C92.636,81 21,152.636 21,241 C21,263.092 38.908,281 61,281 C149.364,281 221,209.368 22
      <path d="M121,121 C98.908,121 81,138.908 81,161 L81,201 C81,223.092 98.908,241 121,241 C143.092,241 161
      <path d="M51,31 C39.954,31 31,39.954 31,51 L31,71 C31,82.046 39.954,91 51,91 C62.046,91 71,82.046 71,71
    </g>
    <g>
      <g>
        <path d="M1742,1 L1902,1 C1924.091,1 1942,18.909 1942,41 L1942,321 C1942,343.091 1924.091,361 1902,36
        <path d="M1742,1 L1902,1 C1924.091,1 1942,18.909 1942,41 L1942,321 C1942,343.091 1924.091,361 1902,36
      </g>
      <path d="M1742,21 L1902,21 C1913.046,21 1922,29.954 1922,41 L1922,321 C1922,332.046 1913.046,341 1902,3
      <path d="M1882,81 C1793.636,81 1722,152.636 1722,241 C1722,263.092 1739.908,281 1762,281 C1850.364,281
      <path d="M1788,121 L1788,161 L1808,161 L1808,141 L1848,141 L1808,241 L1828,241 L1868,141 L1868,121 L178
      <path d="M1732,31 L1732,51 L1742,51 L1742,41 L1762,41 L1742,91 L1752,91 L1772,41 L1772,31 L1732,31 z M1
    </g>
    <g>
      <g>
        <path d="M527,1 L687,1 C709.091,1 727,18.909 727,41 L727,321 C727,343.091 709.091,361 687,361 L527,36
        <path d="M527,1 L687,1 C709.091,1 727,18.909 727,41 L727,321 C727,343.091 709.091,361 687,361 L527,36
```

```
    </g>
    <path d="M527,21 L687,21 C698.046,21 707,29.954 707,41 L707,321 C707,332.046 698.046,341 687,341 L527,3
    <path d="M667,81 C578.636,81 507,152.636 507,241 C507,263.092 524.908,281 547,281 C635.364,281 707,209.
    <path d="M603,121 C580.84,121 563,138.84 563,161 L563,165 L583,165 L583,161 C583,149.92 591.92,141 603,
    <path d="M537,31 C525.92,31 517,39.92 517,51 L517,53 L527,53 L527,51 C527,45.46 531.46,41 537,41 C542.5
  </g>
  <g>
    <g>
      <path d="M770,1 L930,1 C952.091,1 970,18.909 970,41 L970,321 C970,343.091 952.091,361 930,361 L770,36
      <path d="M770,1 L930,1 C952.091,1 970,18.909 970,41 L970,321 C970,343.091 952.091,361 930,361 L770,36
    </g>
    <path d="M770,21 L930,21 C941.046,21 950,29.954 950,41 L950,321 C950,332.046 941.046,341 930,341 L770,3
    <path d="M910,81 C821.636,81 750,152.636 750,241 C750,263.092 767.908,281 790,281 C878.364,281 950,209.
    <path d="M850,121 C833.431,121 820,134.431 820,151 L840,151 C840,145.477 844.477,141 850,141 C855.523,1
    <path d="M780,31 C771.716,31 765,37.716 765,46 L775,46 C775,43.239 777.239,41 780,41 C782.761,41 785,43
  </g>
  <g>
    <g>
      <path d="M1013,1 L1173,1 C1195.091,1 1213,18.909 1213,41 L1213,321 C1213,343.091 1195.091,361 1173,36
      <path d="M1013,1 L1173,1 C1195.091,1 1213,18.909 1213,41 L1213,321 C1213,343.091 1195.091,361 1173,36
    </g>
    <path d="M1013,21 L1173,21 C1184.046,21 1193,29.954 1193,41 L1193,321 C1193,332.046 1184.046,341 1173,3
    <path d="M1153,81 C1064.636,81 993,152.636 993,241 C993,263.092 1010.908,281 1033,281 C1121.364,281 119
    <path d="M1088.2,121 L1049,201 L1049,221 L1097,221 L1097,241 L1117,241 L1117,221 L1129,221 L1129,201 L1
    <path d="M1022.2,31 L1002.6,71 L1002.6,81 L1026.6,81 L1026.6,91 L1036.6,91 L1036.6,81 L1042.6,81 L1042.
  </g>
  <g>
    <g>
      <path d="M1256,1 L1416,1 C1438.091,1 1456,18.909 1456,41 L1456,321 C1456,343.091 1438.091,361 1416,36
      <path d="M1256,1 L1416,1 C1438.091,1 1456,18.909 1456,41 L1456,321 C1456,343.091 1438.091,361 1416,36
    </g>
    <path d="M1256,21 L1416,21 C1427.046,21 1436,29.954 1436,41 L1436,321 C1436,332.046 1427.046,341 1416,3
    <path d="M1396,81 C1307.636,81 1236,152.636 1236,241 C1236,263.092 1253.908,281 1276,281 C1364.364,281
    <path d="M1300,121 L1300,181 L1340,181 C1351.08,181 1360,189.92 1360,201 C1360,212.08 1351.08,221 1340,
    <path d="M1246,31 L1246,61 L1266,61 C1271.54,61 1276,65.46 1276,71 C1276,76.54 1271.54,81 1266,81 C1263
  </g>
  <g>
    <g>
      <path d="M1499,1 L1659,1 C1681.091,1 1699,18.909 1699,41 L1699,321 C1699,343.091 1681.091,361 1659,36
      <path d="M1499,1 L1659,1 C1681.091,1 1699,18.909 1699,41 L1699,321 C1699,343.091 1681.091,361 1659,36
    </g>
    <path d="M1499,21 L1659,21 C1670.046,21 1679,29.954 1679,41 L1679,321 C1679,332.046 1670.046,341 1659,3
    <path d="M1639,81 C1550.636,81 1479,152.636 1479,241 C1479,263.092 1496.908,281 1519,281 C1607.364,281
    <path d="M1579,121 C1556.908,121 1539,138.908 1539,161 L1539,201 C1539,223.092 1556.908,241 1579,241 C1
    <path d="M1509,31 C1497.954,31 1489,39.954 1489,51 L1489,71 C1488.996,80.053 1495.074,87.979 1503.818,9
  </g>
  <g>
    <g>
      <path d="M1985,1 L2145,1 C2167.091,1 2185,18.909 2185,41 L2185,321 C2185,343.091 2167.091,361 2145,36
      <path d="M1985,1 L2145,1 C2167.091,1 2185,18.909 2185,41 L2185,321 C2185,343.091 2167.091,361 2145,36
    </g>
    <path d="M1985,21 L2145,21 C2156.046,21 2165,29.954 2165,41 L2165,321 C2165,332.046 2156.046,341 2145,3
    <path d="M2125,81 C2036.636,81 1965,152.636 1965,241 C1965,263.092 1982.908,281 2005,281 C2093.364,281
    <path d="M2065,121 C2048.431,121 2035,134.431 2035,151 C2035,157.796 2037.32,163.972 2041.124,169 C2031
    <path d="M1995,31 C1986.716,31 1980,37.716 1980,46 C1980,49.4 1981.16,52.484 1983.064,55 C1977.987,58.7
  </g>
  <g>
    <g>
      <path d="M2228,1 L2388,1 C2410.091,1 2428,18.909 2428,41 L2428,321 C2428,343.091 2410.091,361 2388,36
      <path d="M2228,1 L2388,1 C2410.091,1 2428,18.909 2428,41 L2428,321 C2428,343.091 2410.091,361 2388,36
    </g>
    <path d="M2228,21 L2388,21 C2399.046,21 2408,29.954 2408,41 L2408,321 C2408,332.046 2399.046,341 2388,3
    <path d="M2368,81 C2279.636,81 2208,152.636 2208,241 C2208,263.092 2225.908,281 2248,281 C2336.364,281
    <path d="M2308,121 C2285.908,121 2268,138.908 2268,161 C2268,183.092 2285.908,201 2308,201 C2315.268,20
    <path d="M2238,91 C2249.046,91 2258,82.046 2258,71 L2258,51 C2258.004,41.947 2251.926,34.02 2243.183,31
  </g>
  <g>
    <g>
      <path d="M2471,1 L2631,1 C2653.091,1 2671,18.909 2671,41 L2671,321 C2671,343.091 2653.091,361 2631,36
      <path d="M2471,1 L2631,1 C2653.091,1 2671,18.909 2671,41 L2671,321 C2671,343.091 2653.091,361 2631,36
    </g>
    <path d="M2471,21 L2631,21 C2642.046,21 2651,29.954 2651,41 L2651,321 C2651,332.046 2642.046,341 2631,3
    <path d="M2611,81 C2522.636,81 2451,152.636 2451,241 C2451,263.092 2468.908,281 2491,281 C2579.364,281
    <path d="M2551.124,121 C2535.194,120.951 2519.901,127.248 2508.624,138.5 C2485.164,161.9 2485.1,199.912
    <path d="M2491.064,31 C2483.1,30.974 2475.453,34.122 2469.816,39.748 C2458.079,51.447 2458.05,70.445 24
  </g>
  <g>
    <g>
      <path d="M2714,1 L2874,1 C2896.091,1 2914,18.909 2914,41 L2914,321 C2914,343.091 2896.091,361 2874,36
      <path d="M2714,1 L2874,1 C2896.091,1 2914,18.909 2914,41 L2914,321 C2914,343.091 2896.091,361 2874,36
    </g>
```

```
    <path d="M2714,21 L2874,21 C2885.046,21 2894,29.954 2894,41 L2894,321 C2894,332.046 2885.046,341 2874,3
    <path d="M2854,81 C2765.636,81 2694,152.636 2694,241 C2694,263.092 2711.908,281 2734,281 C2822.364,281
    <path d="M2794,116 L2804,126 L2764,166 C2754,176 2754,196 2764,206 L2784,186 L2824,146 L2834,156 L2834,
    <path d="M2794,246 L2784,236 L2824,196 C2834,186 2834,166 2824,156 L2804,176 L2764,216 L2754,206 L2754,
    <path d="M2724,31 L2729,36 L2709,56 C2704,61 2704,71 2709,76 L2719,66 L2739,46 L2744,51 L2744,31 z" fil
    <path d="M2724,96 L2719,91 L2739,71 C2744,66 2744,56 2739,51 L2729,61 L2709,81 L2704,76 L2704,96 z M286
    <path d="M2864,331 L2859,326 L2879,306 C2884,301 2884,291 2879,286 L2869,296 L2849,316 L2844,311 L2844,
  </g>
  <g>
    <g>
      <path d="M3443,1453 L3603,1453 C3625.091,1453 3643,1470.909 3643,1493 L3643,1773 C3643,1795.091 3625.
      <path d="M3443,1453 L3603,1453 C3625.091,1453 3643,1470.909 3643,1493 L3643,1773 C3643,1795.091 3625.
    </g>
    <path d="M3443,1473 L3603,1473 C3614.046,1473 3623,1481.954 3623,1493 L3623,1773 C3623,1784.046 3614.04
    <path d="M3583,1533 C3494.636,1533 3423,1604.636 3423,1693 C3423,1715.092 3440.908,1733 3463,1733 C3551
    <path d="M3482.6,1483 L3463,1523 L3463,1533 L3487,1533 L3487,1543 L3497,1543 L3497,1533 L3503,1533 L350
    <path d="M3437,1503 L3437,1513 L3427,1513 L3427,1523 L3437,1523 L3437,1533 L3447,1533 L3447,1523 L3457,
    <path d="M3563.4,1783 L3583,1743 L3583,1733 L3559,1733 L3559,1723 L3549,1723 L3549,1733 L3543,1733 L354
    <path d="M3609,1763 L3609,1753 L3619,1753 L3619,1743 L3609,1743 L3609,1733 L3599,1733 L3599,1743 L3589,
    <g>
      <g>
        <path d="M3485.86,1582.995 L3525.86,1582.995 C3531.383,1582.995 3534.705,1587.038 3533.28,1592.025
        <path d="M3485.86,1582.995 L3525.86,1582.995 C3531.383,1582.995 3534.705,1587.038 3533.28,1592.025
      </g>
      <path d="M3493.003,1592.995 L3513.003,1592.995 C3518.526,1592.995 3521.848,1597.038 3520.423,1602.025
    </g>
    <g>
      <g>
        <path d="M3525.86,1552.995 L3565.86,1552.995 C3571.383,1552.995 3574.705,1557.038 3573.28,1562.025
        <path d="M3525.86,1552.995 L3565.86,1552.995 C3571.383,1552.995 3574.705,1557.038 3573.28,1562.025
      </g>
      <path d="M3533.003,1562.995 L3553.003,1562.995 C3558.526,1562.995 3561.848,1567.038 3560.423,1572.025
    </g>
    <g>
      <g>
        <path d="M3515.86,1612.995 L3555.86,1612.995 C3561.383,1612.995 3564.705,1617.038 3563.28,1622.026
        <path d="M3515.86,1612.995 L3555.86,1612.995 C3561.383,1612.995 3564.705,1617.038 3563.28,1622.026
      </g>
      <path d="M3523.003,1622.995 L3543.003,1622.995 C3548.526,1622.995 3551.848,1627.038 3550.423,1632.025
    </g>
    <g>
      <g>
        <path d="M3555.86,1582.995 L3595.86,1582.995 C3601.383,1582.995 3604.705,1587.038 3603.28,1592.025
        <path d="M3555.86,1582.995 L3595.86,1582.995 C3601.383,1582.995 3604.705,1587.038 3603.28,1592.025
      </g>
      <path d="M3563.003,1592.995 L3583.003,1592.995 C3588.526,1592.995 3591.848,1597.038 3590.423,1602.025
    </g>
  </g>
  <g>
    <g>
      <path d="M284,364 L444,364 C466.091,364 484,381.909 484,404 L484,684 C484,706.091 466.091,724 444,724
      <path d="M284,364 L444,364 C466.091,364 484,381.909 484,404 L484,684 C484,706.091 466.091,724 444,724
    </g>
    <path d="M284,384 L444,384 C455.046,384 464,392.954 464,404 L464,684 C464,695.046 455.046,704 444,704 L
    <path d="M424,444 C335.636,444 264,515.636 264,604 C264,626.092 281.908,644 304,644 C392.364,644 464,57
    <path d="M356,484 L336,504 L336,528 L356,508 L356,604 L376,604 L376,484 z" fill="#FFAA00"/>
    <path d="M284,394 L274,404 L274,416 L284,406 L284,454 L294,454 L294,394 z M444,694 L454,684 L454,672 L4
  </g>
  <g>
    <g>
      <path d="M41,364 L201,364 C223.091,364 241,381.909 241,404 L241,684 C241,706.091 223.091,724 201,724
      <path d="M41,364 L201,364 C223.091,364 241,381.909 241,404 L241,684 C241,706.091 223.091,724 201,724
    </g>
    <path d="M41,384 L201,384 C212.046,384 221,392.954 221,404 L221,684 C221,695.046 212.046,704 201,704 L4
    <path d="M181,444 C92.636,444 21,515.636 21,604 C21,626.092 38.908,644 61,644 C149.364,644 221,572.368
    <path d="M121,484 C98.908,484 81,501.908 81,524 L81,564 C81,586.092 98.908,604 121,604 C143.092,604 161
    <path d="M51,394 C39.954,394 31,402.954 31,414 L31,434 C31,445.046 39.954,454 51,454 C62.046,454 71,445
  </g>
  <g>
    <g>
      <path d="M1742,364 L1902,364 C1924.091,364 1942,381.909 1942,404 L1942,684 C1942,706.091 1924.091,724
      <path d="M1742,364 L1902,364 C1924.091,364 1942,381.909 1942,404 L1942,684 C1942,706.091 1924.091,724
    </g>
    <path d="M1742,384 L1902,384 C1913.046,384 1922,392.954 1922,404 L1922,684 C1922,695.046 1913.046,704 1
    <path d="M1882,444 C1793.636,444 1722,515.636 1722,604 C1722,626.092 1739.908,644 1762,644 C1850.364,64
    <path d="M1788,484 L1788,524 L1808,524 L1808,504 L1848,504 L1808,604 L1828,604 L1868,504 L1868,484 L178
    <path d="M1732,394 L1732,414 L1742,414 L1742,404 L1762,404 L1742,454 L1752,454 L1772,404 L1772,394 L173
  </g>
  <g>
    <g>
      <path d="M527,364 L687,364 C709.091,364 727,381.909 727,404 L727,684 C727,706.091 709.091,724 687,724
```

```
      <path d="M527,364 L687,364 C709.091,364 727,381.909 727,404 L727,684 C727,706.091 709.091,724 687,724
    </g>
    <path d="M527,384 L687,384 C698.046,384 707,392.954 707,404 L707,684 C707,695.046 698.046,704 687,704 L
    <path d="M667,444 C578.636,444 507,515.636 507,604 C507,626.092 524.908,644 547,644 C635.364,644 707,57
    <path d="M603,484 C580.84,484 563,501.84 563,524 L563,528 L583,528 L583,524 C583,512.92 591.92,504 603,
    <path d="M537,394 C525.92,394 517,402.92 517,414 L517,416 L527,416 L527,414 C527,408.46 531.46,404 537,
  </g>
  <g>
    <g>
      <path d="M770,364 L930,364 C952.091,364 970,381.909 970,404 L970,684 C970,706.091 952.091,724 930,724
      <path d="M770,364 L930,364 C952.091,364 970,381.909 970,404 L970,684 C970,706.091 952.091,724 930,724
    </g>
    <path d="M770,384 L930,384 C941.046,384 950,392.954 950,404 L950,684 C950,695.046 941.046,704 930,704 L
    <path d="M910,444 C821.636,444 750,515.636 750,604 C750,626.092 767.908,644 790,644 C878.364,644 950,57
    <path d="M850,484 C833.431,484 820,497.431 820,514 L840,514 C840,508.477 844.477,504 850,504 C855.523,5
    <path d="M780,394 C771.716,394 765,400.716 765,409 L775,409 C775,406.239 777.239,404 780,404 C782.761,4
  </g>
  <g>
    <g>
      <path d="M1013,364 L1173,364 C1195.091,364 1213,381.909 1213,404 L1213,684 C1213,706.091 1195.091,724
      <path d="M1013,364 L1173,364 C1195.091,364 1213,381.909 1213,404 L1213,684 C1213,706.091 1195.091,724
    </g>
    <path d="M1013,384 L1173,384 C1184.046,384 1193,392.954 1193,404 L1193,684 C1193,695.046 1184.046,704 1
    <path d="M1153,444 C1064.636,444 993,515.636 993,604 C993,626.092 1010.908,644 1033,644 C1121.364,644 1
    <path d="M1088.2,484 L1049,564 L1049,584 L1097,584 L1097,604 L1117,604 L1117,584 L1129,584 L1129,564 L1
    <path d="M1022.2,394 L1002.6,434 L1002.6,444 L1026.6,444 L1026.6,454 L1036.6,454 L1036.6,444 L1042.6,44
  </g>
  <g>
    <g>
      <path d="M1256,364 L1416,364 C1438.091,364 1456,381.909 1456,404 L1456,684 C1456,706.091 1438.091,724
      <path d="M1256,364 L1416,364 C1438.091,364 1456,381.909 1456,404 L1456,684 C1456,706.091 1438.091,724
    </g>
    <path d="M1256,384 L1416,384 C1427.046,384 1436,392.954 1436,404 L1436,684 C1436,695.046 1427.046,704 1
    <path d="M1396,444 C1307.636,444 1236,515.636 1236,604 C1236,626.092 1253.908,644 1276,644 C1364.364,64
    <path d="M1300,484 L1300,544 L1340,544 C1351.08,544 1360,552.92 1360,564 C1360,575.08 1351.08,584 1340,
    <path d="M1246,394 L1246,424 L1266,424 C1271.54,424 1276,428.46 1276,434 C1276,439.54 1271.54,444 1266,
  </g>
  <g>
    <g>
      <path d="M1499,364 L1659,364 C1681.091,364 1699,381.909 1699,404 L1699,684 C1699,706.091 1681.091,724
      <path d="M1499,364 L1659,364 C1681.091,364 1699,381.909 1699,404 L1699,684 C1699,706.091 1681.091,724
    </g>
    <path d="M1499,384 L1659,384 C1670.046,384 1679,392.954 1679,404 L1679,684 C1679,695.046 1670.046,704 1
    <path d="M1639,444 C1550.636,444 1479,515.636 1479,604 C1479,626.092 1496.908,644 1519,644 C1607.364,64
    <path d="M1579,484 C1556.908,484 1539,501.908 1539,524 L1539,564 C1539,586.092 1556.908,604 1579,604 C1
    <path d="M1509,394 C1497.954,394 1489,402.954 1489,414 L1489,434 C1488.996,443.053 1495.074,450.98 1503
  </g>
  <g>
    <g>
      <path d="M1985,364 L2145,364 C2167.091,364 2185,381.909 2185,404 L2185,684 C2185,706.091 2167.091,724
      <path d="M1985,364 L2145,364 C2167.091,364 2185,381.909 2185,404 L2185,684 C2185,706.091 2167.091,724
    </g>
    <path d="M1985,384 L2145,384 C2156.046,384 2165,392.954 2165,404 L2165,684 C2165,695.046 2156.046,704 2
    <path d="M2125,444 C2036.636,444 1965,515.636 1965,604 C1965,626.092 1982.908,644 2005,644 C2093.364,64
    <path d="M2065,484 C2048.431,484 2035,497.431 2035,514 C2035,520.796 2037.32,526.972 2041.124,532 C2031
    <path d="M1995,394 C1986.716,394 1980,400.716 1980,409 C1980,412.4 1981.16,415.484 1983.064,418 C1977.9
  </g>
  <g>
    <g>
      <path d="M2228,364 L2388,364 C2410.091,364 2428,381.909 2428,404 L2428,684 C2428,706.091 2410.091,724
      <path d="M2228,364 L2388,364 C2410.091,364 2428,381.909 2428,404 L2428,684 C2428,706.091 2410.091,724
    </g>
    <path d="M2228,384 L2388,384 C2399.046,384 2408,392.954 2408,404 L2408,684 C2408,695.046 2399.046,704 2
    <path d="M2368,444 C2279.636,444 2208,515.636 2208,604 C2208,626.092 2225.908,644 2248,644 C2336.364,64
    <path d="M2308,484 C2285.908,484 2268,501.908 2268,524 C2268,546.092 2285.908,564 2308,564 C2315.268,56
    <path d="M2238,454 C2249.046,454 2258,445.046 2258,434 L2258,414 C2258.004,404.947 2251.926,397.02 2243
  </g>
  <g>
    <g>
      <path d="M2471,364 L2631,364 C2653.091,364 2671,381.909 2671,404 L2671,684 C2671,706.091 2653.091,724
      <path d="M2471,364 L2631,364 C2653.091,364 2671,381.909 2671,404 L2671,684 C2671,706.091 2653.091,724
    </g>
    <path d="M2471,384 L2631,384 C2642.046,384 2651,392.954 2651,404 L2651,684 C2651,695.046 2642.046,704 2
    <path d="M2611,444 C2522.636,444 2451,515.636 2451,604 C2451,626.092 2468.908,644 2491,644 C2579.364,64
    <path d="M2551.124,484 C2535.194,483.951 2519.901,490.248 2508.624,501.5 C2485.164,524.9 2485.1,562.912
    <path d="M2491.064,394 C2483.1,393.974 2475.453,397.122 2469.816,402.748 C2458.079,414.447 2458.05,433.
  </g>
  <g>
    <g>
      <path d="M2714,364 L2874,364 C2896.091,364 2914,381.909 2914,404 L2914,684 C2914,706.091 2896.091,724
      <path d="M2714,364 L2874,364 C2896.091,364 2914,381.909 2914,404 L2914,684 C2914,706.091 2896.091,724
```

```
    </g>
    <path d="M2714,384 L2874,384 C2885.046,384 2894,392.954 2894,404 L2894,684 C2894,695.046 2885.046,704 2
    <path d="M2854,444 C2765.636,444 2694,515.636 2694,604 C2694,626.092 2711.908,644 2734,644 C2822.364,64
    <path d="M2794,479 L2804,489 L2764,529 C2754,539 2754,559 2764,569 L2784,549 L2824,509 L2834,519 L2834,
    <path d="M2794,609 L2784,599 L2824,559 C2834,549 2834,529 2824,519 L2804,539 L2764,579 L2754,569 L2754,
    <path d="M2724,394 L2729,399 L2709,419 C2704,424 2704,434 2709,439 L2719,429 L2739,409 L2744,414 L2744,
    <path d="M2724,459 L2719,454 L2739,434 C2744,429 2744,419 2739,414 L2729,424 L2709,444 L2704,439 L2704,
    <path d="M2864,694 L2859,689 L2879,669 C2884,664 2884,654 2879,649 L2869,659 L2849,679 L2844,674 L2844,
  </g>
  <g>
    <g>
      <path d="M284,727 L444,727 C466.091,727 484,744.909 484,767 L484,1047 C484,1069.091 466.091,1087 444,
      <path d="M284,727 L444,727 C466.091,727 484,744.909 484,767 L484,1047 C484,1069.091 466.091,1087 444,
    </g>
    <path d="M284,747 L444,747 C455.046,747 464,755.954 464,767 L464,1047 C464,1058.046 455.046,1067 444,10
    <path d="M424,807 C335.636,807 264,878.636 264,967 C264,989.092 281.908,1007 304,1007 C392.364,1007 464
    <path d="M356,847 L336,867 L336,891 L356,871 L356,967 L376,967 L376,847 z" fill="#55AA55"/>
    <path d="M284,757 L274,767 L274,779 L284,769 L284,817 L294,817 L294,757 z M444,1057 L454,1047 L454,1035
  </g>
  <g>
    <g>
      <path d="M41,727 L201,727 C223.091,727 241,744.909 241,767 L241,1047 C241,1069.092 223.091,1087 201,1
      <path d="M41,727 L201,727 C223.091,727 241,744.909 241,767 L241,1047 C241,1069.092 223.091,1087 201,1
    </g>
    <path d="M41,747 L201,747 C212.046,747 221,755.955 221,767 L221,1047 C221,1058.046 212.046,1067 201,106
    <path d="M181,807 C92.636,807 21,878.636 21,967 C21,989.092 38.908,1007 61,1007 C149.364,1007 221,935.3
    <path d="M121,847 C98.908,847 81,864.908 81,887 L81,927 C81,949.092 98.908,967 121,967 C143.092,967 161
    <path d="M51,757 C39.954,757 31,765.955 31,777 L31,797 C31,808.046 39.954,817 51,817 C62.046,817 71,808
  </g>
  <g>
    <g>
      <path d="M1742,727 L1902,727 C1924.091,727 1942,744.909 1942,767 L1942,1047 C1942,1069.091 1924.091,1
      <path d="M1742,727 L1902,727 C1924.091,727 1942,744.909 1942,767 L1942,1047 C1942,1069.091 1924.091,1
    </g>
    <path d="M1742,747 L1902,747 C1913.046,747 1922,755.954 1922,767 L1922,1047 C1922,1058.046 1913.046,106
    <path d="M1882,807 C1793.636,807 1722,878.636 1722,967 C1722,989.092 1739.908,1007 1762,1007 C1850.364,
    <path d="M1788,847 L1788,887 L1808,887 L1808,867 L1848,867 L1808,967 L1828,967 L1868,867 L1868,847 L178
    <path d="M1732,757 L1732,777 L1742,777 L1742,767 L1762,767 L1742,817 L1752,817 L1772,767 L1772,757 L173
  </g>
  <g>
    <g>
      <path d="M527,727 L687,727 C709.091,727 727,744.909 727,767 L727,1047 C727,1069.091 709.091,1087 687,
      <path d="M527,727 L687,727 C709.091,727 727,744.909 727,767 L727,1047 C727,1069.091 709.091,1087 687,
    </g>
    <path d="M527,747 L687,747 C698.046,747 707,755.954 707,767 L707,1047 C707,1058.046 698.046,1067 687,10
    <path d="M667,807 C578.636,807 507,878.636 507,967 C507,989.092 524.908,1007 547,1007 C635.364,1007 707
    <path d="M603,847 C580.84,847 563,864.84 563,887 L563,891 L583,891 L583,887 C583,875.92 591.92,867 603,
    <path d="M537,757 C525.92,757 517,765.92 517,777 L517,779 L527,779 L527,777 C527,771.46 531.46,767 537,
  </g>
  <g>
    <g>
      <path d="M770,727 L930,727 C952.091,727 970,744.909 970,767 L970,1047 C970,1069.091 952.091,1087 930,
      <path d="M770,727 L930,727 C952.091,727 970,744.909 970,767 L970,1047 C970,1069.091 952.091,1087 930,
    </g>
    <path d="M770,747 L930,747 C941.046,747 950,755.954 950,767 L950,1047 C950,1058.046 941.046,1067 930,10
    <path d="M910,807 C821.636,807 750,878.636 750,967 C750,989.092 767.908,1007 790,1007 C878.364,1007 950
    <path d="M850,847 C833.431,847 820,860.431 820,877 L840,877 C840,871.477 844.477,867 850,867 C855.523,8
    <path d="M780,757 C771.716,757 765,763.716 765,772 L775,772 C775,769.239 777.239,767 780,767 C782.761,7
  </g>
  <g>
    <g>
      <path d="M1013,727 L1173,727 C1195.091,727 1213,744.909 1213,767 L1213,1047 C1213,1069.091 1195.091,1
      <path d="M1013,727 L1173,727 C1195.091,727 1213,744.909 1213,767 L1213,1047 C1213,1069.091 1195.091,1
    </g>
    <path d="M1013,747 L1173,747 C1184.046,747 1193,755.954 1193,767 L1193,1047 C1193,1058.046 1184.046,106
    <path d="M1153,807 C1064.636,807 993,878.636 993,967 C993,989.092 1010.908,1007 1033,1007 C1121.364,100
    <path d="M1088.2,847 L1049,927 L1049,947 L1097,947 L1097,967 L1117,967 L1117,947 L1129,947 L1129,927 L1
    <path d="M1022.2,757 L1002.6,797 L1002.6,807 L1026.6,807 L1026.6,817 L1036.6,817 L1036.6,807 L1042.6,80
  </g>
  <g>
    <g>
      <path d="M1256,727 L1416,727 C1438.091,727 1456,744.909 1456,767 L1456,1047 C1456,1069.091 1438.091,1
      <path d="M1256,727 L1416,727 C1438.091,727 1456,744.909 1456,767 L1456,1047 C1456,1069.091 1438.091,1
    </g>
    <path d="M1256,747 L1416,747 C1427.046,747 1436,755.954 1436,767 L1436,1047 C1436,1058.046 1427.046,106
    <path d="M1396,807 C1307.636,807 1236,878.636 1236,967 C1236,989.092 1253.908,1007 1276,1007 C1364.364,
    <path d="M1300,847 L1300,907 L1340,907 C1351.08,907 1360,915.92 1360,927 C1360,938.08 1351.08,947 1340,
    <path d="M1246,757 L1246,787 L1266,787 C1271.54,787 1276,791.46 1276,797 C1276,802.54 1271.54,807 1266,
  </g>
  <g>
    <g>
```

```
      <path d="M1499,727 L1659,727 C1681.091,727 1699,744.909 1699,767 L1699,1047 C1699,1069.091 1681.091,1
      <path d="M1499,727 L1659,727 C1681.091,727 1699,744.909 1699,767 L1699,1047 C1699,1069.091 1681.091,1
    </g>
    <path d="M1499,747 L1659,747 C1670.046,747 1679,755.954 1679,767 L1679,1047 C1679,1058.046 1670.046,106
    <path d="M1639,807 C1550.636,807 1479,878.636 1479,967 C1479,989.092 1496.908,1007 1519,1007 C1607.364,
    <path d="M1579,847 C1556.908,847 1539,864.908 1539,887 L1539,927 C1539,949.092 1556.908,967 1579,967 C1
    <path d="M1509,757 C1497.954,757 1489,765.954 1489,777 L1489,797 C1488.996,806.053 1495.074,813.98 1503
  </g>
  <g>
    <g>
      <path d="M1985,727 L2145,727 C2167.091,727 2185,744.909 2185,767 L2185,1047 C2185,1069.091 2167.091,1
      <path d="M1985,727 L2145,727 C2167.091,727 2185,744.909 2185,767 L2185,1047 C2185,1069.091 2167.091,1
    </g>
    <path d="M1985,747 L2145,747 C2156.046,747 2165,755.954 2165,767 L2165,1047 C2165,1058.046 2156.046,106
    <path d="M2125,807 C2036.636,807 1965,878.636 1965,967 C1965,989.092 1982.908,1007 2005,1007 C2093.364,
    <path d="M2065,847 C2048.431,847 2035,860.431 2035,877 C2035,883.796 2037.32,889.972 2041.124,895 C2031
    <path d="M1995,757 C1986.716,757 1980,763.716 1980,772 C1980,775.4 1981.16,778.484 1983.064,781 C1977.9
  </g>
  <g>
    <g>
      <path d="M2228,727 L2388,727 C2410.091,727 2428,744.909 2428,767 L2428,1047 C2428,1069.091 2410.091,1
      <path d="M2228,727 L2388,727 C2410.091,727 2428,744.909 2428,767 L2428,1047 C2428,1069.091 2410.091,1
    </g>
    <path d="M2228,747 L2388,747 C2399.046,747 2408,755.954 2408,767 L2408,1047 C2408,1058.046 2399.046,106
    <path d="M2368,807 C2279.636,807 2208,878.636 2208,967 C2208,989.092 2225.908,1007 2248,1007 C2336.364,
    <path d="M2308,847 C2285.908,847 2268,864.908 2268,887 C2268,909.092 2285.908,927 2308,927 C2315.268,92
    <path d="M2238,817 C2249.046,817 2258,808.046 2258,797 L2258,777 C2258.004,767.947 2251.926,760.02 2243
  </g>
  <g>
    <g>
      <path d="M2471,727 L2631,727 C2653.091,727 2671,744.909 2671,767 L2671,1047 C2671,1069.091 2653.091,1
      <path d="M2471,727 L2631,727 C2653.091,727 2671,744.909 2671,767 L2671,1047 C2671,1069.091 2653.091,1
    </g>
    <path d="M2471,747 L2631,747 C2642.046,747 2651,755.954 2651,767 L2651,1047 C2651,1058.046 2642.046,106
    <path d="M2611,807 C2522.636,807 2451,878.636 2451,967 C2451,989.092 2468.908,1007 2491,1007 C2579.364,
    <path d="M2551.124,847 C2535.194,846.951 2519.901,853.248 2508.624,864.5 C2485.164,887.9 2485.1,925.912
    <path d="M2491.064,757 C2483.1,756.974 2475.453,760.122 2469.816,765.748 C2458.079,777.447 2458.05,796.
  </g>
  <g>
    <g>
      <path d="M2714,727 L2874,727 C2896.091,727 2914,744.909 2914,767 L2914,1047 C2914,1069.091 2896.091,1
      <path d="M2714,727 L2874,727 C2896.091,727 2914,744.909 2914,767 L2914,1047 C2914,1069.091 2896.091,1
    </g>
    <path d="M2714,747 L2874,747 C2885.046,747 2894,755.954 2894,767 L2894,1047 C2894,1058.046 2885.046,106
    <path d="M2854,807 C2765.636,807 2694,878.636 2694,967 C2694,989.092 2711.908,1007 2734,1007 C2822.364,
    <path d="M2794,842 L2804,852 L2764,892 C2754,902 2754,922 2764,932 L2784,912 L2824,872 L2834,882 L2834,
    <path d="M2794,972 L2784,962 L2824,922 C2834,912 2834,892 2824,882 L2804,902 L2764,942 L2754,932 L2754,
    <path d="M2724,757 L2729,762 L2709,782 C2704,787 2704,797 2709,802 L2719,792 L2739,772 L2744,777 L2744,
    <path d="M2724,822 L2719,817 L2739,797 C2744,792 2744,782 2739,777 L2729,787 L2709,807 L2704,802 L2704,
    <path d="M2864,1057 L2859,1052 L2879,1032 C2884,1027 2884,1017 2879,1012 L2869,1022 L2849,1042 L2844,10
  </g>
  <g>
    <g>
      <path d="M284,1090 L444,1090 C466.091,1090 484,1107.908 484,1130 L484,1410 C484,1432.091 466.091,1450
      <path d="M284,1090 L444,1090 C466.091,1090 484,1107.908 484,1130 L484,1410 C484,1432.091 466.091,1450
    </g>
    <path d="M284,1110 L444,1110 C455.046,1110 464,1118.954 464,1130 L464,1410 C464,1421.045 455.046,1430 4
    <path d="M424,1170 C335.636,1170 264,1241.636 264,1330 C264,1352.092 281.908,1370 304,1370 C392.364,137
    <path d="M356,1210 L336,1230 L336,1254 L356,1234 L356,1330 L376,1330 L376,1210 z" fill="#5555FF"/>
    <path d="M284,1120 L274,1130 L274,1142 L284,1132 L284,1180 L294,1180 L294,1120 z M444,1420 L454,1410 L4
  </g>
  <g>
    <g>
      <path d="M41,1090 L201,1090 C223.091,1090 241,1107.909 241,1130 L241,1410 C241,1432.091 223.091,1450
      <path d="M41,1090 L201,1090 C223.091,1090 241,1107.909 241,1130 L241,1410 C241,1432.091 223.091,1450
    </g>
    <path d="M41,1110 L201,1110 C212.046,1110 221,1118.954 221,1130 L221,1410 C221,1421.046 212.046,1430 20
    <path d="M181,1170 C92.636,1170 21,1241.636 21,1330 C21,1352.092 38.908,1370 61,1370 C149.364,1370 221,
    <path d="M121,1210 C98.908,1210 81,1227.908 81,1250 L81,1290 C81,1312.092 98.908,1330 121,1330 C143.092
    <path d="M51,1120 C39.954,1120 31,1128.954 31,1140 L31,1160 C31,1171.046 39.954,1180 51,1180 C62.046,11
  </g>
  <g>
    <g>
      <path d="M1742,1090 L1902,1090 C1924.091,1090 1942,1107.908 1942,1130 L1942,1410 C1942,1432.091 1924.
      <path d="M1742,1090 L1902,1090 C1924.091,1090 1942,1107.908 1942,1130 L1942,1410 C1942,1432.091 1924.
    </g>
    <path d="M1742,1110 L1902,1110 C1913.046,1110 1922,1118.954 1922,1130 L1922,1410 C1922,1421.045 1913.04
    <path d="M1882,1170 C1793.636,1170 1722,1241.636 1722,1330 C1722,1352.092 1739.908,1370 1762,1370 C185
    <path d="M1788,1210 L1788,1250 L1808,1250 L1808,1230 L1848,1230 L1808,1330 L1828,1330 L1868,1230 L1868,
    <path d="M1732,1120 L1732,1140 L1742,1140 L1742,1130 L1762,1130 L1742,1180 L1752,1180 L1772,1130 L1772,
  </g>
```

```
    <g>
      <g>
        <path d="M527,1090 L687,1090 C709.091,1090 727,1107.908 727,1130 L727,1410 C727,1432.091 709.091,1450
        <path d="M527,1090 L687,1090 C709.091,1090 727,1107.908 727,1130 L727,1410 C727,1432.091 709.091,1450
      </g>
      <path d="M527,1110 L687,1110 C698.046,1110 707,1118.954 707,1130 L707,1410 C707,1421.045 698.046,1430 6
      <path d="M667,1170 C578.636,1170 507,1241.636 507,1330 C507,1352.092 524.908,1370 547,1370 C635.364,137
      <path d="M603,1210 C580.84,1210 563,1227.84 563,1250 L563,1254 L583,1254 L583,1250 C583,1238.92 591.92,
      <path d="M537,1120 C525.92,1120 517,1128.92 517,1140 L517,1142 L527,1142 L527,1140 C527,1134.46 531.46,
    </g>
    <g>
      <g>
        <path d="M770,1090 L930,1090 C952.091,1090 970,1107.908 970,1130 L970,1410 C970,1432.091 952.091,1450
        <path d="M770,1090 L930,1090 C952.091,1090 970,1107.908 970,1130 L970,1410 C970,1432.091 952.091,1450
      </g>
      <path d="M770,1110 L930,1110 C941.046,1110 950,1118.954 950,1130 L950,1410 C950,1421.045 941.046,1430 9
      <path d="M910,1170 C821.636,1170 750,1241.636 750,1330 C750,1352.092 767.908,1370 790,1370 C878.364,137
      <path d="M850,1210 C833.431,1210 820,1223.431 820,1240 L840,1240 C840,1234.477 844.477,1230 850,1230 C8
      <path d="M780,1120 C771.716,1120 765,1126.715 765,1135 L775,1135 C775,1132.238 777.239,1130 780,1130 C7
    </g>
    <g>
      <g>
        <path d="M1013,1090 L1173,1090 C1195.091,1090 1213,1107.908 1213,1130 L1213,1410 C1213,1432.091 1195.
        <path d="M1013,1090 L1173,1090 C1195.091,1090 1213,1107.908 1213,1130 L1213,1410 C1213,1432.091 1195.
      </g>
      <path d="M1013,1110 L1173,1110 C1184.046,1110 1193,1118.954 1193,1130 L1193,1410 C1193,1421.045 1184.04
      <path d="M1153,1170 C1064.636,1170 993,1241.636 993,1330 C993,1352.092 1010.908,1370 1033,1370 C1121.36
      <path d="M1088.2,1210 L1049,1290 L1049,1310 L1097,1310 L1097,1330 L1117,1330 L1117,1310 L1129,1310 L112
      <path d="M1022.2,1120 L1002.6,1160 L1002.6,1170 L1026.6,1170 L1026.6,1180 L1036.6,1180 L1036.6,1170 L10
    </g>
    <g>
      <g>
        <path d="M1256,1090 L1416,1090 C1438.091,1090 1456,1107.908 1456,1130 L1456,1410 C1456,1432.091 1438.
        <path d="M1256,1090 L1416,1090 C1438.091,1090 1456,1107.908 1456,1130 L1456,1410 C1456,1432.091 1438.
      </g>
      <path d="M1256,1110 L1416,1110 C1427.046,1110 1436,1118.954 1436,1130 L1436,1410 C1436,1421.045 1427.04
      <path d="M1396,1170 C1307.636,1170 1236,1241.636 1236,1330 C1236,1352.092 1253.908,1370 1276,1370 C1364
      <path d="M1300,1210 L1300,1270 L1340,1270 C1351.08,1270 1360,1278.92 1360,1290 C1360,1301.08 1351.08,13
      <path d="M1246,1120 L1246,1150 L1266,1150 C1271.54,1150 1276,1154.46 1276,1160 C1276,1165.54 1271.54,11
    </g>
    <g>
      <g>
        <path d="M1499,1090 L1659,1090 C1681.091,1090 1699,1107.908 1699,1130 L1699,1410 C1699,1432.091 1681.
        <path d="M1499,1090 L1659,1090 C1681.091,1090 1699,1107.908 1699,1130 L1699,1410 C1699,1432.091 1681.
      </g>
      <path d="M1499,1110 L1659,1110 C1670.046,1110 1679,1118.954 1679,1130 L1679,1410 C1679,1421.045 1670.04
      <path d="M1639,1170 C1550.636,1170 1479,1241.636 1479,1330 C1479,1352.092 1496.908,1370 1519,1370 C1607
      <path d="M1579,1210 C1556.908,1210 1539,1227.908 1539,1250 L1539,1290 C1539,1312.092 1556.908,1330 1579
      <path d="M1509,1120 C1497.954,1120 1489,1128.954 1489,1140 L1489,1160 C1488.996,1169.052 1495.074,1176.
    </g>
    <g>
      <g>
        <path d="M1985,1090 L2145,1090 C2167.091,1090 2185,1107.908 2185,1130 L2185,1410 C2185,1432.091 2167.
        <path d="M1985,1090 L2145,1090 C2167.091,1090 2185,1107.908 2185,1130 L2185,1410 C2185,1432.091 2167.
      </g>
      <path d="M1985,1110 L2145,1110 C2156.046,1110 2165,1118.954 2165,1130 L2165,1410 C2165,1421.045 2156.04
      <path d="M2125,1170 C2036.636,1170 1965,1241.636 1965,1330 C1965,1352.092 1982.908,1370 2005,1370 C2093
      <path d="M2065,1210 C2048.431,1210 2035,1223.431 2035,1240 C2035,1246.796 2037.32,1252.972 2041.124,125
      <path d="M1995,1120 C1986.716,1120 1980,1126.715 1980,1135 C1980,1138.4 1981.16,1141.484 1983.064,1144
    </g>
    <g>
      <g>
        <path d="M2228,1090 L2388,1090 C2410.091,1090 2428,1107.908 2428,1130 L2428,1410 C2428,1432.091 2410.
        <path d="M2228,1090 L2388,1090 C2410.091,1090 2428,1107.908 2428,1130 L2428,1410 C2428,1432.091 2410.
      </g>
      <path d="M2228,1110 L2388,1110 C2399.046,1110 2408,1118.954 2408,1130 L2408,1410 C2408,1421.045 2399.04
      <path d="M2368,1170 C2279.636,1170 2208,1241.636 2208,1330 C2208,1352.092 2225.908,1370 2248,1370 C2336
      <path d="M2308,1210 C2285.908,1210 2268,1227.908 2268,1250 C2268,1272.092 2285.908,1290 2308,1290 C2315
      <path d="M2238,1180 C2249.046,1180 2258,1171.045 2258,1160 L2258,1140 C2258.004,1130.947 2251.926,1123.
    </g>
    <g>
      <g>
        <path d="M2471,1090 L2631,1090 C2653.091,1090 2671,1107.908 2671,1130 L2671,1410 C2671,1432.091 2653.
        <path d="M2471,1090 L2631,1090 C2653.091,1090 2671,1107.908 2671,1130 L2671,1410 C2671,1432.091 2653.
      </g>
      <path d="M2471,1110 L2631,1110 C2642.046,1110 2651,1118.954 2651,1130 L2651,1410 C2651,1421.045 2642.04
      <path d="M2611,1170 C2522.636,1170 2451,1241.636 2451,1330 C2451,1352.092 2468.908,1370 2491,1370 C2579
      <path d="M2551.124,1210 C2535.194,1209.95 2519.901,1216.248 2508.624,1227.5 C2485.164,1250.9 2485.1,128
      <path d="M2491.064,1120 C2483.1,1119.974 2475.453,1123.122 2469.816,1128.748 C2458.079,1140.447 2458.05
    </g>
    <g>
```

```
    <g>
      <path d="M2714,1090 L2874,1090 C2896.091,1090 2914,1107.908 2914,1130 L2914,1410 C2914,1432.091 2896.
      <path d="M2714,1090 L2874,1090 C2896.091,1090 2914,1107.908 2914,1130 L2914,1410 C2914,1432.091 2896.
    </g>
    <path d="M2714,1110 L2874,1110 C2885.046,1110 2894,1118.954 2894,1130 L2894,1410 C2894,1421.045 2885.04
    <path d="M2854,1170 C2765.636,1170 2694,1241.636 2694,1330 C2694,1352.092 2711.908,1370 2734,1370 C2822
    <path d="M2794,1205 L2804,1215 L2764,1255 C2754,1265 2754,1285 2764,1295 L2784,1275 L2824,1235 L2834,12
    <path d="M2794,1335 L2784,1325 L2824,1285 C2834,1275 2834,1255 2824,1245 L2804,1265 L2764,1305 L2754,12
    <path d="M2724,1120 L2729,1125 L2709,1145 C2704,1150 2704,1160 2709,1165 L2719,1155 L2739,1135 L2744,11
    <path d="M2724,1185 L2719,1180 L2739,1160 C2744,1155 2744,1145 2739,1140 L2729,1150 L2709,1170 L2704,11
    <path d="M2864,1420 L2859,1415 L2879,1395 C2884,1390 2884,1380 2879,1375 L2869,1385 L2849,1405 L2844,14
  </g>
  <g>
    <g>
      <path d="M2957,1090 L3117,1090 C3139.091,1090 3157,1107.908 3157,1130 L3157,1410 C3157,1432.091 3139.
      <path d="M2957,1090 L3117,1090 C3139.091,1090 3157,1107.908 3157,1130 L3157,1410 C3157,1432.091 3139.
    </g>
    <path d="M2957,1110 L3117,1110 C3128.046,1110 3137,1118.954 3137,1130 L3137,1410 C3137,1421.045 3128.04
    <path d="M3097,1170 C3008.636,1170 2937,1241.636 2937,1330 C2937,1352.092 2954.908,1370 2977,1370 C3065
    <g>
      <path d="M3009.894,1239.986 L3049.894,1239.986 C3055.417,1239.986 3058.739,1244.03 3057.314,1249.017
      <path d="M3009.894,1239.986 L3049.894,1239.986 C3055.417,1239.986 3058.739,1244.03 3057.314,1249.017
    </g>
    <path d="M3017.037,1249.986 L3037.037,1249.986 C3042.56,1249.986 3045.882,1254.029 3044.457,1259.017 L3
    <g>
      <path d="M3049.894,1209.983 L3089.894,1209.983 C3095.417,1209.983 3098.739,1214.026 3097.314,1219.014
      <path d="M3049.894,1209.983 L3089.894,1209.983 C3095.417,1209.983 3098.739,1214.026 3097.314,1219.014
    </g>
    <path d="M3057.037,1219.983 L3077.037,1219.983 C3082.56,1219.983 3085.882,1224.026 3084.457,1229.014 L3
    <path d="M2951,1140 L2951,1150 L2941,1150 L2941,1160 L2951,1160 L2951,1170 L2961,1170 L2961,1160 L2971,
    <path d="M2997,1120 C2985.92,1120 2977,1128.92 2977,1140 L2977,1142 L2987,1142 L2987,1140 C2987,1134.46
    <path d="M3123,1400 L3123,1390 L3133,1390 L3133,1380 L3123,1380 L3123,1370 L3113,1370 L3113,1380 L3103,
    <path d="M3077,1420 C3088.08,1420 3097,1411.08 3097,1400 L3097,1398 L3087,1398 L3087,1400 C3087,1405.54
  </g>
  <g>
    <g>
      <path d="M2957,727 L3117,727 C3139.091,727 3157,744.909 3157,767 L3157,1047 C3157,1069.091 3139.091,1
      <path d="M2957,727 L3117,727 C3139.091,727 3157,744.909 3157,767 L3157,1047 C3157,1069.091 3139.091,1
    </g>
    <path d="M2957,747 L3117,747 C3128.046,747 3137,755.954 3137,767 L3137,1047 C3137,1058.046 3128.046,106
    <path d="M3097,807 C3008.636,807 2937,878.636 2937,967 C2937,989.092 2954.908,1007 2977,1007 C3065.364,
    <g>
      <path d="M3009.893,876.986 L3049.893,876.986 C3055.416,876.986 3058.738,881.029 3057.313,886.016 L303
      <path d="M3009.893,876.986 L3049.893,876.986 C3055.416,876.986 3058.738,881.029 3057.313,886.016 L303
    </g>
    <path d="M3017.036,886.986 L3037.036,886.986 C3042.559,886.986 3045.881,891.029 3044.456,896.016 L3029.
    <g>
      <path d="M3049.892,846.987 L3089.892,846.987 C3095.415,846.987 3098.737,851.03 3097.312,856.017 L3076
      <path d="M3049.892,846.987 L3089.892,846.987 C3095.415,846.987 3098.737,851.03 3097.312,856.017 L3076
    </g>
    <path d="M3057.035,856.986 L3077.035,856.986 C3082.558,856.986 3085.88,861.029 3084.455,866.017 L3069.6
    <path d="M2951,777 L2951,787 L2941,787 L2941,797 L2951,797 L2951,807 L2961,807 L2961,797 L2971,797 L297
    <path d="M2997,757 C2985.92,757 2977,765.92 2977,777 L2977,779 L2987,779 L2987,777 C2987,771.46 2991.46
    <path d="M3123,1037 L3123,1027 L3133,1027 L3133,1017 L3123,1017 L3123,1007 L3113,1007 L3113,1017 L3103,
    <path d="M3077,1057 C3088.08,1057 3097,1048.08 3097,1037 L3097,1035 L3087,1035 L3087,1037 C3087,1042.54
  </g>
  <g>
    <g>
      <path d="M2957,364 L3117,364 C3139.091,364 3157,381.909 3157,404 L3157,684 C3157,706.091 3139.091,724
      <path d="M2957,364 L3117,364 C3139.091,364 3157,381.909 3157,404 L3157,684 C3157,706.091 3139.091,724
    </g>
    <path d="M2957,384 L3117,384 C3128.046,384 3137,392.954 3137,404 L3137,684 C3137,695.046 3128.046,704 3
    <path d="M3097,444 C3008.636,444 2937,515.636 2937,604 C2937,626.092 2954.908,644 2977,644 C3065.364,64
    <g>
      <path d="M3009.886,513.989 L3049.886,513.989 C3055.409,513.989 3058.731,518.032 3057.306,523.02 L3036
      <path d="M3009.886,513.989 L3049.886,513.989 C3055.409,513.989 3058.731,518.032 3057.306,523.02 L3036
    </g>
    <path d="M3017.03,523.989 L3037.03,523.989 C3042.552,523.989 3045.874,528.032 3044.449,533.019 L3029.61
    <g>
      <path d="M3049.885,483.99 L3089.885,483.99 C3095.408,483.99 3098.73,488.033 3097.305,493.02 L3076.752
      <path d="M3049.885,483.99 L3089.885,483.99 C3095.408,483.99 3098.73,488.033 3097.305,493.02 L3076.752
    </g>
    <path d="M3057.029,493.989 L3077.029,493.989 C3082.552,493.989 3085.873,498.033 3084.448,503.02 L3069.6
    <path d="M2951,414 L2951,424 L2941,424 L2941,434 L2951,434 L2951,444 L2961,444 L2961,434 L2971,434 L297
    <path d="M2997,394 C2985.92,394 2977,402.92 2977,414 L2977,416 L2987,416 L2987,414 C2987,408.46 2991.46
    <path d="M3123,674 L3123,664 L3133,664 L3133,654 L3123,654 L3123,644 L3113,644 L3113,654 L3103,654 L310
    <path d="M3077,694 C3088.08,694 3097,685.08 3097,674 L3097,672 L3087,672 L3087,674 C3087,679.54 3082.54
  </g>
  <g>
    <g>
      <path d="M2957,1 L3117,1 C3139.091,1 3157,18.909 3157,41 L3157,321 C3157,343.091 3139.091,361 3117,36
```

```
      <path d="M2957,1 L3117,1 C3139.091,1 3157,18.909 3157,41 L3157,321 C3157,343.091 3139.091,361 3117,36
    </g>
    <path d="M2957,21 L3117,21 C3128.046,21 3137,29.954 3137,41 L3137,321 C3137,332.046 3128.046,341 3117,3
    <path d="M3097,81 C3008.636,81 2937,152.636 2937,241 C2937,263.092 2954.908,281 2977,281 C3065.364,281
    <g>
      <path d="M3009.885,150.988 L3049.885,150.988 C3055.408,150.988 3058.73,155.031 3057.305,160.019 L3036
      <path d="M3009.885,150.988 L3049.885,150.988 C3055.408,150.988 3058.73,155.031 3057.305,160.019 L3036
    </g>
    <path d="M3017.028,160.988 L3037.028,160.988 C3042.551,160.988 3045.873,165.031 3044.448,170.019 L3029.
    <g>
      <path d="M3049.884,120.989 L3089.884,120.989 C3095.407,120.989 3098.729,125.032 3097.304,130.019 L307
      <path d="M3049.884,120.989 L3089.884,120.989 C3095.407,120.989 3098.729,125.032 3097.304,130.019 L307
    </g>
    <path d="M3057.027,130.989 L3077.027,130.989 C3082.55,130.989 3085.872,135.032 3084.447,140.019 L3069.6
    <path d="M2951,51 L2951,61 L2941,61 L2941,71 L2951,71 L2951,81 L2961,81 L2961,71 L2971,71 L2971,61 L296
    <path d="M2997,31 C2985.92,31 2977,39.92 2977,51 L2977,53 L2987,53 L2987,51 C2987,45.46 2991.46,41 2997
    <path d="M3123,311 L3123,301 L3133,301 L3133,291 L3123,291 L3123,281 L3113,281 L3113,291 L3103,291 L310
    <path d="M3077,331 C3088.08,331 3097,322.08 3097,311 L3097,309 L3087,309 L3087,311 C3087,316.54 3082.54
  </g>
  <g>
    <g>
      <path d="M3200,1453 L3360,1453 C3382.091,1453 3400,1470.909 3400,1493 L3400,1773 C3400,1795.091 3382.
      <path d="M3200,1453 L3360,1453 C3382.091,1453 3400,1470.909 3400,1493 L3400,1773 C3400,1795.091 3382.
    </g>
    <path d="M3200,1473 L3360,1473 C3371.046,1473 3380,1481.954 3380,1493 L3380,1773 C3380,1784.046 3371.04
    <path d="M3340,1533 C3251.634,1533 3180,1604.635 3180,1693 C3180,1715.092 3197.909,1733 3220,1733 C3308
    <path d="M3191.75,1633 C3184.237,1651.542 3180,1671.764 3180,1693 C3180,1715.092 3197.909,1733 3220,173
    <path d="M3280,1633 L3220,1733 C3287.129,1733 3344.502,1691.61 3368.25,1633 L3280,1633 z" fill="#00AA00
    <path d="M3340,1533 C3272.871,1533 3215.498,1574.39 3191.75,1633 L3280,1633 L3340,1533 z" fill="#FF5555
    <path d="M3340,1533 L3280,1633 L3368.25,1633 C3375.763,1614.458 3380,1594.237 3380,1573 C3380,1550.909
    <path d="M3192.938,1508 C3191,1512.763 3190.002,1517.857 3190,1523 C3190,1528.523 3194.477,1533 3200,15
    <path d="M3215,1508 L3200,1533 C3216.782,1533 3231.126,1522.653 3237.063,1508 L3215,1508 z" fill="#00AA
    <path d="M3230,1483 C3213.218,1483 3198.874,1493.347 3192.937,1508 L3215,1508 L3230,1483 z" fill="#FF55
    <path d="M3230,1483 L3215,1508 L3237.063,1508 C3239.001,1503.236 3239.998,1498.143 3240,1493 C3240,1487
    <path d="M3322.938,1758 C3321,1762.763 3320.002,1767.857 3320,1773 C3320,1778.523 3324.477,1783 3330,17
    <path d="M3345,1758 L3330,1783 C3346.782,1783 3361.126,1772.653 3367.063,1758 L3345,1758 z" fill="#00AA
    <path d="M3360,1733 C3343.218,1733 3328.874,1743.347 3322.937,1758 L3345,1758 L3360,1733 z" fill="#FF55
    <path d="M3360,1733 L3345,1758 L3367.063,1758 C3369.001,1753.236 3369.998,1748.143 3370,1743 C3370,1737
    <path d="M3340,1533 C3272.871,1533 3215.498,1574.39 3191.75,1633 C3184.237,1651.542 3180,1671.763 3180,
  </g>
  <g>
    <g>
      <path d="M2957,1453 L3117,1453 C3139.091,1453 3157,1470.909 3157,1493 L3157,1773 C3157,1795.091 3139.
      <path d="M2957,1453 L3117,1453 C3139.091,1453 3157,1470.909 3157,1493 L3157,1773 C3157,1795.091 3139.
    </g>
    <path d="M2957,1473 L3117,1473 C3128.046,1473 3137,1481.954 3137,1493 L3137,1773 C3137,1784.046 3128.04
    <path d="M3097,1533 C3008.636,1533 2937,1604.636 2937,1693 C2937,1715.092 2954.908,1733 2977,1733 C3065
    <g>
      <text transform="matrix(0.715, -0.7, 0.7, 0.715, 3040.547, 1632.188)">
        <tspan x="-100.498" y="30" font-family="Arial-BoldMT" font-size="90" fill="#FFFFFF">UNO</tspan>
      </text>
    </g>
  </g>
  <g>
    <g>
      <path d="M3200,1 L3360,1 C3382.091,1 3400,18.909 3400,41 L3400,321 C3400,343.091 3382.091,361 3360,36
      <path d="M3200,1 L3360,1 C3382.091,1 3400,18.909 3400,41 L3400,321 C3400,343.091 3382.091,361 3360,36
    </g>
    <path d="M3200,21 L3360,21 C3371.046,21 3380,29.954 3380,41 L3380,321 C3380,332.046 3371.046,341 3360,3
    <path d="M3340,81 C3251.634,81 3180,152.635 3180,241 C3180,263.092 3197.909,281 3220,281 C3308.366,281
    <path d="M3191.75,181 C3184.237,199.542 3180,219.764 3180,241 C3180,263.092 3197.909,281 3220,281 L3280
    <path d="M3280,181 L3220,281 C3287.129,281 3344.502,239.61 3368.25,181 L3280,181 z" fill="#00AA00"/>
    <path d="M3340,81 C3272.871,81 3215.498,122.39 3191.75,181 L3280,181 L3340,81 z" fill="#FF5555"/>
    <path d="M3340,81 L3280,181 L3368.25,181 C3375.763,162.458 3380,142.237 3380,121 C3380,98.909 3362.091,
    <path d="M3192.938,56 C3191,60.763 3190.002,65.857 3190,71 C3190,76.523 3194.477,81 3200,81 L3215,56 L3
    <path d="M3215,56 L3200,81 C3216.782,81 3231.126,70.653 3237.063,56 L3215,56 z" fill="#00AA00"/>
    <path d="M3230,31 C3213.218,31 3198.874,41.347 3192.937,56 L3215,56 L3230,31 z" fill="#FF5555"/>
    <path d="M3230,31 L3215,56 L3237.063,56 C3239.001,51.236 3239.998,46.143 3240,41 C3240,35.477 3235.523,
    <path d="M3322.938,306 C3321,310.763 3320.002,315.857 3320,321 C3320,326.523 3324.477,331 3330,331 L334
    <path d="M3345,306 L3330,331 C3346.782,331 3361.126,320.653 3367.063,306 L3345,306 z" fill="#00AA00"/>
    <path d="M3360,281 C3343.218,281 3328.874,291.347 3322.937,306 L3345,306 L3360,281 z" fill="#FF5555"/>
    <path d="M3360,281 L3345,306 L3367.063,306 C3369.001,301.236 3369.998,296.143 3370,291 C3370,285.477 33
    <path d="M3340,81 C3272.871,81 3215.498,122.39 3191.75,181 C3184.237,199.542 3180,219.763 3180,241 C318
  </g>
  <g>
    <g>
      <path d="M3200,364 L3360,364 C3382.091,364 3400,381.909 3400,404 L3400,684 C3400,706.091 3382.091,724
      <path d="M3200,364 L3360,364 C3382.091,364 3400,381.909 3400,404 L3400,684 C3400,706.091 3382.091,724
    </g>
    <path d="M3200,384 L3360,384 C3371.046,384 3380,392.954 3380,404 L3380,684 C3380,695.046 3371.046,704 3
    <path d="M3340,444 C3251.634,444 3180,515.635 3180,604 C3180,626.092 3197.909,644 3220,644 C3308.366,64
```

```
    <path d="M3191.75,544 C3184.237,562.542 3180,582.764 3180,604 C3180,626.092 3197.909,644 3220,644 L3280
    <path d="M3280,544 L3220,644 C3287.129,644 3344.502,602.61 3368.25,544 L3280,544 z" fill="#00AA00"/>
    <path d="M3340,444 C3272.871,444 3215.498,485.39 3191.75,544 L3280,544 L3340,444 z" fill="#FF5555"/>
    <path d="M3340,444 L3280,544 L3368.25,544 C3375.763,525.458 3380,505.237 3380,484 C3380,461.909 3362.09
    <path d="M3192.938,419 C3191,423.763 3190.002,428.857 3190,434 C3190,439.523 3194.477,444 3200,444 L321
    <path d="M3215,419 L3200,444 C3216.782,444 3231.126,433.653 3237.063,419 L3215,419 z" fill="#00AA00"/>
    <path d="M3230,394 C3213.218,394 3198.874,404.347 3192.937,419 L3215,419 L3230,394 z" fill="#FF5555"/>
    <path d="M3230,394 L3215,419 L3237.063,419 C3239.001,414.236 3239.998,409.143 3240,404 C3240,398.477 32
    <path d="M3322.938,669 C3321,673.763 3320.002,678.857 3320,684 C3320,689.523 3324.477,694 3330,694 L334
    <path d="M3345,669 L3330,694 C3346.782,694 3361.126,683.653 3367.063,669 L3345,669 z" fill="#00AA00"/>
    <path d="M3360,644 C3343.218,644 3328.874,654.347 3322.937,669 L3345,669 L3360,644 z" fill="#FF5555"/>
    <path d="M3360,644 L3345,669 L3367.063,669 C3369.001,664.236 3369.998,659.143 3370,654 C3370,648.477 33
    <path d="M3340,444 C3272.871,444 3215.498,485.39 3191.75,544 C3184.237,562.542 3180,582.763 3180,604 C3
  </g>
  <g>
    <g>
      <path d="M3200,727 L3360,727 C3382.091,727 3400,744.909 3400,767 L3400,1047 C3400,1069.091 3382.091,1
      <path d="M3200,727 L3360,727 C3382.091,727 3400,744.909 3400,767 L3400,1047 C3400,1069.091 3382.091,1
    </g>
    <path d="M3200,747 L3360,747 C3371.046,747 3380,755.954 3380,767 L3380,1047 C3380,1058.046 3371.046,106
    <path d="M3340,807 C3251.634,807 3180,878.635 3180,967 C3180,989.092 3197.909,1007 3220,1007 C3308.366,
    <path d="M3191.75,907 C3184.237,925.542 3180,945.764 3180,967 C3180,989.092 3197.909,1007 3220,1007 L32
    <path d="M3280,907 L3220,1007 C3287.129,1007 3344.502,965.61 3368.25,907 L3280,907 z" fill="#00AA00"/>
    <path d="M3340,807 C3272.871,807 3215.498,848.39 3191.75,907 L3280,907 L3340,807 z" fill="#FF5555"/>
    <path d="M3340,807 L3280,907 L3368.25,907 C3375.763,888.458 3380,868.237 3380,847 C3380,824.909 3362.09
    <path d="M3192.938,782 C3191,786.763 3190.002,791.857 3190,797 C3190,802.523 3194.477,807 3200,807 L321
    <path d="M3215,782 L3200,807 C3216.782,807 3231.126,796.653 3237.063,782 L3215,782 z" fill="#00AA00"/>
    <path d="M3230,757 C3213.218,757 3198.874,767.347 3192.937,782 L3215,782 L3230,757 z" fill="#FF5555"/>
    <path d="M3230,757 L3215,782 L3237.063,782 C3239.001,777.236 3239.998,772.143 3240,767 C3240,761.477 32
    <path d="M3322.938,1032 C3321,1036.763 3320.002,1041.857 3320,1047 C3320,1052.523 3324.477,1057 3330,10
    <path d="M3345,1032 L3330,1057 C3346.782,1057 3361.126,1046.653 3367.063,1032 L3345,1032 z" fill="#00AA
    <path d="M3360,1007 C3343.218,1007 3328.874,1017.347 3322.937,1032 L3345,1032 L3360,1007 z" fill="#FF55
    <path d="M3360,1007 L3345,1032 L3367.063,1032 C3369.001,1027.236 3369.998,1022.143 3370,1017 C3370,1011
    <path d="M3340,807 C3272.871,807 3215.498,848.39 3191.75,907 C3184.237,925.542 3180,945.763 3180,967 C3
  </g>
  <g>
    <g>
      <path d="M3200,1090 L3360,1090 C3382.091,1090 3400,1107.909 3400,1130 L3400,1410 C3400,1432.091 3382.
      <path d="M3200,1090 L3360,1090 C3382.091,1090 3400,1107.909 3400,1130 L3400,1410 C3400,1432.091 3382.
    </g>
    <path d="M3200,1110 L3360,1110 C3371.046,1110 3380,1118.954 3380,1130 L3380,1410 C3380,1421.046 3371.04
    <path d="M3340,1170 C3251.634,1170 3180,1241.635 3180,1330 C3180,1352.092 3197.909,1370 3220,1370 C3308
    <path d="M3191.75,1270 C3184.237,1288.542 3180,1308.764 3180,1330 C3180,1352.092 3197.909,1370 3220,137
    <path d="M3280,1270 L3220,1370 C3287.129,1370 3344.502,1328.61 3368.25,1270 L3280,1270 z" fill="#00AA00
    <path d="M3340,1170 C3272.871,1170 3215.498,1211.39 3191.75,1270 L3280,1270 L3340,1170 z" fill="#FF5555
    <path d="M3340,1170 L3280,1270 L3368.25,1270 C3375.763,1251.458 3380,1231.237 3380,1210 C3380,1187.909
    <path d="M3192.938,1145 C3191,1149.763 3190.002,1154.857 3190,1160 C3190,1165.523 3194.477,1170 3200,11
    <path d="M3215,1145 L3200,1170 C3216.782,1170 3231.126,1159.653 3237.063,1145 L3215,1145 z" fill="#00AA
    <path d="M3230,1120 C3213.218,1120 3198.874,1130.347 3192.937,1145 L3215,1145 L3230,1120 z" fill="#FF55
    <path d="M3230,1120 L3215,1145 L3237.063,1145 C3239.001,1140.236 3239.998,1135.143 3240,1130 C3240,1124
    <path d="M3322.938,1395 C3321,1399.763 3320.002,1404.857 3320,1410 C3320,1415.523 3324.477,1420 3330,14
    <path d="M3345,1395 L3330,1420 C3346.782,1420 3361.126,1409.653 3367.063,1395 L3345,1395 z" fill="#00AA
    <path d="M3360,1370 C3343.218,1370 3328.874,1380.347 3322.937,1395 L3345,1395 L3360,1370 z" fill="#FF55
    <path d="M3360,1370 L3345,1395 L3367.063,1395 C3369.001,1390.236 3369.998,1385.143 3370,1380 C3370,1374
    <path d="M3340,1170 C3272.871,1170 3215.498,1211.39 3191.75,1270 C3184.237,1288.542 3180,1308.763 3180,
  </g>
  <g>
    <g>
      <path d="M3443,1 L3603,1 C3625.091,1 3643,18.909 3643,41 L3643,321 C3643,343.091 3625.091,361 3603,36
      <path d="M3443,1 L3603,1 C3625.091,1 3643,18.909 3643,41 L3643,321 C3643,343.091 3625.091,361 3603,36
    </g>
    <path d="M3443,21 L3603,21 C3614.046,21 3623,29.954 3623,41 L3623,321 C3623,332.046 3614.046,341 3603,3
    <path d="M3583,81 C3494.636,81 3423,152.636 3423,241 C3423,263.092 3440.908,281 3463,281 C3551.364,281
    <path d="M3482.6,31 L3463,71 L3463,81 L3487,81 L3487,91 L3497,91 L3497,81 L3503,81 L3503,71 L3497,71 L3
    <path d="M3437,51 L3437,61 L3427,61 L3427,71 L3437,71 L3437,81 L3447,81 L3447,71 L3457,71 L3457,61 L344
    <path d="M3563.4,331 L3583,291 L3583,281 L3559,281 L3559,271 L3549,271 L3549,281 L3543,281 L3543,291 L3
    <path d="M3609,311 L3609,301 L3619,301 L3619,291 L3609,291 L3609,281 L3599,281 L3599,291 L3589,291 L358
    <g>
      <g>
        <path d="M3485.86,130.995 L3525.86,130.995 C3531.383,130.995 3534.705,135.038 3533.28,140.025 L3512
        <path d="M3485.86,130.995 L3525.86,130.995 C3531.383,130.995 3534.705,135.038 3533.28,140.025 L3512
      </g>
      <path d="M3493.003,140.995 L3513.003,140.995 C3518.526,140.995 3521.848,145.038 3520.423,150.025 L350
    </g>
    <g>
      <g>
        <path d="M3525.86,100.995 L3565.86,100.995 C3571.383,100.995 3574.705,105.038 3573.28,110.025 L3552
        <path d="M3525.86,100.995 L3565.86,100.995 C3571.383,100.995 3574.705,105.038 3573.28,110.025 L3552
      </g>
      <path d="M3533.003,110.995 L3553.003,110.995 C3558.526,110.995 3561.848,115.038 3560.423,120.025 L354
    </g>
```

```
      <g>
        <g>
          <path d="M3515.86,160.995 L3555.86,160.995 C3561.383,160.995 3564.705,165.038 3563.28,170.026 L3542
          <path d="M3515.86,160.995 L3555.86,160.995 C3561.383,160.995 3564.705,165.038 3563.28,170.026 L3542
        </g>
        <path d="M3523.003,170.995 L3543.003,170.995 C3548.526,170.995 3551.848,175.038 3550.423,180.025 L353
      </g>
      <g>
        <g>
          <path d="M3555.86,130.995 L3595.86,130.995 C3601.383,130.995 3604.705,135.038 3603.28,140.025 L3582
          <path d="M3555.86,130.995 L3595.86,130.995 C3601.383,130.995 3604.705,135.038 3603.28,140.025 L3582
        </g>
        <path d="M3563.003,140.995 L3583.003,140.995 C3588.526,140.995 3591.848,145.038 3590.423,150.025 L357
      </g>
    </g>
    <g>
      <g>
        <path d="M3443,364 L3603,364 C3625.091,364 3643,381.909 3643,404 L3643,684 C3643,706.091 3625.091,724
        <path d="M3443,364 L3603,364 C3625.091,364 3643,381.909 3643,404 L3643,684 C3643,706.091 3625.091,724
      </g>
      <path d="M3443,384 L3603,384 C3614.046,384 3623,392.954 3623,404 L3623,684 C3623,695.046 3614.046,704 3
      <path d="M3583,444 C3494.636,444 3423,515.636 3423,604 C3423,626.092 3440.908,644 3463,644 C3551.364,64
      <path d="M3482.6,394 L3463,434 L3463,444 L3487,444 L3487,454 L3497,454 L3497,444 L3503,444 L3503,434 L3
      <path d="M3437,414 L3437,424 L3427,424 L3427,434 L3437,434 L3437,444 L3447,444 L3447,434 L3457,434 L345
      <path d="M3563.4,694 L3583,654 L3583,644 L3559,644 L3559,634 L3549,634 L3549,644 L3543,644 L3543,654 L3
      <path d="M3609,674 L3609,664 L3619,664 L3619,654 L3609,654 L3609,644 L3599,644 L3599,654 L3589,654 L358
      <g>
        <g>
          <path d="M3485.86,493.995 L3525.86,493.995 C3531.383,493.995 3534.705,498.038 3533.28,503.025 L3512
          <path d="M3485.86,493.995 L3525.86,493.995 C3531.383,493.995 3534.705,498.038 3533.28,503.025 L3512
        </g>
        <path d="M3493.003,503.995 L3513.003,503.995 C3518.526,503.995 3521.848,508.038 3520.423,513.025 L350
      </g>
      <g>
        <g>
          <path d="M3525.86,463.995 L3565.86,463.995 C3571.383,463.995 3574.705,468.038 3573.28,473.025 L3552
          <path d="M3525.86,463.995 L3565.86,463.995 C3571.383,463.995 3574.705,468.038 3573.28,473.025 L3552
        </g>
        <path d="M3533.003,473.995 L3553.003,473.995 C3558.526,473.995 3561.848,478.038 3560.423,483.025 L354
      </g>
      <g>
        <g>
          <path d="M3515.86,523.995 L3555.86,523.995 C3561.383,523.995 3564.705,528.038 3563.28,533.026 L3542
          <path d="M3515.86,523.995 L3555.86,523.995 C3561.383,523.995 3564.705,528.038 3563.28,533.026 L3542
        </g>
        <path d="M3523.003,533.995 L3543.003,533.995 C3548.526,533.995 3551.848,538.038 3550.423,543.025 L353
      </g>
      <g>
        <g>
          <path d="M3555.86,493.995 L3595.86,493.995 C3601.383,493.995 3604.705,498.038 3603.28,503.025 L3582
          <path d="M3555.86,493.995 L3595.86,493.995 C3601.383,493.995 3604.705,498.038 3603.28,503.025 L3582
        </g>
        <path d="M3563.003,503.995 L3583.003,503.995 C3588.526,503.995 3591.848,508.038 3590.423,513.025 L357
      </g>
    </g>
    <g>
      <g>
        <path d="M3443,727 L3603,727 C3625.091,727 3643,744.909 3643,767 L3643,1047 C3643,1069.091 3625.091,1
        <path d="M3443,727 L3603,727 C3625.091,727 3643,744.909 3643,767 L3643,1047 C3643,1069.091 3625.091,1
      </g>
      <path d="M3443,747 L3603,747 C3614.046,747 3623,755.954 3623,767 L3623,1047 C3623,1058.046 3614.046,106
      <path d="M3583,807 C3494.636,807 3423,878.636 3423,967 C3423,989.092 3440.908,1007 3463,1007 C3551.364,
      <path d="M3482.6,757 L3463,797 L3463,807 L3487,807 L3487,817 L3497,817 L3497,807 L3503,807 L3503,797 L3
      <path d="M3437,777 L3437,787 L3427,787 L3427,797 L3437,797 L3437,807 L3447,807 L3447,797 L3457,797 L345
      <path d="M3563.4,1057 L3583,1017 L3583,1007 L3559,1007 L3559,997 L3549,997 L3549,1007 L3543,1007 L3543,
      <path d="M3609,1037 L3609,1027 L3619,1027 L3619,1017 L3609,1017 L3609,1007 L3599,1007 L3599,1017 L3589,
      <g>
        <g>
          <path d="M3485.86,856.995 L3525.86,856.995 C3531.383,856.995 3534.705,861.038 3533.28,866.025 L3512
          <path d="M3485.86,856.995 L3525.86,856.995 C3531.383,856.995 3534.705,861.038 3533.28,866.025 L3512
        </g>
        <path d="M3493.003,866.995 L3513.003,866.995 C3518.526,866.995 3521.848,871.038 3520.423,876.025 L350
      </g>
      <g>
        <g>
          <path d="M3525.86,826.995 L3565.86,826.995 C3571.383,826.995 3574.705,831.038 3573.28,836.025 L3552
          <path d="M3525.86,826.995 L3565.86,826.995 C3571.383,826.995 3574.705,831.038 3573.28,836.025 L3552
        </g>
        <path d="M3533.003,836.995 L3553.003,836.995 C3558.526,836.995 3561.848,841.038 3560.423,846.025 L354
      </g>
      <g>
```

```
        <g>
          <path d="M3515.86,886.995 L3555.86,886.995 C3561.383,886.995 3564.705,891.038 3563.28,896.026 L3542
          <path d="M3515.86,886.995 L3555.86,886.995 C3561.383,886.995 3564.705,891.038 3563.28,896.026 L3542
        </g>
        <path d="M3523.003,896.995 L3543.003,896.995 C3548.526,896.995 3551.848,901.038 3550.423,906.025 L353
      </g>
      <g>
        <g>
          <path d="M3555.86,856.995 L3595.86,856.995 C3601.383,856.995 3604.705,861.038 3603.28,866.025 L3582
          <path d="M3555.86,856.995 L3595.86,856.995 C3601.383,856.995 3604.705,861.038 3603.28,866.025 L3582
        </g>
        <path d="M3563.003,866.995 L3583.003,866.995 C3588.526,866.995 3591.848,871.038 3590.423,876.025 L357
      </g>
    </g>
    <g>
      <g>
        <path d="M3443,1090 L3603,1090 C3625.091,1090 3643,1107.909 3643,1130 L3643,1410 C3643,1432.091 3625.
        <path d="M3443,1090 L3603,1090 C3625.091,1090 3643,1107.909 3643,1130 L3643,1410 C3643,1432.091 3625.
      </g>
      <path d="M3443,1110 L3603,1110 C3614.046,1110 3623,1118.954 3623,1130 L3623,1410 C3623,1421.046 3614.04
      <path d="M3583,1170 C3494.636,1170 3423,1241.636 3423,1330 C3423,1352.092 3440.908,1370 3463,1370 C3551
      <path d="M3482.6,1120 L3463,1160 L3463,1170 L3487,1170 L3487,1180 L3497,1180 L3497,1170 L3503,1170 L350
      <path d="M3437,1140 L3437,1150 L3427,1150 L3427,1160 L3437,1160 L3437,1170 L3447,1170 L3447,1160 L3457,
      <path d="M3563.4,1420 L3583,1380 L3583,1370 L3559,1370 L3559,1360 L3549,1360 L3549,1370 L3543,1370 L354
      <path d="M3609,1400 L3609,1390 L3619,1390 L3619,1380 L3609,1380 L3609,1370 L3599,1370 L3599,1380 L3589,
      <g>
        <g>
          <path d="M3485.86,1219.995 L3525.86,1219.995 C3531.383,1219.995 3534.705,1224.038 3533.28,1229.025
          <path d="M3485.86,1219.995 L3525.86,1219.995 C3531.383,1219.995 3534.705,1224.038 3533.28,1229.025
        </g>
        <path d="M3493.003,1229.995 L3513.003,1229.995 C3518.526,1229.995 3521.848,1234.038 3520.423,1239.025
      </g>
      <g>
        <g>
          <path d="M3525.86,1189.995 L3565.86,1189.995 C3571.383,1189.995 3574.705,1194.038 3573.28,1199.025
          <path d="M3525.86,1189.995 L3565.86,1189.995 C3571.383,1189.995 3574.705,1194.038 3573.28,1199.025
        </g>
        <path d="M3533.003,1199.995 L3553.003,1199.995 C3558.526,1199.995 3561.848,1204.038 3560.423,1209.025
      </g>
      <g>
        <g>
          <path d="M3515.86,1249.995 L3555.86,1249.995 C3561.383,1249.995 3564.705,1254.038 3563.28,1259.026
          <path d="M3515.86,1249.995 L3555.86,1249.995 C3561.383,1249.995 3564.705,1254.038 3563.28,1259.026
        </g>
        <path d="M3523.003,1259.995 L3543.003,1259.995 C3548.526,1259.995 3551.848,1264.038 3550.423,1269.025
      </g>
      <g>
        <g>
          <path d="M3555.86,1219.995 L3595.86,1219.995 C3601.383,1219.995 3604.705,1224.038 3603.28,1229.025
          <path d="M3555.86,1219.995 L3595.86,1219.995 C3601.383,1219.995 3604.705,1224.038 3603.28,1229.025
        </g>
        <path d="M3563.003,1229.995 L3583.003,1229.995 C3588.526,1229.995 3591.848,1234.038 3590.423,1239.025
      </g>
    </g>
  </g>
</svg>
```

cp8.zip (52025 bytes)

```
(binary file)
```

inside of cp8.zip: UNO_deck.svg (170457 bytes)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0" y="0"
  <g id="Layer_1">
    <g>
      <g>
        <path d="M284,1 L444,1 C466.091,1 484,18.909 484,41 L484,321 C484,343.091 466.091,361 444,361 L284,36
        <path d="M284,1 L444,1 C466.091,1 484,18.909 484,41 L484,321 C484,343.091 466.091,361 444,361 L284,36
      </g>
      <path d="M284,21 L444,21 C455.046,21 464,29.954 464,41 L464,321 C464,332.046 455.046,341 444,341 L284,3
      <path d="M424,81 C335.636,81 264,152.636 264,241 C264,263.092 281.908,281 304,281 C392.364,281 464,209.
      <path d="M356,121 L336,141 L336,165 L356,145 L356,241 L376,241 L376,121 z" fill="#FF5555"/>
      <path d="M284,31 L274,41 L274,53 L284,43 L284,91 L294,91 L294,31 z M444,331 L454,321 L454,309 L444,319
    </g>
    <g>
      <g>
```

```
      <path d="M41,1 L201,1 C223.091,1 241,18.909 241,41 L241,321 C241,343.091 223.091,361 201,361 L41,361
      <path d="M41,1 L201,1 C223.091,1 241,18.909 241,41 L241,321 C241,343.091 223.091,361 201,361 L41,361
    </g>
    <path d="M41,21 L201,21 C212.046,21 221,29.954 221,41 L221,321 C221,332.046 212.046,341 201,341 L41,341
    <path d="M181,81 C92.636,81 21,152.636 21,241 C21,263.092 38.908,281 61,281 C149.364,281 221,209.368 22
    <path d="M121,121 C98.908,121 81,138.908 81,161 L81,201 C81,223.092 98.908,241 121,241 C143.092,241 161
    <path d="M51,31 C39.954,31 31,39.954 31,51 L31,71 C31,82.046 39.954,91 51,91 C62.046,91 71,82.046 71,71
  </g>
  <g>
    <g>
      <path d="M1742,1 L1902,1 C1924.091,1 1942,18.909 1942,41 L1942,321 C1942,343.091 1924.091,361 1902,36
      <path d="M1742,1 L1902,1 C1924.091,1 1942,18.909 1942,41 L1942,321 C1942,343.091 1924.091,361 1902,36
    </g>
    <path d="M1742,21 L1902,21 C1913.046,21 1922,29.954 1922,41 L1922,321 C1922,332.046 1913.046,341 1902,3
    <path d="M1882,81 C1793.636,81 1722,152.636 1722,241 C1722,263.092 1739.908,281 1762,281 C1850.364,281
    <path d="M1788,121 L1788,161 L1808,161 L1808,141 L1848,141 L1808,241 L1828,241 L1868,141 L1868,121 L178
    <path d="M1732,31 L1732,51 L1742,51 L1742,41 L1762,41 L1742,91 L1752,91 L1772,41 L1772,31 L1732,31 z M1
  </g>
  <g>
    <g>
      <path d="M527,1 L687,1 C709.091,1 727,18.909 727,41 L727,321 C727,343.091 709.091,361 687,361 L527,36
      <path d="M527,1 L687,1 C709.091,1 727,18.909 727,41 L727,321 C727,343.091 709.091,361 687,361 L527,36
    </g>
    <path d="M527,21 L687,21 C698.046,21 707,29.954 707,41 L707,321 C707,332.046 698.046,341 687,341 L527,3
    <path d="M667,81 C578.636,81 507,152.636 507,241 C507,263.092 524.908,281 547,281 C635.364,281 707,209.
    <path d="M603,121 C580.84,121 563,138.84 563,161 L563,165 L583,165 L583,161 C583,149.92 591.92,141 603,
    <path d="M537,31 C525.92,31 517,39.92 517,51 L517,53 L527,53 L527,51 C527,45.46 531.46,41 537,41 C542.5
  </g>
  <g>
    <g>
      <path d="M770,1 L930,1 C952.091,1 970,18.909 970,41 L970,321 C970,343.091 952.091,361 930,361 L770,36
      <path d="M770,1 L930,1 C952.091,1 970,18.909 970,41 L970,321 C970,343.091 952.091,361 930,361 L770,36
    </g>
    <path d="M770,21 L930,21 C941.046,21 950,29.954 950,41 L950,321 C950,332.046 941.046,341 930,341 L770,3
    <path d="M910,81 C821.636,81 750,152.636 750,241 C750,263.092 767.908,281 790,281 C878.364,281 950,209.
    <path d="M850,121 C833.431,121 820,134.431 820,151 L840,151 C840,145.477 844.477,141 850,141 C855.523,1
    <path d="M780,31 C771.716,31 765,37.716 765,46 L775,46 C775,43.239 777.239,41 780,41 C782.761,41 785,43
  </g>
  <g>
    <g>
      <path d="M1013,1 L1173,1 C1195.091,1 1213,18.909 1213,41 L1213,321 C1213,343.091 1195.091,361 1173,36
      <path d="M1013,1 L1173,1 C1195.091,1 1213,18.909 1213,41 L1213,321 C1213,343.091 1195.091,361 1173,36
    </g>
    <path d="M1013,21 L1173,21 C1184.046,21 1193,29.954 1193,41 L1193,321 C1193,332.046 1184.046,341 1173,3
    <path d="M1153,81 C1064.636,81 993,152.636 993,241 C993,263.092 1010.908,281 1033,281 C1121.364,281 119
    <path d="M1088.2,121 L1049,201 L1049,221 L1097,221 L1097,241 L1117,241 L1117,221 L1129,221 L1129,201 L1
    <path d="M1022.2,31 L1002.6,71 L1002.6,81 L1026.6,81 L1026.6,91 L1036.6,91 L1036.6,81 L1042.6,81 L1042.
  </g>
  <g>
    <g>
      <path d="M1256,1 L1416,1 C1438.091,1 1456,18.909 1456,41 L1456,321 C1456,343.091 1438.091,361 1416,36
      <path d="M1256,1 L1416,1 C1438.091,1 1456,18.909 1456,41 L1456,321 C1456,343.091 1438.091,361 1416,36
    </g>
    <path d="M1256,21 L1416,21 C1427.046,21 1436,29.954 1436,41 L1436,321 C1436,332.046 1427.046,341 1416,3
    <path d="M1396,81 C1307.636,81 1236,152.636 1236,241 C1236,263.092 1253.908,281 1276,281 C1364.364,281
    <path d="M1300,121 L1300,181 L1340,181 C1351.08,181 1360,189.92 1360,201 C1360,212.08 1351.08,221 1340,
    <path d="M1246,31 L1246,61 L1266,61 C1271.54,61 1276,65.46 1276,71 C1276,76.54 1271.54,81 1266,81 C1263
  </g>
  <g>
    <g>
      <path d="M1499,1 L1659,1 C1681.091,1 1699,18.909 1699,41 L1699,321 C1699,343.091 1681.091,361 1659,36
      <path d="M1499,1 L1659,1 C1681.091,1 1699,18.909 1699,41 L1699,321 C1699,343.091 1681.091,361 1659,36
    </g>
    <path d="M1499,21 L1659,21 C1670.046,21 1679,29.954 1679,41 L1679,321 C1679,332.046 1670.046,341 1659,3
    <path d="M1639,81 C1550.636,81 1479,152.636 1479,241 C1479,263.092 1496.908,281 1519,281 C1607.364,281
    <path d="M1579,121 C1556.908,121 1539,138.908 1539,161 L1539,201 C1539,223.092 1556.908,241 1579,241 C1
    <path d="M1509,31 C1497.954,31 1489,39.954 1489,51 L1489,71 C1488.996,80.053 1495.074,87.979 1503.818,9
  </g>
  <g>
    <g>
      <path d="M1985,1 L2145,1 C2167.091,1 2185,18.909 2185,41 L2185,321 C2185,343.091 2167.091,361 2145,36
      <path d="M1985,1 L2145,1 C2167.091,1 2185,18.909 2185,41 L2185,321 C2185,343.091 2167.091,361 2145,36
    </g>
    <path d="M1985,21 L2145,21 C2156.046,21 2165,29.954 2165,41 L2165,321 C2165,332.046 2156.046,341 2145,3
    <path d="M2125,81 C2036.636,81 1965,152.636 1965,241 C1965,263.092 1982.908,281 2005,281 C2093.364,281
    <path d="M2065,121 C2048.431,121 2035,134.431 2035,151 C2035,157.796 2037.32,163.972 2041.124,169 C2031
    <path d="M1995,31 C1986.716,31 1980,37.716 1980,46 C1980,49.4 1981.16,52.484 1983.064,55 C1977.987,58.7
  </g>
  <g>
    <g>
      <path d="M2228,1 L2388,1 C2410.091,1 2428,18.909 2428,41 L2428,321 C2428,343.091 2410.091,361 2388,36
```

```
      <path d="M2228,1 L2388,1 C2410.091,1 2428,18.909 2428,41 L2428,321 C2428,343.091 2410.091,361 2388,36
    </g>
    <path d="M2228,21 L2388,21 C2399.046,21 2408,29.954 2408,41 L2408,321 C2408,332.046 2399.046,341 2388,3
    <path d="M2368,81 C2279.636,81 2208,152.636 2208,241 C2208,263.092 2225.908,281 2248,281 C2336.364,281
    <path d="M2308,121 C2285.908,121 2268,138.908 2268,161 C2268,183.092 2285.908,201 2308,201 C2315.268,20
    <path d="M2238,91 C2249.046,91 2258,82.046 2258,71 L2258,51 C2258.004,41.947 2251.926,34.02 2243.183,31
  </g>
  <g>
    <g>
      <path d="M2471,1 L2631,1 C2653.091,1 2671,18.909 2671,41 L2671,321 C2671,343.091 2653.091,361 2631,36
      <path d="M2471,1 L2631,1 C2653.091,1 2671,18.909 2671,41 L2671,321 C2671,343.091 2653.091,361 2631,36
    </g>
    <path d="M2471,21 L2631,21 C2642.046,21 2651,29.954 2651,41 L2651,321 C2651,332.046 2642.046,341 2631,3
    <path d="M2611,81 C2522.636,81 2451,152.636 2451,241 C2451,263.092 2468.908,281 2491,281 C2579.364,281
    <path d="M2551.124,121 C2535.194,120.951 2519.901,127.248 2508.624,138.5 C2485.164,161.9 2485.1,199.912
    <path d="M2491.064,31 C2483.1,30.974 2475.453,34.122 2469.816,39.748 C2458.079,51.447 2458.05,70.445 24
  </g>
  <g>
    <g>
      <path d="M2714,1 L2874,1 C2896.091,1 2914,18.909 2914,41 L2914,321 C2914,343.091 2896.091,361 2874,36
      <path d="M2714,1 L2874,1 C2896.091,1 2914,18.909 2914,41 L2914,321 C2914,343.091 2896.091,361 2874,36
    </g>
    <path d="M2714,21 L2874,21 C2885.046,21 2894,29.954 2894,41 L2894,321 C2894,332.046 2885.046,341 2874,3
    <path d="M2854,81 C2765.636,81 2694,152.636 2694,241 C2694,263.092 2711.908,281 2734,281 C2822.364,281
    <path d="M2794,116 L2804,126 L2764,166 C2754,176 2754,196 2764,206 L2784,186 L2824,146 L2834,156 L2834,
    <path d="M2794,246 L2784,236 L2824,196 C2834,186 2834,166 2824,156 L2804,176 L2764,216 L2754,206 L2754,
    <path d="M2724,31 L2729,36 L2709,56 C2704,61 2704,71 2709,76 L2719,66 L2739,46 L2744,51 L2744,31 z" fil
    <path d="M2724,96 L2719,91 L2739,71 C2744,66 2744,56 2739,51 L2729,61 L2709,81 L2704,76 L2704,96 z M286
    <path d="M2864,331 L2859,326 L2879,306 C2884,301 2884,291 2879,286 L2869,296 L2849,316 L2844,311 L2844,
  </g>
  <g>
    <g>
      <path d="M3443,1453 L3603,1453 C3625.091,1453 3643,1470.909 3643,1493 L3643,1773 C3643,1795.091 3625.
      <path d="M3443,1453 L3603,1453 C3625.091,1453 3643,1470.909 3643,1493 L3643,1773 C3643,1795.091 3625.
    </g>
    <path d="M3443,1473 L3603,1473 C3614.046,1473 3623,1481.954 3623,1493 L3623,1773 C3623,1784.046 3614.04
    <path d="M3583,1533 C3494.636,1533 3423,1604.636 3423,1693 C3423,1715.092 3440.908,1733 3463,1733 C3551
    <path d="M3482.6,1483 L3463,1523 L3463,1533 L3487,1533 L3487,1543 L3497,1543 L3497,1533 L3503,1533 L350
    <path d="M3437,1503 L3437,1513 L3427,1513 L3427,1523 L3437,1523 L3437,1533 L3447,1533 L3447,1523 L3457,
    <path d="M3563.4,1783 L3583,1743 L3583,1733 L3559,1733 L3559,1723 L3549,1723 L3549,1733 L3543,1733 L354
    <path d="M3609,1763 L3609,1753 L3619,1753 L3619,1743 L3609,1743 L3609,1733 L3599,1733 L3599,1743 L3589,
    <g>
      <g>
        <path d="M3485.86,1582.995 L3525.86,1582.995 C3531.383,1582.995 3534.705,1587.038 3533.28,1592.025
        <path d="M3485.86,1582.995 L3525.86,1582.995 C3531.383,1582.995 3534.705,1587.038 3533.28,1592.025
      </g>
      <path d="M3493.003,1592.995 L3513.003,1592.995 C3518.526,1592.995 3521.848,1597.038 3520.423,1602.025
    </g>
    <g>
      <g>
        <path d="M3525.86,1552.995 L3565.86,1552.995 C3571.383,1552.995 3574.705,1557.038 3573.28,1562.025
        <path d="M3525.86,1552.995 L3565.86,1552.995 C3571.383,1552.995 3574.705,1557.038 3573.28,1562.025
      </g>
      <path d="M3533.003,1562.995 L3553.003,1562.995 C3558.526,1562.995 3561.848,1567.038 3560.423,1572.025
    </g>
    <g>
      <g>
        <path d="M3515.86,1612.995 L3555.86,1612.995 C3561.383,1612.995 3564.705,1617.038 3563.28,1622.026
        <path d="M3515.86,1612.995 L3555.86,1612.995 C3561.383,1612.995 3564.705,1617.038 3563.28,1622.026
      </g>
      <path d="M3523.003,1622.995 L3543.003,1622.995 C3548.526,1622.995 3551.848,1627.038 3550.423,1632.025
    </g>
    <g>
      <g>
        <path d="M3555.86,1582.995 L3595.86,1582.995 C3601.383,1582.995 3604.705,1587.038 3603.28,1592.025
        <path d="M3555.86,1582.995 L3595.86,1582.995 C3601.383,1582.995 3604.705,1587.038 3603.28,1592.025
      </g>
      <path d="M3563.003,1592.995 L3583.003,1592.995 C3588.526,1592.995 3591.848,1597.038 3590.423,1602.025
    </g>
  </g>
  <g>
    <g>
      <path d="M284,364 L444,364 C466.091,364 484,381.909 484,404 L484,684 C484,706.091 466.091,724 444,724
      <path d="M284,364 L444,364 C466.091,364 484,381.909 484,404 L484,684 C484,706.091 466.091,724 444,724
    </g>
    <path d="M284,384 L444,384 C455.046,384 464,392.954 464,404 L464,684 C464,695.046 455.046,704 444,704 L
    <path d="M424,444 C335.636,444 264,515.636 264,604 C264,626.092 281.908,644 304,644 C392.364,644 464,57
    <path d="M356,484 L336,504 L336,528 L356,508 L356,604 L376,604 L376,484 z" fill="#FFAA00"/>
    <path d="M284,394 L274,404 L274,416 L284,406 L284,454 L294,454 L294,394 z M444,694 L454,684 L454,672 L4
  </g>
  <g>
```

```
    <g>
      <path d="M41,364 L201,364 C223.091,364 241,381.909 241,404 L241,684 C241,706.091 223.091,724 201,724
      <path d="M41,364 L201,364 C223.091,364 241,381.909 241,404 L241,684 C241,706.091 223.091,724 201,724
    </g>
    <path d="M41,384 L201,384 C212.046,384 221,392.954 221,404 L221,684 C221,695.046 212.046,704 201,704 L4
    <path d="M181,444 C92.636,444 21,515.636 21,604 C21,626.092 38.908,644 61,644 C149.364,644 221,572.368
    <path d="M121,484 C98.908,484 81,501.908 81,524 L81,564 C81,586.092 98.908,604 121,604 C143.092,604 161
    <path d="M51,394 C39.954,394 31,402.954 31,414 L31,434 C31,445.046 39.954,454 51,454 C62.046,454 71,445
  </g>
  <g>
    <g>
      <path d="M1742,364 L1902,364 C1924.091,364 1942,381.909 1942,404 L1942,684 C1942,706.091 1924.091,724
      <path d="M1742,364 L1902,364 C1924.091,364 1942,381.909 1942,404 L1942,684 C1942,706.091 1924.091,724
    </g>
    <path d="M1742,384 L1902,384 C1913.046,384 1922,392.954 1922,404 L1922,684 C1922,695.046 1913.046,704 1
    <path d="M1882,444 C1793.636,444 1722,515.636 1722,604 C1722,626.092 1739.908,644 1762,644 C1850.364,64
    <path d="M1788,484 L1788,524 L1808,524 L1808,504 L1848,504 L1808,604 L1828,604 L1868,504 L1868,484 L178
    <path d="M1732,394 L1732,414 L1742,414 L1742,404 L1762,404 L1742,454 L1752,454 L1772,404 L1772,394 L173
  </g>
  <g>
    <g>
      <path d="M527,364 L687,364 C709.091,364 727,381.909 727,404 L727,684 C727,706.091 709.091,724 687,724
      <path d="M527,364 L687,364 C709.091,364 727,381.909 727,404 L727,684 C727,706.091 709.091,724 687,724
    </g>
    <path d="M527,384 L687,384 C698.046,384 707,392.954 707,404 L707,684 C707,695.046 698.046,704 687,704 L
    <path d="M667,444 C578.636,444 507,515.636 507,604 C507,626.092 524.908,644 547,644 C635.364,644 707,57
    <path d="M603,484 C580.84,484 563,501.84 563,524 L563,528 L583,528 L583,524 C583,512.92 591.92,504 603,
    <path d="M537,394 C525.92,394 517,402.92 517,414 L517,416 L527,416 L527,414 C527,408.46 531.46,404 537,
  </g>
  <g>
    <g>
      <path d="M770,364 L930,364 C952.091,364 970,381.909 970,404 L970,684 C970,706.091 952.091,724 930,724
      <path d="M770,364 L930,364 C952.091,364 970,381.909 970,404 L970,684 C970,706.091 952.091,724 930,724
    </g>
    <path d="M770,384 L930,384 C941.046,384 950,392.954 950,404 L950,684 C950,695.046 941.046,704 930,704 L
    <path d="M910,444 C821.636,444 750,515.636 750,604 C750,626.092 767.908,644 790,644 C878.364,644 950,57
    <path d="M850,484 C833.431,484 820,497.431 820,514 L840,514 C840,508.477 844.477,504 850,504 C855.523,5
    <path d="M780,394 C771.716,394 765,400.716 765,409 L775,409 C775,406.239 777.239,404 780,404 C782.761,4
  </g>
  <g>
    <g>
      <path d="M1013,364 L1173,364 C1195.091,364 1213,381.909 1213,404 L1213,684 C1213,706.091 1195.091,724
      <path d="M1013,364 L1173,364 C1195.091,364 1213,381.909 1213,404 L1213,684 C1213,706.091 1195.091,724
    </g>
    <path d="M1013,384 L1173,384 C1184.046,384 1193,392.954 1193,404 L1193,684 C1193,695.046 1184.046,704 1
    <path d="M1153,444 C1064.636,444 993,515.636 993,604 C993,626.092 1010.908,644 1033,644 C1121.364,644 1
    <path d="M1088.2,484 L1049,564 L1049,584 L1097,584 L1097,604 L1117,604 L1117,584 L1129,584 L1129,564 L1
    <path d="M1022.2,394 L1002.6,434 L1002.6,444 L1026.6,444 L1026.6,454 L1036.6,454 L1036.6,444 L1042.6,44
  </g>
  <g>
    <g>
      <path d="M1256,364 L1416,364 C1438.091,364 1456,381.909 1456,404 L1456,684 C1456,706.091 1438.091,724
      <path d="M1256,364 L1416,364 C1438.091,364 1456,381.909 1456,404 L1456,684 C1456,706.091 1438.091,724
    </g>
    <path d="M1256,384 L1416,384 C1427.046,384 1436,392.954 1436,404 L1436,684 C1436,695.046 1427.046,704 1
    <path d="M1396,444 C1307.636,444 1236,515.636 1236,604 C1236,626.092 1253.908,644 1276,644 C1364.364,64
    <path d="M1300,484 L1300,544 L1340,544 C1351.08,544 1360,552.92 1360,564 C1360,575.08 1351.08,584 1340,
    <path d="M1246,394 L1246,424 L1266,424 C1271.54,424 1276,428.46 1276,434 C1276,439.54 1271.54,444 1266,
  </g>
  <g>
    <g>
      <path d="M1499,364 L1659,364 C1681.091,364 1699,381.909 1699,404 L1699,684 C1699,706.091 1681.091,724
      <path d="M1499,364 L1659,364 C1681.091,364 1699,381.909 1699,404 L1699,684 C1699,706.091 1681.091,724
    </g>
    <path d="M1499,384 L1659,384 C1670.046,384 1679,392.954 1679,404 L1679,684 C1679,695.046 1670.046,704 1
    <path d="M1639,444 C1550.636,444 1479,515.636 1479,604 C1479,626.092 1496.908,644 1519,644 C1607.364,64
    <path d="M1579,484 C1556.908,484 1539,501.908 1539,524 L1539,564 C1539,586.092 1556.908,604 1579,604 C1
    <path d="M1509,394 C1497.954,394 1489,402.954 1489,414 L1489,434 C1488.996,443.053 1495.074,450.98 1503
  </g>
  <g>
    <g>
      <path d="M1985,364 L2145,364 C2167.091,364 2185,381.909 2185,404 L2185,684 C2185,706.091 2167.091,724
      <path d="M1985,364 L2145,364 C2167.091,364 2185,381.909 2185,404 L2185,684 C2185,706.091 2167.091,724
    </g>
    <path d="M1985,384 L2145,384 C2156.046,384 2165,392.954 2165,404 L2165,684 C2165,695.046 2156.046,704 2
    <path d="M2125,444 C2036.636,444 1965,515.636 1965,604 C1965,626.092 1982.908,644 2005,644 C2093.364,64
    <path d="M2065,484 C2048.431,484 2035,497.431 2035,514 C2035,520.796 2037.32,526.972 2041.124,532 C2031
    <path d="M1995,394 C1986.716,394 1980,400.716 1980,409 C1980,412.4 1981.16,415.484 1983.064,418 C1977.9
  </g>
  <g>
    <g>
```

```
      <path d="M2228,364 L2388,364 C2410.091,364 2428,381.909 2428,404 L2428,684 C2428,706.091 2410.091,724
      <path d="M2228,364 L2388,364 C2410.091,364 2428,381.909 2428,404 L2428,684 C2428,706.091 2410.091,724
    </g>
    <path d="M2228,384 L2388,384 C2399.046,384 2408,392.954 2408,404 L2408,684 C2408,695.046 2399.046,704 2
    <path d="M2368,444 C2279.636,444 2208,515.636 2208,604 C2208,626.092 2225.908,644 2248,644 C2336.364,64
    <path d="M2308,484 C2285.908,484 2268,501.908 2268,524 C2268,546.092 2285.908,564 2308,564 C2315.268,56
    <path d="M2238,454 C2249.046,454 2258,445.046 2258,434 L2258,414 C2258.004,404.947 2251.926,397.02 2243
  </g>
  <g>
    <g>
      <path d="M2471,364 L2631,364 C2653.091,364 2671,381.909 2671,404 L2671,684 C2671,706.091 2653.091,724
      <path d="M2471,364 L2631,364 C2653.091,364 2671,381.909 2671,404 L2671,684 C2671,706.091 2653.091,724
    </g>
    <path d="M2471,384 L2631,384 C2642.046,384 2651,392.954 2651,404 L2651,684 C2651,695.046 2642.046,704 2
    <path d="M2611,444 C2522.636,444 2451,515.636 2451,604 C2451,626.092 2468.908,644 2491,644 C2579.364,64
    <path d="M2551.124,484 C2535.194,483.951 2519.901,490.248 2508.624,501.5 C2485.164,524.9 2485.1,562.912
    <path d="M2491.064,394 C2483.1,393.974 2475.453,397.122 2469.816,402.748 C2458.079,414.447 2458.05,433.
  </g>
  <g>
    <g>
      <path d="M2714,364 L2874,364 C2896.091,364 2914,381.909 2914,404 L2914,684 C2914,706.091 2896.091,724
      <path d="M2714,364 L2874,364 C2896.091,364 2914,381.909 2914,404 L2914,684 C2914,706.091 2896.091,724
    </g>
    <path d="M2714,384 L2874,384 C2885.046,384 2894,392.954 2894,404 L2894,684 C2894,695.046 2885.046,704 2
    <path d="M2854,444 C2765.636,444 2694,515.636 2694,604 C2694,626.092 2711.908,644 2734,644 C2822.364,64
    <path d="M2794,479 L2804,489 L2764,529 C2754,539 2754,559 2764,569 L2784,549 L2824,509 L2834,519 L2834,
    <path d="M2794,609 L2784,599 L2824,559 C2834,549 2834,529 2824,519 L2804,539 L2764,579 L2754,569 L2754,
    <path d="M2724,394 L2729,399 L2709,419 C2704,424 2704,434 2709,439 L2719,429 L2739,409 L2744,414 L2744,
    <path d="M2724,459 L2719,454 L2739,434 C2744,429 2744,419 2739,414 L2729,424 L2709,444 L2704,439 L2704,
    <path d="M2864,694 L2859,689 L2879,669 C2884,664 2884,654 2879,649 L2869,659 L2849,679 L2844,674 L2844,
  </g>
  <g>
    <g>
      <path d="M284,727 L444,727 C466.091,727 484,744.909 484,767 L484,1047 C484,1069.091 466.091,1087 444,
      <path d="M284,727 L444,727 C466.091,727 484,744.909 484,767 L484,1047 C484,1069.091 466.091,1087 444,
    </g>
    <path d="M284,747 L444,747 C455.046,747 464,755.954 464,767 L464,1047 C464,1058.046 455.046,1067 444,10
    <path d="M424,807 C335.636,807 264,878.636 264,967 C264,989.092 281.908,1007 304,1007 C392.364,1007 464
    <path d="M356,847 L336,867 L336,891 L356,871 L356,967 L376,967 L376,847 z" fill="#55AA55"/>
    <path d="M284,757 L274,767 L274,779 L284,769 L284,817 L294,817 L294,757 z M444,1057 L454,1047 L454,1035
  </g>
  <g>
    <g>
      <path d="M41,727 L201,727 C223.091,727 241,744.909 241,767 L241,1047 C241,1069.092 223.091,1087 201,1
      <path d="M41,727 L201,727 C223.091,727 241,744.909 241,767 L241,1047 C241,1069.092 223.091,1087 201,1
    </g>
    <path d="M41,747 L201,747 C212.046,747 221,755.955 221,767 L221,1047 C221,1058.046 212.046,1067 201,106
    <path d="M181,807 C92.636,807 21,878.636 21,967 C21,989.092 38.908,1007 61,1007 C149.364,1007 221,935.3
    <path d="M121,847 C98.908,847 81,864.908 81,887 L81,927 C81,949.092 98.908,967 121,967 C143.092,967 161
    <path d="M51,757 C39.954,757 31,765.955 31,777 L31,797 C31,808.046 39.954,817 51,817 C62.046,817 71,808
  </g>
  <g>
    <g>
      <path d="M1742,727 L1902,727 C1924.091,727 1942,744.909 1942,767 L1942,1047 C1942,1069.091 1924.091,1
      <path d="M1742,727 L1902,727 C1924.091,727 1942,744.909 1942,767 L1942,1047 C1942,1069.091 1924.091,1
    </g>
    <path d="M1742,747 L1902,747 C1913.046,747 1922,755.954 1922,767 L1922,1047 C1922,1058.046 1913.046,106
    <path d="M1882,807 C1793.636,807 1722,878.636 1722,967 C1722,989.092 1739.908,1007 1762,1007 C1850.364,
    <path d="M1788,847 L1788,887 L1808,887 L1808,867 L1848,867 L1808,967 L1828,967 L1868,867 L1868,847 L178
    <path d="M1732,757 L1732,777 L1742,777 L1742,767 L1762,767 L1742,817 L1752,817 L1772,767 L1772,757 L173
  </g>
  <g>
    <g>
      <path d="M527,727 L687,727 C709.091,727 727,744.909 727,767 L727,1047 C727,1069.091 709.091,1087 687,
      <path d="M527,727 L687,727 C709.091,727 727,744.909 727,767 L727,1047 C727,1069.091 709.091,1087 687,
    </g>
    <path d="M527,747 L687,747 C698.046,747 707,755.954 707,767 L707,1047 C707,1058.046 698.046,1067 687,10
    <path d="M667,807 C578.636,807 507,878.636 507,967 C507,989.092 524.908,1007 547,1007 C635.364,1007 707
    <path d="M603,847 C580.84,847 563,864.84 563,887 L563,891 L583,891 L583,887 C583,875.92 591.92,867 603,
    <path d="M537,757 C525.92,757 517,765.92 517,777 L517,779 L527,779 L527,777 C527,771.46 531.46,767 537,
  </g>
  <g>
    <g>
      <path d="M770,727 L930,727 C952.091,727 970,744.909 970,767 L970,1047 C970,1069.091 952.091,1087 930,
      <path d="M770,727 L930,727 C952.091,727 970,744.909 970,767 L970,1047 C970,1069.091 952.091,1087 930,
    </g>
    <path d="M770,747 L930,747 C941.046,747 950,755.954 950,767 L950,1047 C950,1058.046 941.046,1067 930,10
    <path d="M910,807 C821.636,807 750,878.636 750,967 C750,989.092 767.908,1007 790,1007 C878.364,1007 950
    <path d="M850,847 C833.431,847 820,860.431 820,877 L840,877 C840,871.477 844.477,867 850,867 C855.523,8
    <path d="M780,757 C771.716,757 765,763.716 765,772 L775,772 C775,769.239 777.239,767 780,767 C782.761,7
  </g>
```

```
    <g>
      <g>
        <path d="M1013,727 L1173,727 C1195.091,727 1213,744.909 1213,767 L1213,1047 C1213,1069.091 1195.091,1
        <path d="M1013,727 L1173,727 C1195.091,727 1213,744.909 1213,767 L1213,1047 C1213,1069.091 1195.091,1
      </g>
      <path d="M1013,747 L1173,747 C1184.046,747 1193,755.954 1193,767 L1193,1047 C1193,1058.046 1184.046,106
      <path d="M1153,807 C1064.636,807 993,878.636 993,967 C993,989.092 1010.908,1007 1033,1007 C1121.364,100
      <path d="M1088.2,847 L1049,927 L1049,947 L1097,947 L1097,967 L1117,967 L1117,947 L1129,947 L1129,927 L1
      <path d="M1022.2,757 L1002.6,797 L1002.6,807 L1026.6,807 L1026.6,817 L1036.6,817 L1036.6,807 L1042.6,80
    </g>
    <g>
      <g>
        <path d="M1256,727 L1416,727 C1438.091,727 1456,744.909 1456,767 L1456,1047 C1456,1069.091 1438.091,1
        <path d="M1256,727 L1416,727 C1438.091,727 1456,744.909 1456,767 L1456,1047 C1456,1069.091 1438.091,1
      </g>
      <path d="M1256,747 L1416,747 C1427.046,747 1436,755.954 1436,767 L1436,1047 C1436,1058.046 1427.046,106
      <path d="M1396,807 C1307.636,807 1236,878.636 1236,967 C1236,989.092 1253.908,1007 1276,1007 C1364.364,
      <path d="M1300,847 L1300,907 L1340,907 C1351.08,907 1360,915.92 1360,927 C1360,938.08 1351.08,947 1340,
      <path d="M1246,757 L1246,787 L1266,787 C1271.54,787 1276,791.46 1276,797 C1276,802.54 1271.54,807 1266,
    </g>
    <g>
      <g>
        <path d="M1499,727 L1659,727 C1681.091,727 1699,744.909 1699,767 L1699,1047 C1699,1069.091 1681.091,1
        <path d="M1499,727 L1659,727 C1681.091,727 1699,744.909 1699,767 L1699,1047 C1699,1069.091 1681.091,1
      </g>
      <path d="M1499,747 L1659,747 C1670.046,747 1679,755.954 1679,767 L1679,1047 C1679,1058.046 1670.046,106
      <path d="M1639,807 C1550.636,807 1479,878.636 1479,967 C1479,989.092 1496.908,1007 1519,1007 C1607.364,
      <path d="M1579,847 C1556.908,847 1539,864.908 1539,887 L1539,927 C1539,949.092 1556.908,967 1579,967 C1
      <path d="M1509,757 C1497.954,757 1489,765.954 1489,777 L1489,797 C1488.996,806.053 1495.074,813.98 1503
    </g>
    <g>
      <g>
        <path d="M1985,727 L2145,727 C2167.091,727 2185,744.909 2185,767 L2185,1047 C2185,1069.091 2167.091,1
        <path d="M1985,727 L2145,727 C2167.091,727 2185,744.909 2185,767 L2185,1047 C2185,1069.091 2167.091,1
      </g>
      <path d="M1985,747 L2145,747 C2156.046,747 2165,755.954 2165,767 L2165,1047 C2165,1058.046 2156.046,106
      <path d="M2125,807 C2036.636,807 1965,878.636 1965,967 C1965,989.092 1982.908,1007 2005,1007 C2093.364,
      <path d="M2065,847 C2048.431,847 2035,860.431 2035,877 C2035,883.796 2037.32,889.972 2041.124,895 C2031
      <path d="M1995,757 C1986.716,757 1980,763.716 1980,772 C1980,775.4 1981.16,778.484 1983.064,781 C1977.9
    </g>
    <g>
      <g>
        <path d="M2228,727 L2388,727 C2410.091,727 2428,744.909 2428,767 L2428,1047 C2428,1069.091 2410.091,1
        <path d="M2228,727 L2388,727 C2410.091,727 2428,744.909 2428,767 L2428,1047 C2428,1069.091 2410.091,1
      </g>
      <path d="M2228,747 L2388,747 C2399.046,747 2408,755.954 2408,767 L2408,1047 C2408,1058.046 2399.046,106
      <path d="M2368,807 C2279.636,807 2208,878.636 2208,967 C2208,989.092 2225.908,1007 2248,1007 C2336.364,
      <path d="M2308,847 C2285.908,847 2268,864.908 2268,887 C2268,909.092 2285.908,927 2308,927 C2315.268,92
      <path d="M2238,817 C2249.046,817 2258,808.046 2258,797 L2258,777 C2258.004,767.947 2251.926,760.02 2243
    </g>
    <g>
      <g>
        <path d="M2471,727 L2631,727 C2653.091,727 2671,744.909 2671,767 L2671,1047 C2671,1069.091 2653.091,1
        <path d="M2471,727 L2631,727 C2653.091,727 2671,744.909 2671,767 L2671,1047 C2671,1069.091 2653.091,1
      </g>
      <path d="M2471,747 L2631,747 C2642.046,747 2651,755.954 2651,767 L2651,1047 C2651,1058.046 2642.046,106
      <path d="M2611,807 C2522.636,807 2451,878.636 2451,967 C2451,989.092 2468.908,1007 2491,1007 C2579.364,
      <path d="M2551.124,847 C2535.194,846.951 2519.901,853.248 2508.624,864.5 C2485.164,887.9 2485.1,925.912
      <path d="M2491.064,757 C2483.1,756.974 2475.453,760.122 2469.816,765.748 C2458.079,777.447 2458.05,796.
    </g>
    <g>
      <g>
        <path d="M2714,727 L2874,727 C2896.091,727 2914,744.909 2914,767 L2914,1047 C2914,1069.091 2896.091,1
        <path d="M2714,727 L2874,727 C2896.091,727 2914,744.909 2914,767 L2914,1047 C2914,1069.091 2896.091,1
      </g>
      <path d="M2714,747 L2874,747 C2885.046,747 2894,755.954 2894,767 L2894,1047 C2894,1058.046 2885.046,106
      <path d="M2854,807 C2765.636,807 2694,878.636 2694,967 C2694,989.092 2711.908,1007 2734,1007 C2822.364,
      <path d="M2794,842 L2804,852 L2764,892 C2754,902 2754,922 2764,932 L2784,912 L2824,872 L2834,882 L2834,
      <path d="M2794,972 L2784,962 L2824,922 C2834,912 2834,892 2824,882 L2804,902 L2764,942 L2754,932 L2754,
      <path d="M2724,757 L2729,762 L2709,782 C2704,787 2704,797 2709,802 L2719,792 L2739,772 L2744,777 L2744,
      <path d="M2724,822 L2719,817 L2739,797 C2744,792 2744,782 2739,777 L2729,787 L2709,807 L2704,802 L2704,
      <path d="M2864,1057 L2859,1052 L2879,1032 C2884,1027 2884,1017 2879,1012 L2869,1022 L2849,1042 L2844,10
    </g>
    <g>
      <g>
        <path d="M284,1090 L444,1090 C466.091,1090 484,1107.908 484,1130 L484,1410 C484,1432.091 466.091,1450
        <path d="M284,1090 L444,1090 C466.091,1090 484,1107.908 484,1130 L484,1410 C484,1432.091 466.091,1450
      </g>
      <path d="M284,1110 L444,1110 C455.046,1110 464,1118.954 464,1130 L464,1410 C464,1421.045 455.046,1430 4
      <path d="M424,1170 C335.636,1170 264,1241.636 264,1330 C264,1352.092 281.908,1370 304,1370 C392.364,137
      <path d="M356,1210 L336,1230 L336,1254 L356,1234 L356,1330 L376,1330 L376,1210 z" fill="#5555FF"/>
```

```
    <path d="M284,1120 L274,1130 L274,1142 L284,1132 L284,1180 L294,1180 L294,1120 z M444,1420 L454,1410 L4
  </g>
  <g>
    <g>
      <path d="M41,1090 L201,1090 C223.091,1090 241,1107.909 241,1130 L241,1410 C241,1432.091 223.091,1450
      <path d="M41,1090 L201,1090 C223.091,1090 241,1107.909 241,1130 L241,1410 C241,1432.091 223.091,1450
    </g>
    <path d="M41,1110 L201,1110 C212.046,1110 221,1118.954 221,1130 L221,1410 C221,1421.046 212.046,1430 2C
    <path d="M181,1170 C92.636,1170 21,1241.636 21,1330 C21,1352.092 38.908,1370 61,1370 C149.364,1370 221,
    <path d="M121,1210 C98.908,1210 81,1227.908 81,1250 L81,1290 C81,1312.092 98.908,1330 121,1330 C143.092
    <path d="M51,1120 C39.954,1120 31,1128.954 31,1140 L31,1160 C31,1171.046 39.954,1180 51,1180 C62.046,11
  </g>
  <g>
    <g>
      <path d="M1742,1090 L1902,1090 C1924.091,1090 1942,1107.908 1942,1130 L1942,1410 C1942,1432.091 1924.
      <path d="M1742,1090 L1902,1090 C1924.091,1090 1942,1107.908 1942,1130 L1942,1410 C1942,1432.091 1924.
    </g>
    <path d="M1742,1110 L1902,1110 C1913.046,1110 1922,1118.954 1922,1130 L1922,1410 C1922,1421.045 1913.04
    <path d="M1882,1170 C1793.636,1170 1722,1241.636 1722,1330 C1722,1352.092 1739.908,1370 1762,1370 C1850
    <path d="M1788,1210 L1788,1250 L1808,1250 L1808,1230 L1848,1230 L1808,1330 L1828,1330 L1868,1230 L1868,
    <path d="M1732,1120 L1732,1140 L1742,1140 L1742,1130 L1762,1130 L1742,1180 L1752,1180 L1772,1130 L1772,
  </g>
  <g>
    <g>
      <path d="M527,1090 L687,1090 C709.091,1090 727,1107.908 727,1130 L727,1410 C727,1432.091 709.091,1450
      <path d="M527,1090 L687,1090 C709.091,1090 727,1107.908 727,1130 L727,1410 C727,1432.091 709.091,1450
    </g>
    <path d="M527,1110 L687,1110 C698.046,1110 707,1118.954 707,1130 L707,1410 C707,1421.045 698.046,1430 6
    <path d="M667,1170 C578.636,1170 507,1241.636 507,1330 C507,1352.092 524.908,1370 547,1370 C635.364,137
    <path d="M603,1210 C580.84,1210 563,1227.84 563,1250 L563,1254 L583,1254 L583,1250 C583,1238.92 591.92,
    <path d="M537,1120 C525.92,1120 517,1128.92 517,1140 L517,1142 L527,1142 L527,1140 C527,1134.46 531.46,
  </g>
  <g>
    <g>
      <path d="M770,1090 L930,1090 C952.091,1090 970,1107.908 970,1130 L970,1410 C970,1432.091 952.091,1450
      <path d="M770,1090 L930,1090 C952.091,1090 970,1107.908 970,1130 L970,1410 C970,1432.091 952.091,1450
    </g>
    <path d="M770,1110 L930,1110 C941.046,1110 950,1118.954 950,1130 L950,1410 C950,1421.045 941.046,1430 9
    <path d="M910,1170 C821.636,1170 750,1241.636 750,1330 C750,1352.092 767.908,1370 790,1370 C878.364,137
    <path d="M850,1210 C833.431,1210 820,1223.431 820,1240 L840,1240 C840,1234.477 844.477,1230 850,1230 C8
    <path d="M780,1120 C771.716,1120 765,1126.715 765,1135 L775,1135 C775,1132.238 777.239,1130 780,1130 C7
  </g>
  <g>
    <g>
      <path d="M1013,1090 L1173,1090 C1195.091,1090 1213,1107.908 1213,1130 L1213,1410 C1213,1432.091 1195.
      <path d="M1013,1090 L1173,1090 C1195.091,1090 1213,1107.908 1213,1130 L1213,1410 C1213,1432.091 1195.
    </g>
    <path d="M1013,1110 L1173,1110 C1184.046,1110 1193,1118.954 1193,1130 L1193,1410 C1193,1421.045 1184.04
    <path d="M1153,1170 C1064.636,1170 993,1241.636 993,1330 C993,1352.092 1010.908,1370 1033,1370 C1121.36
    <path d="M1088.2,1210 L1049,1290 L1049,1310 L1097,1310 L1097,1330 L1117,1330 L1117,1310 L1129,1310 L112
    <path d="M1022.2,1120 L1002.6,1160 L1002.6,1170 L1026.6,1170 L1026.6,1180 L1036.6,1180 L1036.6,1170 L10
  </g>
  <g>
    <g>
      <path d="M1256,1090 L1416,1090 C1438.091,1090 1456,1107.908 1456,1130 L1456,1410 C1456,1432.091 1438.
      <path d="M1256,1090 L1416,1090 C1438.091,1090 1456,1107.908 1456,1130 L1456,1410 C1456,1432.091 1438.
    </g>
    <path d="M1256,1110 L1416,1110 C1427.046,1110 1436,1118.954 1436,1130 L1436,1410 C1436,1421.045 1427.04
    <path d="M1396,1170 C1307.636,1170 1236,1241.636 1236,1330 C1236,1352.092 1253.908,1370 1276,1370 C1364
    <path d="M1300,1210 L1300,1270 L1340,1270 C1351.08,1270 1360,1278.92 1360,1290 C1360,1301.08 1351.08,13
    <path d="M1246,1120 L1246,1150 L1266,1150 C1271.54,1150 1276,1154.46 1276,1160 C1276,1165.54 1271.54,11
  </g>
  <g>
    <g>
      <path d="M1499,1090 L1659,1090 C1681.091,1090 1699,1107.908 1699,1130 L1699,1410 C1699,1432.091 1681.
      <path d="M1499,1090 L1659,1090 C1681.091,1090 1699,1107.908 1699,1130 L1699,1410 C1699,1432.091 1681.
    </g>
    <path d="M1499,1110 L1659,1110 C1670.046,1110 1679,1118.954 1679,1130 L1679,1410 C1679,1421.045 1670.04
    <path d="M1639,1170 C1550.636,1170 1479,1241.636 1479,1330 C1479,1352.092 1496.908,1370 1519,1370 C1607
    <path d="M1579,1210 C1556.908,1210 1539,1227.908 1539,1250 L1539,1290 C1539,1312.092 1556.908,1330 1579
    <path d="M1509,1120 C1497.954,1120 1489,1128.954 1489,1140 L1489,1160 C1488.996,1169.052 1495.074,1176.
  </g>
  <g>
    <g>
      <path d="M1985,1090 L2145,1090 C2167.091,1090 2185,1107.908 2185,1130 L2185,1410 C2185,1432.091 2167.
      <path d="M1985,1090 L2145,1090 C2167.091,1090 2185,1107.908 2185,1130 L2185,1410 C2185,1432.091 2167.
    </g>
    <path d="M1985,1110 L2145,1110 C2156.046,1110 2165,1118.954 2165,1130 L2165,1410 C2165,1421.045 2156.04
    <path d="M2125,1170 C2036.636,1170 1965,1241.636 1965,1330 C1965,1352.092 1982.908,1370 2005,1370 C2093
    <path d="M2065,1210 C2048.431,1210 2035,1223.431 2035,1240 C2035,1246.796 2037.32,1252.972 2041.124,125
    <path d="M1995,1120 C1986.716,1120 1980,1126.715 1980,1135 C1980,1138.4 1981.16,1141.484 1983.064,1144
```

```
    </g>
    <g>
      <g>
        <path d="M2228,1090 L2388,1090 C2410.091,1090 2428,1107.908 2428,1130 L2428,1410 C2428,1432.091 2410.
        <path d="M2228,1090 L2388,1090 C2410.091,1090 2428,1107.908 2428,1130 L2428,1410 C2428,1432.091 2410.
      </g>
      <path d="M2228,1110 L2388,1110 C2399.046,1110 2408,1118.954 2408,1130 L2408,1410 C2408,1421.045 2399.04
      <path d="M2368,1170 C2279.636,1170 2208,1241.636 2208,1330 C2208,1352.092 2225.908,1370 2248,1370 C2336
      <path d="M2308,1210 C2285.908,1210 2268,1227.908 2268,1250 C2268,1272.092 2285.908,1290 2308,1290 C2315
      <path d="M2238,1180 C2249.046,1180 2258,1171.045 2258,1160 L2258,1140 C2258.004,1130.947 2251.926,1123.
    </g>
    <g>
      <g>
        <path d="M2471,1090 L2631,1090 C2653.091,1090 2671,1107.908 2671,1130 L2671,1410 C2671,1432.091 2653.
        <path d="M2471,1090 L2631,1090 C2653.091,1090 2671,1107.908 2671,1130 L2671,1410 C2671,1432.091 2653.
      </g>
      <path d="M2471,1110 L2631,1110 C2642.046,1110 2651,1118.954 2651,1130 L2651,1410 C2651,1421.045 2642.04
      <path d="M2611,1170 C2522.636,1170 2451,1241.636 2451,1330 C2451,1352.092 2468.908,1370 2491,1370 C2579
      <path d="M2551.124,1210 C2535.194,1209.95 2519.901,1216.248 2508.624,1227.5 C2485.164,1250.9 2485.1,128
      <path d="M2491.064,1120 C2483.1,1119.974 2475.453,1123.122 2469.816,1128.748 C2458.079,1140.447 2458.05
    </g>
    <g>
      <g>
        <path d="M2714,1090 L2874,1090 C2896.091,1090 2914,1107.908 2914,1130 L2914,1410 C2914,1432.091 2896.
        <path d="M2714,1090 L2874,1090 C2896.091,1090 2914,1107.908 2914,1130 L2914,1410 C2914,1432.091 2896.
      </g>
      <path d="M2714,1110 L2874,1110 C2885.046,1110 2894,1118.954 2894,1130 L2894,1410 C2894,1421.045 2885.04
      <path d="M2854,1170 C2765.636,1170 2694,1241.636 2694,1330 C2694,1352.092 2711.908,1370 2734,1370 C2822
      <path d="M2794,1205 L2804,1215 L2764,1255 C2754,1265 2754,1285 2764,1295 L2784,1275 L2824,1235 L2834,12
      <path d="M2794,1335 L2784,1325 L2824,1285 C2834,1275 2834,1255 2824,1245 L2804,1265 L2764,1305 L2754,12
      <path d="M2724,1120 L2729,1125 L2709,1145 C2704,1150 2704,1160 2709,1165 L2719,1155 L2739,1135 L2744,11
      <path d="M2724,1185 L2719,1180 L2739,1160 C2744,1155 2744,1145 2739,1140 L2729,1150 L2709,1170 L2704,11
      <path d="M2864,1420 L2859,1415 L2879,1395 C2884,1390 2884,1380 2879,1375 L2869,1385 L2849,1405 L2844,14
    </g>
    <g>
      <g>
        <path d="M2957,1090 L3117,1090 C3139.091,1090 3157,1107.908 3157,1130 L3157,1410 C3157,1432.091 3139.
        <path d="M2957,1090 L3117,1090 C3139.091,1090 3157,1107.908 3157,1130 L3157,1410 C3157,1432.091 3139.
      </g>
      <path d="M2957,1110 L3117,1110 C3128.046,1110 3137,1118.954 3137,1130 L3137,1410 C3137,1421.045 3128.04
      <path d="M3097,1170 C3008.636,1170 2937,1241.636 2937,1330 C2937,1352.092 2954.908,1370 2977,1370 C3065
      <g>
        <path d="M3009.894,1239.986 L3049.894,1239.986 C3055.417,1239.986 3058.739,1244.03 3057.314,1249.017
        <path d="M3009.894,1239.986 L3049.894,1239.986 C3055.417,1239.986 3058.739,1244.03 3057.314,1249.017
      </g>
      <path d="M3017.037,1249.986 L3037.037,1249.986 C3042.56,1249.986 3045.882,1254.029 3044.457,1259.017 L3
      <g>
        <path d="M3049.894,1209.983 L3089.894,1209.983 C3095.417,1209.983 3098.739,1214.026 3097.314,1219.014
        <path d="M3049.894,1209.983 L3089.894,1209.983 C3095.417,1209.983 3098.739,1214.026 3097.314,1219.014
      </g>
      <path d="M3057.037,1219.983 L3077.037,1219.983 C3082.56,1219.983 3085.882,1224.026 3084.457,1229.014 L3
      <path d="M2951,1140 L2951,1150 L2941,1150 L2941,1160 L2951,1160 L2951,1170 L2961,1170 L2961,1160 L2971,
      <path d="M2997,1120 C2985.92,1120 2977,1128.92 2977,1140 L2977,1142 L2987,1142 L2987,1140 C2987,1134.46
      <path d="M3123,1400 L3123,1390 L3133,1390 L3133,1380 L3123,1380 L3123,1370 L3113,1370 L3113,1380 L3103,
      <path d="M3077,1420 C3088.08,1420 3097,1411.08 3097,1400 L3097,1398 L3087,1398 L3087,1400 C3087,1405.54
    </g>
    <g>
      <g>
        <path d="M2957,727 L3117,727 C3139.091,727 3157,744.909 3157,767 L3157,1047 C3157,1069.091 3139.091,1
        <path d="M2957,727 L3117,727 C3139.091,727 3157,744.909 3157,767 L3157,1047 C3157,1069.091 3139.091,1
      </g>
      <path d="M2957,747 L3117,747 C3128.046,747 3137,755.954 3137,767 L3137,1047 C3137,1058.046 3128.046,106
      <path d="M3097,807 C3008.636,807 2937,878.636 2937,967 C2937,989.092 2954.908,1007 2977,1007 C3065.364,
      <g>
        <path d="M3009.893,876.986 L3049.893,876.986 C3055.416,876.986 3058.738,881.029 3057.313,886.016 L303
        <path d="M3009.893,876.986 L3049.893,876.986 C3055.416,876.986 3058.738,881.029 3057.313,886.016 L303
      </g>
      <path d="M3017.036,886.986 L3037.036,886.986 C3042.559,886.986 3045.881,891.029 3044.456,896.016 L3029.
      <g>
        <path d="M3049.892,846.987 L3089.892,846.987 C3095.415,846.987 3098.737,851.03 3097.312,856.017 L3076
        <path d="M3049.892,846.987 L3089.892,846.987 C3095.415,846.987 3098.737,851.03 3097.312,856.017 L3076
      </g>
      <path d="M3057.035,856.986 L3077.035,856.986 C3082.558,856.986 3085.88,861.029 3084.455,866.017 L3069.6
      <path d="M2951,777 L2951,787 L2941,787 L2941,797 L2951,797 L2951,807 L2961,807 L2961,797 L2971,797 L297
      <path d="M2997,757 C2985.92,757 2977,765.92 2977,777 L2977,779 L2987,779 L2987,777 C2987,771.46 2991.46
      <path d="M3123,1037 L3123,1027 L3133,1027 L3133,1017 L3123,1017 L3123,1007 L3113,1007 L3113,1017 L3103,
      <path d="M3077,1057 C3088.08,1057 3097,1048.08 3097,1037 L3097,1035 L3087,1035 L3087,1037 C3087,1042.54
    </g>
    <g>
      <g>
        <path d="M2957,364 L3117,364 C3139.091,364 3157,381.909 3157,404 L3157,684 C3157,706.091 3139.091,724
```

```
      <path d="M2957,364 L3117,364 C3139.091,364 3157,381.909 3157,404 L3157,684 C3157,706.091 3139.091,724
    </g>
    <path d="M2957,384 L3117,384 C3128.046,384 3137,392.954 3137,404 L3137,684 C3137,695.046 3128.046,704 3
    <path d="M3097,444 C3008.636,444 2937,515.636 2937,604 C2937,626.092 2954.908,644 2977,644 C3065.364,64
    <g>
      <path d="M3009.886,513.989 L3049.886,513.989 C3055.409,513.989 3058.731,518.032 3057.306,523.02 L3036
      <path d="M3009.886,513.989 L3049.886,513.989 C3055.409,513.989 3058.731,518.032 3057.306,523.02 L3036
    </g>
    <path d="M3017.03,523.989 L3037.03,523.989 C3042.552,523.989 3045.874,528.032 3044.449,533.019 L3029.61
    <g>
      <path d="M3049.885,483.99 L3089.885,483.99 C3095.408,483.99 3098.73,488.033 3097.305,493.02 L3076.752
      <path d="M3049.885,483.99 L3089.885,483.99 C3095.408,483.99 3098.73,488.033 3097.305,493.02 L3076.752
    </g>
    <path d="M3057.029,493.989 L3077.029,493.989 C3082.552,493.989 3085.873,498.033 3084.448,503.02 L3069.6
    <path d="M2951,414 L2951,424 L2941,424 L2941,434 L2951,434 L2951,444 L2961,444 L2961,434 L2971,434 L297
    <path d="M2997,394 C2985.92,394 2977,402.92 2977,414 L2977,416 L2987,416 L2987,414 C2987,408.46 2991.46
    <path d="M3123,674 L3123,664 L3133,664 L3133,654 L3123,654 L3123,644 L3113,644 L3113,654 L3103,654 L310
    <path d="M3077,694 C3088.08,694 3097,685.08 3097,674 L3097,672 L3087,672 L3087,674 C3087,679.54 3082.54
  </g>
  <g>
    <g>
      <path d="M2957,1 L3117,1 C3139.091,1 3157,18.909 3157,41 L3157,321 C3157,343.091 3139.091,361 3117,36
      <path d="M2957,1 L3117,1 C3139.091,1 3157,18.909 3157,41 L3157,321 C3157,343.091 3139.091,361 3117,36
    </g>
    <path d="M2957,21 L3117,21 C3128.046,21 3137,29.954 3137,41 L3137,321 C3137,332.046 3128.046,341 3117,3
    <path d="M3097,81 C3008.636,81 2937,152.636 2937,241 C2937,263.092 2954.908,281 2977,281 C3065.364,281
    <g>
      <path d="M3009.885,150.988 L3049.885,150.988 C3055.408,150.988 3058.73,155.031 3057.305,160.019 L3036
      <path d="M3009.885,150.988 L3049.885,150.988 C3055.408,150.988 3058.73,155.031 3057.305,160.019 L3036
    </g>
    <path d="M3017.028,160.988 L3037.028,160.988 C3042.551,160.988 3045.873,165.031 3044.448,170.019 L3029.
    <g>
      <path d="M3049.884,120.989 L3089.884,120.989 C3095.407,120.989 3098.729,125.032 3097.304,130.019 L307
      <path d="M3049.884,120.989 L3089.884,120.989 C3095.407,120.989 3098.729,125.032 3097.304,130.019 L307
    </g>
    <path d="M3057.027,130.989 L3077.027,130.989 C3082.55,130.989 3085.872,135.032 3084.447,140.019 L3069.6
    <path d="M2951,51 L2951,61 L2941,61 L2941,71 L2951,71 L2951,81 L2961,81 L2961,71 L2971,71 L2971,61 L296
    <path d="M2997,31 C2985.92,31 2977,39.92 2977,51 L2977,53 L2987,53 L2987,51 C2987,45.46 2991.46,41 2997
    <path d="M3123,311 L3123,301 L3133,301 L3133,291 L3123,291 L3123,281 L3113,281 L3113,291 L3103,291 L310
    <path d="M3077,331 C3088.08,331 3097,322.08 3097,311 L3097,309 L3087,309 L3087,311 C3087,316.54 3082.54
  </g>
  <g>
    <g>
      <path d="M3200,1453 L3360,1453 C3382.091,1453 3400,1470.909 3400,1493 L3400,1773 C3400,1795.091 3382.
      <path d="M3200,1453 L3360,1453 C3382.091,1453 3400,1470.909 3400,1493 L3400,1773 C3400,1795.091 3382.
    </g>
    <path d="M3200,1473 L3360,1473 C3371.046,1473 3380,1481.954 3380,1493 L3380,1773 C3380,1784.046 3371.04
    <path d="M3340,1533 C3251.634,1533 3180,1604.635 3180,1693 C3180,1715.092 3197.909,1733 3220,1733 C3308
    <path d="M3191.75,1633 C3184.237,1651.542 3180,1671.764 3180,1693 C3180,1715.092 3197.909,1733 3220,173
    <path d="M3280,1633 L3220,1733 C3287.129,1733 3344.502,1691.61 3368.25,1633 L3280,1633 z" fill="#00AA00
    <path d="M3340,1533 C3272.871,1533 3215.498,1574.39 3191.75,1633 L3280,1633 L3340,1533 z" fill="#FF5555
    <path d="M3340,1533 L3280,1633 L3368.25,1633 C3375.763,1614.458 3380,1594.237 3380,1573 C3380,1550.909
    <path d="M3192.938,1508 C3191,1512.763 3190.002,1517.857 3190,1523 C3190,1528.523 3194.477,1533 3200,15
    <path d="M3215,1508 L3200,1533 C3216.782,1533 3231.126,1522.653 3237.063,1508 L3215,1508 z" fill="#00AA
    <path d="M3230,1483 C3213.218,1483 3198.874,1493.347 3192.937,1508 L3215,1508 L3230,1483 z" fill="#FF55
    <path d="M3230,1483 L3215,1508 L3237.063,1508 C3239.001,1503.236 3239.998,1498.143 3240,1493 C3240,1487
    <path d="M3322.938,1758 C3321,1762.763 3320.002,1767.857 3320,1773 C3320,1778.523 3324.477,1783 3330,17
    <path d="M3345,1758 L3330,1783 C3346.782,1783 3361.126,1772.653 3367.063,1758 L3345,1758 z" fill="#00AA
    <path d="M3360,1733 C3343.218,1733 3328.874,1743.347 3322.937,1758 L3345,1758 L3360,1733 z" fill="#FF55
    <path d="M3360,1733 L3345,1758 L3367.063,1758 C3369.001,1753.236 3369.998,1748.143 3370,1743 C3370,1737
    <path d="M3340,1533 C3272.871,1533 3215.498,1574.39 3191.75,1633 C3184.237,1651.542 3180,1671.763 3180,
  </g>
  <g>
    <g>
      <path d="M2957,1453 L3117,1453 C3139.091,1453 3157,1470.909 3157,1493 L3157,1773 C3157,1795.091 3139.
      <path d="M2957,1453 L3117,1453 C3139.091,1453 3157,1470.909 3157,1493 L3157,1773 C3157,1795.091 3139.
    </g>
    <path d="M2957,1473 L3117,1473 C3128.046,1473 3137,1481.954 3137,1493 L3137,1773 C3137,1784.046 3128.04
    <path d="M3097,1533 C3008.636,1533 2937,1604.636 2937,1693 C2937,1715.092 2954.908,1733 2977,1733 C3065
    <g>
      <text transform="matrix(0.715, -0.7, 0.7, 0.715, 3040.547, 1632.188)">
        <tspan x="-100.498" y="30" font-family="Arial-BoldMT" font-size="90" fill="#FFFFFF">UNO</tspan>
      </text>
    </g>
  </g>
  <g>
    <g>
      <path d="M3200,1 L3360,1 C3382.091,1 3400,18.909 3400,41 L3400,321 C3400,343.091 3382.091,361 3360,36
      <path d="M3200,1 L3360,1 C3382.091,1 3400,18.909 3400,41 L3400,321 C3400,343.091 3382.091,361 3360,36
    </g>
    <path d="M3200,21 L3360,21 C3371.046,21 3380,29.954 3380,41 L3380,321 C3380,332.046 3371.046,341 3360,3
```

```
    <path d="M3340,81 C3251.634,81 3180,152.635 3180,241 C3180,263.092 3197.909,281 3220,281 C3308.366,281
    <path d="M3191.75,181 C3184.237,199.542 3180,219.764 3180,241 C3180,263.092 3197.909,281 3220,281 L3280
    <path d="M3280,181 L3220,281 C3287.129,281 3344.502,239.61 3368.25,181 L3280,181 z" fill="#00AA00"/>
    <path d="M3340,81 C3272.871,81 3215.498,122.39 3191.75,181 L3280,181 L3340,81 z" fill="#FF5555"/>
    <path d="M3340,81 L3280,181 L3368.25,181 C3375.763,162.458 3380,142.237 3380,121 C3380,98.909 3362.091,
    <path d="M3192.938,56 C3191,60.763 3190.002,65.857 3190,71 C3190,76.523 3194.477,81 3200,81 L3215,56 L3
    <path d="M3215,56 L3200,81 C3216.782,81 3231.126,70.653 3237.063,56 L3215,56 z" fill="#00AA00"/>
    <path d="M3230,31 C3213.218,31 3198.874,41.347 3192.937,56 L3215,56 L3230,31 z" fill="#FF5555"/>
    <path d="M3230,31 L3215,56 L3237.063,56 C3239.001,51.236 3239.998,46.143 3240,41 C3240,35.477 3235.523,
    <path d="M3322.938,306 C3321,310.763 3320.002,315.857 3320,321 C3320,326.523 3324.477,331 3330,331 L334
    <path d="M3345,306 L3330,331 C3346.782,331 3361.126,320.653 3367.063,306 L3345,306 z" fill="#00AA00"/>
    <path d="M3360,281 C3343.218,281 3328.874,291.347 3322.937,306 L3345,306 L3360,281 z" fill="#FF5555"/>
    <path d="M3360,281 L3345,306 L3367.063,306 C3369.001,301.236 3369.998,296.143 3370,291 C3370,285.477 33
    <path d="M3340,81 C3272.871,81 3215.498,122.39 3191.75,181 C3184.237,199.542 3180,219.763 3180,241 C318
  </g>
  <g>
    <g>
      <path d="M3200,364 L3360,364 C3382.091,364 3400,381.909 3400,404 L3400,684 C3400,706.091 3382.091,724
      <path d="M3200,364 L3360,364 C3382.091,364 3400,381.909 3400,404 L3400,684 C3400,706.091 3382.091,724
    </g>
    <path d="M3200,384 L3360,384 C3371.046,384 3380,392.954 3380,404 L3380,684 C3380,695.046 3371.046,704 3
    <path d="M3340,444 C3251.634,444 3180,515.635 3180,604 C3180,626.092 3197.909,644 3220,644 C3308.366,64
    <path d="M3191.75,544 C3184.237,562.542 3180,582.764 3180,604 C3180,626.092 3197.909,644 3220,644 L3280
    <path d="M3280,544 L3220,644 C3287.129,644 3344.502,602.61 3368.25,544 L3280,544 z" fill="#00AA00"/>
    <path d="M3340,444 C3272.871,444 3215.498,485.39 3191.75,544 L3280,544 L3340,444 z" fill="#FF5555"/>
    <path d="M3340,444 L3280,544 L3368.25,544 C3375.763,525.458 3380,505.237 3380,484 C3380,461.909 3362.09
    <path d="M3192.938,419 C3191,423.763 3190.002,428.857 3190,434 C3190,439.523 3194.477,444 3200,444 L321
    <path d="M3215,419 L3200,444 C3216.782,444 3231.126,433.653 3237.063,419 L3215,419 z" fill="#00AA00"/>
    <path d="M3230,394 C3213.218,394 3198.874,404.347 3192.937,419 L3215,419 L3230,394 z" fill="#FF5555"/>
    <path d="M3230,394 L3215,419 L3237.063,419 C3239.001,414.236 3239.998,409.143 3240,404 C3240,398.477 32
    <path d="M3322.938,669 C3321,673.763 3320.002,678.857 3320,684 C3320,689.523 3324.477,694 3330,694 L334
    <path d="M3345,669 L3330,694 C3346.782,694 3361.126,683.653 3367.063,669 L3345,669 z" fill="#00AA00"/>
    <path d="M3360,644 C3343.218,644 3328.874,654.347 3322.937,669 L3345,669 L3360,644 z" fill="#FF5555"/>
    <path d="M3360,644 L3345,669 L3367.063,669 C3369.001,664.236 3369.998,659.143 3370,654 C3370,648.477 33
    <path d="M3340,444 C3272.871,444 3215.498,485.39 3191.75,544 C3184.237,562.542 3180,582.763 3180,604 C3
  </g>
  <g>
    <g>
      <path d="M3200,727 L3360,727 C3382.091,727 3400,744.909 3400,767 L3400,1047 C3400,1069.091 3382.091,1
      <path d="M3200,727 L3360,727 C3382.091,727 3400,744.909 3400,767 L3400,1047 C3400,1069.091 3382.091,1
    </g>
    <path d="M3200,747 L3360,747 C3371.046,747 3380,755.954 3380,767 L3380,1047 C3380,1058.046 3371.046,106
    <path d="M3340,807 C3251.634,807 3180,878.635 3180,967 C3180,989.092 3197.909,1007 3220,1007 C3308.366,
    <path d="M3191.75,907 C3184.237,925.542 3180,945.764 3180,967 C3180,989.092 3197.909,1007 3220,1007 L32
    <path d="M3280,907 L3220,1007 C3287.129,1007 3344.502,965.61 3368.25,907 L3280,907 z" fill="#00AA00"/>
    <path d="M3340,807 C3272.871,807 3215.498,848.39 3191.75,907 L3280,907 L3340,807 z" fill="#FF5555"/>
    <path d="M3340,807 L3280,907 L3368.25,907 C3375.763,888.458 3380,868.237 3380,847 C3380,824.909 3362.09
    <path d="M3192.938,782 C3191,786.763 3190.002,791.857 3190,797 C3190,802.523 3194.477,807 3200,807 L321
    <path d="M3215,782 L3200,807 C3216.782,807 3231.126,796.653 3237.063,782 L3215,782 z" fill="#00AA00"/>
    <path d="M3230,757 C3213.218,757 3198.874,767.347 3192.937,782 L3215,782 L3230,757 z" fill="#FF5555"/>
    <path d="M3230,757 L3215,782 L3237.063,782 C3239.001,777.236 3239.998,772.143 3240,767 C3240,761.477 32
    <path d="M3322.938,1032 C3321,1036.763 3320.002,1041.857 3320,1047 C3320,1052.523 3324.477,1057 3330,10
    <path d="M3345,1032 L3330,1057 C3346.782,1057 3361.126,1046.653 3367.063,1032 L3345,1032 z" fill="#00AA
    <path d="M3360,1007 C3343.218,1007 3328.874,1017.347 3322.937,1032 L3345,1032 L3360,1007 z" fill="#FF55
    <path d="M3360,1007 L3345,1032 L3367.063,1032 C3369.001,1027.236 3369.998,1022.143 3370,1017 C3370,1011
    <path d="M3340,807 C3272.871,807 3215.498,848.39 3191.75,907 C3184.237,925.542 3180,945.763 3180,967 C3
  </g>
  <g>
    <g>
      <path d="M3200,1090 L3360,1090 C3382.091,1090 3400,1107.909 3400,1130 L3400,1410 C3400,1432.091 3382.
      <path d="M3200,1090 L3360,1090 C3382.091,1090 3400,1107.909 3400,1130 L3400,1410 C3400,1432.091 3382.
    </g>
    <path d="M3200,1110 L3360,1110 C3371.046,1110 3380,1118.954 3380,1130 L3380,1410 C3380,1421.046 3371.04
    <path d="M3340,1170 C3251.634,1170 3180,1241.635 3180,1330 C3180,1352.092 3197.909,1370 3220,1370 C3308
    <path d="M3191.75,1270 C3184.237,1288.542 3180,1308.764 3180,1330 C3180,1352.092 3197.909,1370 3220,137
    <path d="M3280,1270 L3220,1370 C3287.129,1370 3344.502,1328.61 3368.25,1270 L3280,1270 z" fill="#00AA00
    <path d="M3340,1170 C3272.871,1170 3215.498,1211.39 3191.75,1270 L3280,1270 L3340,1170 z" fill="#FF5555
    <path d="M3340,1170 L3280,1270 L3368.25,1270 C3375.763,1251.458 3380,1231.237 3380,1210 C3380,1187.909
    <path d="M3192.938,1145 C3191,1149.763 3190.002,1154.857 3190,1160 C3190,1165.523 3194.477,1170 3200,11
    <path d="M3215,1145 L3200,1170 C3216.782,1170 3231.126,1159.653 3237.063,1145 L3215,1145 z" fill="#00AA
    <path d="M3230,1120 C3213.218,1120 3198.874,1130.347 3192.937,1145 L3215,1145 L3230,1120 z" fill="#FF55
    <path d="M3230,1120 L3215,1145 L3237.063,1145 C3239.001,1140.236 3239.998,1135.143 3240,1130 C3240,1124
    <path d="M3322.938,1395 C3321,1399.763 3320.002,1404.857 3320,1410 C3320,1415.523 3324.477,1420 3330,14
    <path d="M3345,1395 L3330,1420 C3346.782,1420 3361.126,1409.653 3367.063,1395 L3345,1395 z" fill="#00AA
    <path d="M3360,1370 C3343.218,1370 3328.874,1380.347 3322.937,1395 L3345,1395 L3360,1370 z" fill="#FF55
    <path d="M3360,1370 L3345,1395 L3367.063,1395 C3369.001,1390.236 3369.998,1385.143 3370,1380 C3370,1374
    <path d="M3340,1170 C3272.871,1170 3215.498,1211.39 3191.75,1270 C3184.237,1288.542 3180,1308.763 3180,
  </g>
  <g>
    <g>
      <path d="M3443,1 L3603,1 C3625.091,1 3643,18.909 3643,41 L3643,321 C3643,343.091 3625.091,361 3603,36
```

```
      <path d="M3443,1 L3603,1 C3625.091,1 3643,18.909 3643,41 L3643,321 C3643,343.091 3625.091,361 3603,36
    </g>
    <path d="M3443,21 L3603,21 C3614.046,21 3623,29.954 3623,41 L3623,321 C3623,332.046 3614.046,341 3603,3
    <path d="M3583,81 C3494.636,81 3423,152.636 3423,241 C3423,263.092 3440.908,281 3463,281 C3551.364,281
    <path d="M3482.6,31 L3463,71 L3463,81 L3487,81 L3487,91 L3497,91 L3497,81 L3503,81 L3503,71 L3497,71 L3
    <path d="M3437,51 L3437,61 L3427,61 L3427,71 L3437,71 L3437,81 L3447,81 L3447,71 L3457,71 L3457,61 L344
    <path d="M3563.4,331 L3583,291 L3583,281 L3559,281 L3559,271 L3549,271 L3549,281 L3543,281 L3543,291 L3
    <path d="M3609,311 L3609,301 L3619,301 L3619,291 L3609,291 L3609,281 L3599,281 L3599,291 L3589,291 L358
    <g>
      <g>
        <path d="M3485.86,130.995 L3525.86,130.995 C3531.383,130.995 3534.705,135.038 3533.28,140.025 L3512
        <path d="M3485.86,130.995 L3525.86,130.995 C3531.383,130.995 3534.705,135.038 3533.28,140.025 L3512
      </g>
      <path d="M3493.003,140.995 L3513.003,140.995 C3518.526,140.995 3521.848,145.038 3520.423,150.025 L350
    </g>
    <g>
      <g>
        <path d="M3525.86,100.995 L3565.86,100.995 C3571.383,100.995 3574.705,105.038 3573.28,110.025 L3552
        <path d="M3525.86,100.995 L3565.86,100.995 C3571.383,100.995 3574.705,105.038 3573.28,110.025 L3552
      </g>
      <path d="M3533.003,110.995 L3553.003,110.995 C3558.526,110.995 3561.848,115.038 3560.423,120.025 L354
    </g>
    <g>
      <g>
        <path d="M3515.86,160.995 L3555.86,160.995 C3561.383,160.995 3564.705,165.038 3563.28,170.026 L3542
        <path d="M3515.86,160.995 L3555.86,160.995 C3561.383,160.995 3564.705,165.038 3563.28,170.026 L3542
      </g>
      <path d="M3523.003,170.995 L3543.003,170.995 C3548.526,170.995 3551.848,175.038 3550.423,180.025 L353
    </g>
    <g>
      <g>
        <path d="M3555.86,130.995 L3595.86,130.995 C3601.383,130.995 3604.705,135.038 3603.28,140.025 L3582
        <path d="M3555.86,130.995 L3595.86,130.995 C3601.383,130.995 3604.705,135.038 3603.28,140.025 L3582
      </g>
      <path d="M3563.003,140.995 L3583.003,140.995 C3588.526,140.995 3591.848,145.038 3590.423,150.025 L357
    </g>
  </g>
  <g>
    <g>
      <path d="M3443,364 L3603,364 C3625.091,364 3643,381.909 3643,404 L3643,684 C3643,706.091 3625.091,724
      <path d="M3443,364 L3603,364 C3625.091,364 3643,381.909 3643,404 L3643,684 C3643,706.091 3625.091,724
    </g>
    <path d="M3443,384 L3603,384 C3614.046,384 3623,392.954 3623,404 L3623,684 C3623,695.046 3614.046,704 3
    <path d="M3583,444 C3494.636,444 3423,515.636 3423,604 C3423,626.092 3440.908,644 3463,644 C3551.364,64
    <path d="M3482.6,394 L3463,434 L3463,444 L3487,444 L3487,454 L3497,454 L3497,444 L3503,444 L3503,434 L3
    <path d="M3437,414 L3437,424 L3427,424 L3427,434 L3437,434 L3437,444 L3447,444 L3447,434 L3457,434 L345
    <path d="M3563.4,694 L3583,654 L3583,644 L3559,644 L3559,634 L3549,634 L3549,644 L3543,644 L3543,654 L3
    <path d="M3609,674 L3609,664 L3619,664 L3619,654 L3609,654 L3609,644 L3599,644 L3599,654 L3589,654 L358
    <g>
      <g>
        <path d="M3485.86,493.995 L3525.86,493.995 C3531.383,493.995 3534.705,498.038 3533.28,503.025 L3512
        <path d="M3485.86,493.995 L3525.86,493.995 C3531.383,493.995 3534.705,498.038 3533.28,503.025 L3512
      </g>
      <path d="M3493.003,503.995 L3513.003,503.995 C3518.526,503.995 3521.848,508.038 3520.423,513.025 L350
    </g>
    <g>
      <g>
        <path d="M3525.86,463.995 L3565.86,463.995 C3571.383,463.995 3574.705,468.038 3573.28,473.025 L3552
        <path d="M3525.86,463.995 L3565.86,463.995 C3571.383,463.995 3574.705,468.038 3573.28,473.025 L3552
      </g>
      <path d="M3533.003,473.995 L3553.003,473.995 C3558.526,473.995 3561.848,478.038 3560.423,483.025 L354
    </g>
    <g>
      <g>
        <path d="M3515.86,523.995 L3555.86,523.995 C3561.383,523.995 3564.705,528.038 3563.28,533.026 L3542
        <path d="M3515.86,523.995 L3555.86,523.995 C3561.383,523.995 3564.705,528.038 3563.28,533.026 L3542
      </g>
      <path d="M3523.003,533.995 L3543.003,533.995 C3548.526,533.995 3551.848,538.038 3550.423,543.025 L353
    </g>
    <g>
      <g>
        <path d="M3555.86,493.995 L3595.86,493.995 C3601.383,493.995 3604.705,498.038 3603.28,503.025 L3582
        <path d="M3555.86,493.995 L3595.86,493.995 C3601.383,493.995 3604.705,498.038 3603.28,503.025 L3582
      </g>
      <path d="M3563.003,503.995 L3583.003,503.995 C3588.526,503.995 3591.848,508.038 3590.423,513.025 L357
    </g>
  </g>
  <g>
    <g>
      <path d="M3443,727 L3603,727 C3625.091,727 3643,744.909 3643,767 L3643,1047 C3643,1069.091 3625.091,1
      <path d="M3443,727 L3603,727 C3625.091,727 3643,744.909 3643,767 L3643,1047 C3643,1069.091 3625.091,1
```

```
      </g>
      <path d="M3443,747 L3603,747 C3614.046,747 3623,755.954 3623,767 L3623,1047 C3623,1058.046 3614.046,106
      <path d="M3583,807 C3494.636,807 3423,878.636 3423,967 C3423,989.092 3440.908,1007 3463,1007 C3551.364,
      <path d="M3482.6,757 L3463,797 L3463,807 L3487,807 L3487,817 L3497,817 L3497,807 L3503,807 L3503,797 L3
      <path d="M3437,777 L3437,787 L3427,787 L3427,797 L3437,797 L3437,807 L3447,807 L3447,797 L3457,797 L345
      <path d="M3563.4,1057 L3583,1017 L3583,1007 L3559,1007 L3559,997 L3549,997 L3549,1007 L3543,1007 L3543,
      <path d="M3609,1037 L3609,1027 L3619,1027 L3619,1017 L3609,1017 L3609,1007 L3599,1007 L3599,1017 L3589,
      <g>
        <g>
          <path d="M3485.86,856.995 L3525.86,856.995 C3531.383,856.995 3534.705,861.038 3533.28,866.025 L3512
          <path d="M3485.86,856.995 L3525.86,856.995 C3531.383,856.995 3534.705,861.038 3533.28,866.025 L3512
        </g>
        <path d="M3493.003,866.995 L3513.003,866.995 C3518.526,866.995 3521.848,871.038 3520.423,876.025 L350
      </g>
      <g>
        <g>
          <path d="M3525.86,826.995 L3565.86,826.995 C3571.383,826.995 3574.705,831.038 3573.28,836.025 L3552
          <path d="M3525.86,826.995 L3565.86,826.995 C3571.383,826.995 3574.705,831.038 3573.28,836.025 L3552
        </g>
        <path d="M3533.003,836.995 L3553.003,836.995 C3558.526,836.995 3561.848,841.038 3560.423,846.025 L354
      </g>
      <g>
        <g>
          <path d="M3515.86,886.995 L3555.86,886.995 C3561.383,886.995 3564.705,891.038 3563.28,896.026 L3542
          <path d="M3515.86,886.995 L3555.86,886.995 C3561.383,886.995 3564.705,891.038 3563.28,896.026 L3542
        </g>
        <path d="M3523.003,896.995 L3543.003,896.995 C3548.526,896.995 3551.848,901.038 3550.423,906.025 L353
      </g>
      <g>
        <g>
          <path d="M3555.86,856.995 L3595.86,856.995 C3601.383,856.995 3604.705,861.038 3603.28,866.025 L3582
          <path d="M3555.86,856.995 L3595.86,856.995 C3601.383,856.995 3604.705,861.038 3603.28,866.025 L3582
        </g>
        <path d="M3563.003,866.995 L3583.003,866.995 C3588.526,866.995 3591.848,871.038 3590.423,876.025 L357
      </g>
    </g>
    <g>
      <g>
        <path d="M3443,1090 L3603,1090 C3625.091,1090 3643,1107.909 3643,1130 L3643,1410 C3643,1432.091 3625.
        <path d="M3443,1090 L3603,1090 C3625.091,1090 3643,1107.909 3643,1130 L3643,1410 C3643,1432.091 3625.
      </g>
      <path d="M3443,1110 L3603,1110 C3614.046,1110 3623,1118.954 3623,1130 L3623,1410 C3623,1421.046 3614.04
      <path d="M3583,1170 C3494.636,1170 3423,1241.636 3423,1330 C3423,1352.092 3440.908,1370 3463,1370 C3551
      <path d="M3482.6,1120 L3463,1160 L3463,1170 L3487,1170 L3487,1180 L3497,1180 L3497,1170 L3503,1170 L350
      <path d="M3437,1140 L3437,1150 L3427,1150 L3427,1160 L3437,1160 L3437,1170 L3447,1170 L3447,1160 L3457,
      <path d="M3563.4,1420 L3583,1380 L3583,1370 L3559,1370 L3559,1360 L3549,1360 L3549,1370 L3543,1370 L354
      <path d="M3609,1400 L3609,1390 L3619,1390 L3619,1380 L3609,1380 L3609,1370 L3599,1370 L3599,1380 L3589,
      <g>
        <g>
          <path d="M3485.86,1219.995 L3525.86,1219.995 C3531.383,1219.995 3534.705,1224.038 3533.28,1229.025
          <path d="M3485.86,1219.995 L3525.86,1219.995 C3531.383,1219.995 3534.705,1224.038 3533.28,1229.025
        </g>
        <path d="M3493.003,1229.995 L3513.003,1229.995 C3518.526,1229.995 3521.848,1234.038 3520.423,1239.025
      </g>
      <g>
        <g>
          <path d="M3525.86,1189.995 L3565.86,1189.995 C3571.383,1189.995 3574.705,1194.038 3573.28,1199.025
          <path d="M3525.86,1189.995 L3565.86,1189.995 C3571.383,1189.995 3574.705,1194.038 3573.28,1199.025
        </g>
        <path d="M3533.003,1199.995 L3553.003,1199.995 C3558.526,1199.995 3561.848,1204.038 3560.423,1209.025
      </g>
      <g>
        <g>
          <path d="M3515.86,1249.995 L3555.86,1249.995 C3561.383,1249.995 3564.705,1254.038 3563.28,1259.026
          <path d="M3515.86,1249.995 L3555.86,1249.995 C3561.383,1249.995 3564.705,1254.038 3563.28,1259.026
        </g>
        <path d="M3523.003,1259.995 L3543.003,1259.995 C3548.526,1259.995 3551.848,1264.038 3550.423,1269.025
      </g>
      <g>
        <g>
          <path d="M3555.86,1219.995 L3595.86,1219.995 C3601.383,1219.995 3604.705,1224.038 3603.28,1229.025
          <path d="M3555.86,1219.995 L3595.86,1219.995 C3601.383,1219.995 3604.705,1224.038 3603.28,1229.025
        </g>
        <path d="M3563.003,1229.995 L3583.003,1229.995 C3588.526,1229.995 3591.848,1234.038 3590.423,1239.025
      </g>
    </g>
  </g>
</svg>
```

inside of cp8.zip: favicon-16x16.png (808 bytes)

inside of cp8.zip: favicon-32x32.png (1418 bytes)

inside of cp8.zip: index.html (1908 bytes)

```
<!DOCTYPE html>
<!--
Name: Jack Venberg
Date: 05.14.18
Section: CSE 154 AH

This is the index.html page for my Creative Project in which I have a Uno game
that connects to a custom-built Uno API. This page contains the two players
hands and the two card piles, as well as, information about each play.
-->

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Uno</title>
    <link rel="icon" type="image/png" sizes="32x32" href="favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="favicon-16x16.png">
    <link rel="stylesheet" type="text/css" href="main.css">
    <link rel="stylesheet" type="text/css" href="uno.css">
    <script src="main.js"></script>
    <script src="uno.js"></script>
  </head>
  <body>
    <div class="background-top"></div>
    <main>
      <article id="table">
        <h1>UNO Card Game</h1>
        <h2>Opponents Hand</h2>
        <section id="opponent" class="hand"></section>

        <h2>Results</h2>
        <section id="results"></section>

        <section id="deck" class="card-deck">
          <div id="discard" class="uno blank hidden"></div>
          <div id="pile" class="uno back hidden"></div>
          <div id="topPile" class="uno back hidden"></div>
        </section>

        <section id="start-new" class="hidden">
          <h2>Start New Game</h2>
        </section>

        <section id="color-picker" class="hidden">
          <h3>Pick a Color</h3>
          <div>
            <div class="color" id="red"></div>
            <div class="color" id="yellow"></div>
            <div class="color" id="green"></div>
            <div class="color" id="blue"></div>
          </div>
        </section>

        <h2>Your Hand</h2>
        <section id="player" class="hand"></section>
      </article>
      <footer>
        <p>Created by Jack Venberg &copy; 2018</p>
      </footer>
    </main>
  </body>
</html>
```

inside of cp8.zip: main.css (1454 bytes)

```
/*
Name: Jack Venberg
Date: 05.14.18
Section: CSE 154 AH

This is the main.css style for my Creative Project in which I have a Uno game
that connects to a custom-built uno API. This style is applied to all pages.
```

```
*/

* {
  margin: 0;
}

html, body, button {
  font-family: helvetica, sans-serif;
  padding: 0;
  height: 100%;
  background-color: #E0E0E0;
}

header {
  margin-bottom: 10px;
}

h1 {
  font-size: 50px;
}

html, body, a, .nav-change > .active, .nav-change a:hover {
  text-decoration: none;
  color: #212121;
}

footer, nav, .background-top {
  width: 100%;
}

footer {
  background: #1976D2;
  line-height: 80px;
}

footer, nav a {
  text-align: center;
}

nav {
  z-index: 1;
  display: flex;
}

nav, .card {
  box-shadow: 1px 2px 5px rgba(0, 0, 0, 0.2);
}

nav a {
  padding: 20px 30px;
  font-size: 1.1em;
}

nav {
  transition: background-color 0.5s ease;
}

nav a:hover, .active {
  background-color: #E53935;
}

nav, .background-top {
  position: fixed;
}

.grow {
  transition: all .3s ease-in-out;
}

.grow:hover {
  transform: scale(1.1);
}

.nav-change {
  background-color: #D32F2F;
}

nav, .nav-change a:hover, .nav-change .active, main {
  background-color: #FAFAFA;
}
```

```
footer, .active, nav a:hover, .nav-change a {
  color: white;
}

.background-top {
  background-color: #EF5350;
  height: 55%;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
  top: 0;
}

main {
  position: absolute;
  top: 100px;
  width: 70%;
  right: 15%;
}
```

inside of cp8.zip: main.js (939 bytes)

```
/*
Name: Jack Venberg
Date: 05.14.18
Section: CSE 154 AH

This is the main.js script for my Creative Project in which I have a Uno game
that connects to a custom-built Uno API. This script contains universal
functions between all pages.
*/

"use strict";
(function() {

  /** Called when pages scrolls. Changes nav color scheme. */
  window.onscroll = window.changeNavOnScroll;

  /**
   * Returns the element that has the ID attribute with the specified value.
   * @param {string} id - element ID
   * @return {object} DOM object associated with id.
   */
  window.$ = function(id) {
    return document.getElementById(id);
  };

  /**
   * Changes the color scheme of the nav bar when the nav bar crosses the card of
   * the page.
   */
  window.changeNavOnScroll = function() {
    if (window.pageYOffset > 120) {
      $('nav').classList.add("nav-change");
    } else {
      $('nav').classList.remove("nav-change");
    }
  };
})();
```

inside of cp8.zip: setup.sql (216 bytes)

```
DROP DATABASE IF EXISTS c9;
CREATE DATABASE c9;
USE c9;

DROP TABLE IF EXISTS games;

CREATE TABLE games (
    guid CHAR(13) PRIMARY KEY,
    playerHand TEXT,
    opponentHand TEXT,
    deck TEXT,
    discard TEXT
);
```

inside of cp8.zip: uno.css (2864 bytes)

```
/*
Name: Jack Venberg
```

```
Date: 05.14.18
Section: CSE 154 AH

This is the main.css style for my Creative Project in which I have a Uno game
that connects to a custom-built Uno API. This script contains styling
information for the index.html page.
*/

main > article {
  padding-bottom: 10px;
  height: 70%;
}

h1, h2, p, #color-picker {
  text-align: center;
}

#color-picker .color {
  height: 40px;
  width: 40px;
  margin: 5px;
}

#color-picker .color, #start-new {
  cursor: pointer;
}

#color-picker > div, .hand, .card-deck {
  display: flex;
  justify-content: center;
}

#red, #start-new {
  background-color: #F55;
}

#yellow {
  background-color: #FA0;
}

#green {
  background-color: #5A5;
}

#blue {
  background-color: #55F;
}

#start-new {
  box-shadow: 2px 2px 2px black;
  border-radius: 10px;
  margin: 10px auto;
  width: 200px;
}

#start-new:active {
  box-shadow: inset 1px 1px 2px black;
}

#start-new, h1 {
  color: white;
}

h1 {
  font-size: 40px;
  margin-bottom: 20px;
  background-color: #1976D2;
  padding: 10px 0;
}

h2, button {
  font-size: 30px;
}

button {
  position: relative;
  margin: 0 auto;
}

.uno, #topPile {
  transition: top 0.3s, left 0.3s;
```

```
}

.uno {
  width: 242px;
  height: 362px;
  transform: scale(0.3);
  margin: calc(-363px * (1 - 0.3) / 2 + 5px) calc(((-243px * (1 - 0.3)) / 2) + 5px);
  background-image: url('UNO_deck.svg');
  background-repeat: no-repeat;
  background-size: 3644px 1814px;
}

.blank {
  visibility: hidden;
}

#topPile {
  right: 100px;
  position: absolute;
}

.hand, .card-deck {
  padding: 10px;
  flex-wrap: wrap;
  min-height: 110px;
}

.hidden {
  display: none;
}

.num0 {
  background-position-x: calc(-243px * 0);
}

.num1 {
  background-position-x: calc(-243px * 1);
}

.num2 {
  background-position-x: calc(-243px * 2);
}

.num3 {
  background-position-x: calc(-243px * 3);
}

.num4 {
  background-position-x: calc(-243px * 4);
}

.num5 {
  background-position-x: calc(-243px * 5);
}

.num6 {
  background-position-x: calc(-243px * 6);
}

.num7 {
  background-position-x: calc(-243px * 7);
}

.num8 {
  background-position-x: calc(-243px * 8);
}

.num9 {
  background-position-x: calc(-243px * 9);
}

.skip {
  background-position-x: calc(-243px * 10);
}

.reverse {
  background-position-x: calc(-243px * 11);
}

.draw2 {
  background-position-x: calc(-243px * 12);
```

```
}

.wild {
  background-position-y: calc(-363px * 4);
  background-position-x: calc(-243px * 13);
}

.wildDraw4 {
  background-position-y: calc(-363px * 4);
  background-position-x: calc(-243px * 14);
}

.back {
  background-position-y: calc(-363px * 4);
  background-position-x: calc(-243px * 12);
}

.red {
  background-position-y: calc(-363px * 0);
}

.yellow {
  background-position-y: calc(-363px * 1);
}

.green {
  background-position-y: calc(-363px * 2);
}

.blue {
  background-position-y: calc(-363px * 3);
}
```

inside of cp8.zip: uno.js (9990 bytes)

```
/*
Name: Jack Venberg
Date: 05.14.18
Section: CSE 154 AH

This is the main.js script for my Creative Project in which I have a Uno game
that connects to a custom-built uno API. This script runs on index.html and
connects, makes calls to, and displays from the Uno API.
*/

"use strict";
(function() {
  const GAME_URL = "uno.php";

  let guid; /* Game ID */
  let animations; /* Array of animation to be played */
  let canPlay; /* Boolean signifying whether or not player can play */
  let currentWildMove; /* Currently wild card move */
  let currentGameData; /* Current game data of whole game */
  let handSnapshots = []; /* Still snapshot of each hand */

  /* Runs on page load. Adds onclick functionality and starts game. */
  window.onload = function() {
    $("pile").onclick = function() {
      playMove("new");
    };
    let colors = document.getElementsByClassName("color");
    for (let i = 0; i < colors.length; i++) {
      colors[i].onclick = wildMove;
    }
    $("topPile").addEventListener("transitionend", function(event) {
      if (event.propertyName === "top") {
        this.classList.add("hidden");
        this.style.top = null;
        this.style.left = null;
        $("new-card").classList.remove("blank");
        $("new-card").id = "";
        animateContinue();
      }
    });
    $("start-new").onclick = startGame;
    startGame();
  };

  /* Starts game by fetching from API and displays all cards */
```

```
function startGame() {
  canPlay = true;
  $("start-new").classList.add("hidden");
  guid = undefined;
  let data =  new FormData();
  data.append("startgame", true);

  fetch(GAME_URL, {credentials: 'include', method: "POST", body: data})
    .then(checkStatus)
    .then(tryJSONParse)
    .then(function(gameData) {
      displayTable(gameData);
      $("discard").classList.remove("hidden");
      $("pile").classList.remove("hidden");
    })
    .catch(console.log);
}

/**
 * Attempts to parse JSON response, otherwise prints to console the error
 * @param {object} - Promise response.
 * @return {object} - Parsed JSON object.
 */
function tryJSONParse(response) {
  try {
    return JSON.parse(response);
  } catch (e) {
    console.log(response);
    return Promise.reject(new Error(response.status +
                                    ": " + response.statusText));
  }
}

/**
 * Starts running animations from global animations array.
 * @param {gameData} - JSON object containing all game data.
 */
function runAnimations(gameData) {
  canPlay = false;
  handSnapshots[0] = Array.from(document.querySelectorAll("#player .uno"));
  handSnapshots[1] = Array.from(document.querySelectorAll("#opponent .uno"));
  animations = gameData.animations;
  currentGameData = gameData;
  animateContinue();
}

/* Runs single animation from global animations array. */
function animateContinue() {
  if (animations.length > 0) {
    let animation = animations.shift();
    if (animation.moveType === "play") {
      let deck = animation.isPlayer ? handSnapshots[0] : handSnapshots[1];
      playToDiscard(deck[animation.cardIndex]);
    } else if (animation.moveType === "draw") {
      let deck = animation.isPlayer ? $("player") : $("opponent");
      drawCard(deck);
    } else if (animation.moveType === "won" || animation.moveType === "lost") {
      canPlay = false;
      $("start-new").classList.remove("hidden");
      displayTable(currentGameData);
    } else {
      animateContinue();
    }
  } else {
    displayTable(currentGameData);
    canPlay = true;
  }
}

/**
 * Animates card moving to discard pile.
 * @param {object} card - Card DOM object to animate.
 */
function playToDiscard(card) {
  card.addEventListener("transitionend", function(event) {
    if (event.propertyName === "top") {
      card.parentNode.removeChild(card);
      $("discard").className = card.className;
      animateContinue();
    }
  });
```

```
  flyTo(card, $("discard"));
}

/**
 * Animates drawing card from deck to players hand.
 * @param {object} deck - Deck DOM object.
 */
function drawCard(deck) {
  let newCard = document.createElement("div");
  newCard.id = "new-card";
  newCard.classList.add("blank");
  newCard.classList.add("uno");
  newCard.classList.add("back");
  deck.appendChild(newCard);
  if (deck.id === "player") {
    handSnapshots[0].push(newCard);
  } else {
    handSnapshots[1].push(newCard);
  }

  let targetOffset = getOffset($("pile"));
  $("topPile").style.right = targetOffset.right + "px";
  $("topPile").classList.remove("hidden");
  flyTo($("topPile"), newCard);
}

/**
 * Animates a card object flying to another card.
 * @param {object} card - DOM object to animate.
 * @param {object} target - DOM object to fly to.
 */
function flyTo(card, target) {
  card.style.zIndex = 1;
  let cardOffset = getOffset(card);
  card.style.position = "absolute";
  card.style.top = cardOffset.top + "px";
  card.style.left = cardOffset.left + "px";

  let targetOffset = getOffset(target);
  card.style.position = "absolute";
  card.style.top = targetOffset.top + "px";
  card.style.left = targetOffset.left + "px";
}

/**
 * Returns the offset of a card from its parent container.
 * @param {object} card - DOM object to get offset.
 * @return {object} JSON object containing offsets.
 */
function getOffset(card) {
  let childPos = card.getBoundingClientRect();
  let parentPos = $("table").getBoundingClientRect();
  return {
      top: childPos.top - parentPos.top,
      left: childPos.left - parentPos.left - 5,
      right: parentPos.right - childPos.right + 5
  };
}

/**
 * Displays all card hands and piles on table based on given game data.
 * @param {object} gameData - JSON object containing game data.
 */
function displayTable(gameData) {
  if (!guid) {
    guid = gameData.guid;
  }
  $("results").innerHTML = gameData.results;
  displayCards($("player"), gameData.playerHand);
  displayCards($("opponent"), gameData.opponentHand);
  displayDiscard(gameData.discard);
}

/**
 * Displays the discard based on given discard JSON object.
 * @param {object} discard - JSON object of discard to display.
 */
function displayDiscard(discard) {
  if (discard) {
    $("discard").className = "";
    $("discard").classList.add("uno");
```

```
    $("discard").classList.add(discard.type);
    if (discard.color) {
      $("discard").classList.add(discard.color);
    }
  } else {
    $("discard").classList.add("blank");
  }
}

/**
 * Displays cards in given hand.
 * @param {object} hand - hand DOM object.
 * @param {array} cards - Array of card objects.
 */
function displayCards(hand, cards) {
  while (hand.lastChild) {
    hand.removeChild(hand.lastChild);
  }
  for (let i = 0; i < cards.length; i++) {
    createCard(hand, cards[i]);
  }
}

/**
 * Creates a given card in hand.
 * @param {object} hand - hand DOM object.
 * @param {object} cardData - JSON data of card.
 */
function createCard(hand, cardData) {
    let card = document.createElement("div");
    card.classList.add("uno");
    if (cardData.color) {
      card.classList.add(cardData.color);
    }
    card.classList.add(cardData.type);
    if (hand.id === "player") {
      if (cardData.type === "wild" || cardData.type === "wildDraw4") {
        card.onclick = wildCheckPlayable;
      } else {
        card.onclick = function() {
          playMove(getIndex(this));
        };
      }
    }
    hand.appendChild(card);
}

/* Checks if a wild card is playable */
function wildCheckPlayable() {
  if (canPlay) {
    currentWildMove = getIndex(this);
    let data = new FormData();
    data.append("guid", guid);
    data.append("move", currentWildMove);
    data.append("checkplayable", true);

    fetch(GAME_URL, {credentials: 'include', method: "POST", body: data})
      .then(checkStatus)
      .then(tryJSONParse)
      .then(wildChooseColor)
      .catch(console.log);
  } else {
    animateContinue();
  }
}

/**
 * Shows wild card color chooser
 * @param {object} gameData - JSON object containing game data.
 */
function wildChooseColor(gameData) {
  if (gameData.playable) {
    $("color-picker").classList.remove("hidden");
  }
  $("results").innerHTML = gameData.results;
}

/* Plays wild card. */
function wildMove() {
  playMove(currentWildMove, this.id);
  $("color-picker").classList.add("hidden");
```

```
  }

  /**
   * Fetches given card move with given move index and color.
   * @param {integer} move - card index.
   * @param {string} color - card color.
   */
  function playMove(move, color) {
    if (canPlay) {
      let data = new FormData();
      data.append("guid", guid);
      data.append("move", move);
      if (color) {
        data.append("color", color);
      }
      fetch(GAME_URL, {credentials: 'include', method: "POST", body: data})
        .then(checkStatus)
        .then(tryJSONParse)
        .then(runAnimations)
        .catch(console.log);
    } else {
      animateContinue();
    }
  }

  /* Returns index of element in parent */
  function getIndex(element) {
    return Array.from(element.parentNode.children).indexOf(element);
  }

  /**
  * Checks the status response code.
  * @param {object} response - fetch response object.
  * @return {object} Returns a succesful promise if status is successful, and
  * returns a rejected promise otherwise.
  */
  function checkStatus(response) {
    if (response.status >= 200 && response.status < 300) {
      return response.text();
    } else {
      return Promise.reject(new Error(response.status +
                                      ": " + response.statusText));
    }
  }
})();
```

inside of cp8.zip: uno.php (14728 bytes)

```
<?php
  /*
  Name: Jack Venberg
  Date: 05.14.18
  Section: CSE 154 AH

  This is the uno.php script for my Creative Project in which I have a Uno game
  that connects to a custom-built uno API. This script runs the Uno API on
  cloud9 servers and handles all Uno game logic through POST calls.
  */

  error_reporting(E_ALL);
  /* All card types, colors, and names */
  $CARD_TYPE = ["num0", "num1", "num2", "num3", "num4", "num5", "num6", "num7", "num8", "num9",
                "skip", "reverse", "draw2", "wild", "wildDraw4"];
  $CARD_NAMES = ["Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine",
                 "Skip", "Reverse", "Draw Two", "Wild Card", "Wild Draw Four"];
  $CARD_COLORS = ["red", "yellow", "green", "blue"];
  $skip = false; /* Whether or not currently skipping opponent */

  /* Database connection variables */
  $servername = "localhost";
  $username = "root";
  $password = "";
  $database = "c9";

  /* Make a data source string that will be used in creating the PDO object */
  $ds = "mysql:host={$servername};dbname={$database};charset=utf8";

  try {
    $db = new PDO($ds, $username, $password);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
} catch (PDOException $ex) {
  handle_error("Can not connect to the database. Please try again later.", $ex);
}

header("Content-Type: application/json");
$animations = array(); /* Array for storing animation objects to be sent back */
if (isset($_POST["startgame"]) and $_POST["startgame"]) {
  start_game();
} else if (isset($_POST["guid"]) and isset($_POST["move"])) {
  if (isset($_POST["checkplayable"])) {
    check_if_playable();
  } else {
    play_move();
  }
}

$db = null;

/* Starts game by setting up database and displaying initial values. */
function start_game() {
  $GLOBALS["guid"] = uniqid();
  $GLOBALS["deck"] = create_deck();

  shuffle($GLOBALS["deck"]);
  list($GLOBALS["players_hand"], $GLOBALS["opponents_hand"]) = deal_hands($GLOBALS["deck"]);

  $game_data = array();
  $game_data["guid"] = $GLOBALS["guid"];
  $game_data["playerHand"] = $GLOBALS["players_hand"];
  $game_data["opponentHand"] = hide_cards($GLOBALS["opponents_hand"]);
  $game_data["results"] = "<p>Created new deck of cards.</p>";
  $game_data["animations"] = array();
  array_push($game_data["animations"], new Animation("new_deck", null, null));
  print(json_encode($game_data));

  $stmt = $GLOBALS["db"]->prepare("INSERT INTO games (guid, playerHand, opponentHand, deck)
                    VALUES (:guid, :player_hand, :opponent_hand, :deck)");
  $params = array("guid" => $GLOBALS["guid"],
                  "player_hand" => serialize($GLOBALS["players_hand"]),
                  "opponent_hand" => serialize($GLOBALS["opponents_hand"]),
                  "deck" => serialize($GLOBALS["deck"]));
  $stmt->execute($params);
}

/**
 * Plays a move of a card depending on what "move" is set to in POST
 * request.
 */
function play_move() {
  read_db();

  $game_data = array();
  $move = $_POST["move"];
  if (is_numeric($move) and $move < count($GLOBALS["players_hand"]) and $move >= 0) {
    if (is_playable($GLOBALS["players_hand"][$move], $GLOBALS["players_hand"])) {
      $game_data["results"] = player_play($move);

      if (count($GLOBALS["players_hand"]) == 0) {
        $game_data["results"] .= "<p>YOU WON!</p>";
        array_push($GLOBALS["animations"], new Animation("won", null, null));
      } else {
        $game_data["results"] .= opponent_play();
        if (count($GLOBALS["opponents_hand"]) == 0) {
          $game_data["results"] .= "<p>YOU LOST!</p>";
          array_push($GLOBALS["animations"], new Animation("lost", null, null));
        }
      }
    } else {
      $game_data["results"] = "<p>Card is unplayable.</p>";
    }
  } else if ($move == "new") {
    check_deck();
    array_push($GLOBALS["players_hand"], array_pop($GLOBALS["deck"]));
    array_push($GLOBALS["animations"], new Animation("draw", null, true));
    $game_data["results"] = "<p>You drew a card.</p>";
  }

  $game_data["guid"] = $GLOBALS["guid"];
  $game_data["playerHand"] = $GLOBALS["players_hand"];
  $game_data["opponentHand"] = hide_cards($GLOBALS["opponents_hand"]);
  $game_data["discard"] = end($GLOBALS["discard"]);
```

```
  $game_data["animations"] = $GLOBALS["animations"];
  print(json_encode($game_data));

  write_db();
}

/**
 * Checks if a card specified by what "move" is set to in POST request.
 * Returns true if playable and false otherwise.
 * @return {boolean} Whether or not the move is playable.
 */
function check_if_playable() {
  read_db();

  $game_data = array();
  $game_data["guid"] = $GLOBALS["guid"];
  if (is_playable($GLOBALS["players_hand"][$_POST["move"]], $GLOBALS["players_hand"])) {
    $game_data["playable"] = true;
    $game_data["results"] = "<p>Please choose a color.</p>";
  } else {
    $game_data["playable"] = false;
    $game_data["results"] = "<p>Card is unplayable.</p>";
  }

  print(json_encode($game_data));

  write_db();
}

/**
 * Plays a move by the player based on the given $move index.
 * @param {integer} $move - Index of card to play.
 */
function player_play($move) {
  $played_card = play_card($move, $GLOBALS["players_hand"], $GLOBALS["opponents_hand"], true);
  return "<p>You played a " . $played_card->name . ".</p>";
}

/* Plays a move for the player randomly choosing a playable card. */
function opponent_play() {
  if (!$GLOBALS["skip"]) {
    $drew_count = 0;
    while(true) {
      $playable_moves = array();
      foreach ($GLOBALS["opponents_hand"] as $move=>$card) {
        if (is_playable($card, $GLOBALS["opponents_hand"])) {
          array_push($playable_moves, $move);
        }
      }
      if (count($playable_moves) == 0) {
        check_deck();
        array_push($GLOBALS["opponents_hand"], array_pop($GLOBALS["deck"]));
        array_push($GLOBALS["animations"], new Animation("draw", null, false));
        $drew_count++;
      } else {
        $move = $playable_moves[array_rand($playable_moves)];
        $played_card = play_card($move, $GLOBALS["opponents_hand"], $GLOBALS["players_hand"], false);
        $result = "<p>Opponent ";
        if ($drew_count > 0) {
          $result .= "drew " . $drew_count . " times and " ;
        }
        $result .= "played a " . $played_card->name . ".</p>";
        if ($played_card->type == "skip" || $played_card->type == "reverse") {
          $result .= "<p>Player was skipped</p>";
          $GLOBALS["skip"] = false;
          $result .= opponent_play();
        }
        return $result;
      }
    }
  } else {
    $GLOBALS["skip"] = false;
    return "<p>Opponent was skipped</p>";
  }
}

/**
 * Plays a specific card speficied by the "$move" index. Takes card from given
 * "$hand" and will draw cards to "$otherHand".
 * @param {integer} $move - Index of card to play.
 * @param {array} $hand - Array of Card objects of hand to play from.
```

```
 * @param {array} $other_hand - Array of Card objects to draw to.
 * @param {boolean} $is_player - Whether of not move is for player.
 */
function play_card($move, &$hand, &$other_hand, $is_player) {
  $played_card = $hand[$move];
  $played_card->index = $move;
  array_push($GLOBALS["animations"], new Animation("play", $move, $is_player));
  if ($played_card->type == "wild" or $played_card->type == "wildDraw4") {
    if ($is_player) {
      $played_card->color = $_POST["color"];
    } else {
      $played_card->color = $GLOBALS["CARD_COLORS"][array_rand($GLOBALS["CARD_COLORS"])];
    }
  }
  if ($played_card->type == "wildDraw4") {
    for ($i = 0; $i < 4; $i++) {
      check_deck();
      array_push($other_hand, array_pop($GLOBALS["deck"]));
      array_push($GLOBALS["animations"], new Animation("draw", null, !$is_player));
    }
  } else if ($played_card->type == "draw2") {
    for ($i = 0; $i < 2; $i++) {
      check_deck();
      array_push($other_hand, array_pop($GLOBALS["deck"]));
      array_push($GLOBALS["animations"], new Animation("draw", null, !$is_player));
    }
  } else if ($played_card->type == "skip" or $played_card->type == "reverse") {
    $GLOBALS["skip"] = true;
  }
  array_push($GLOBALS["discard"], $played_card);
  unset($hand[$move]);
  $hand = array_values($hand);
  return $played_card;
}

/**
 * Reads from mySQL database based on what "guid" is set to in POST request
 * and stores data into global variables.
 */
function read_db() {
  $stmt = $GLOBALS["db"]->prepare("SELECT * FROM games WHERE guid=:guid");
  $params = array("guid" => $_POST["guid"]);
  $stmt->execute($params);
  $row = $stmt->fetch();

  $discard = isset($row["discard"]) ? unserialize($row["discard"]) : array();
  $GLOBALS["guid"] = $row["guid"];
  $GLOBALS["deck"] = unserialize($row["deck"]);
  $GLOBALS["players_hand"] = unserialize($row["playerHand"]);
  $GLOBALS["opponents_hand"] = unserialize($row["opponentHand"]);
  $GLOBALS["discard"] = $discard;

}

/* Writes to mySQL database based global table variables. */
function write_db() {
  $stmt = $GLOBALS["db"]->prepare("UPDATE games SET playerHand=:playerHand,
                                   opponentHand=:opponentHand, deck=:deck, discard=:discard
                                   WHERE guid=:guid");
  $params = array("guid" => $GLOBALS["guid"],
                  "playerHand" => serialize($GLOBALS["players_hand"]),
                  "opponentHand" => serialize($GLOBALS["opponents_hand"]),
                  "deck" => serialize($GLOBALS["deck"]),
                  "discard" => serialize($GLOBALS["discard"]));
  $stmt->execute($params);
}

/**
 * Checks to make sure deck has hards. If not, it takes discard pile and puts
 * discarded cards into deck.
 */
function check_deck() {
  if (count($GLOBALS["deck"]) == 0) {
    $GLOBALS["deck"] = $GLOBALS["discard"];
    $GLOBALS["discard"] = array();
  }
}

/**
 * Checks if a given card is playable from a given hand. Wild draw 4 is
 * playable only if no other cards are playable.
```

```
 * @param {object} $card - Card object that's checked for playablility.
 * @param {array} $hand - Array of Card objects to check from.
 * @return {boolean} Whether or not card is playable. True if yes, no otherwise.
 */
function is_playable($card, $hand) {
  if ($card->type == "wildDraw4") {
    foreach($hand as $card_in_hand) {
      if ($card_in_hand->type != "wildDraw4" and is_playable_helper($card_in_hand)) {
        return false;
      }
    }
    return true;
  }
  return is_playable_helper($card);
}

/**
 * Checks if a given card is playable.
 * @param {object} $card - Card object that's checked for playablility.
 * @return {boolean} Whether or not card is playable. True if yes, no otherwise.
 */
function is_playable_helper($card) {
  if (!$GLOBALS["discard"]) {
    return true;
  }
  if ($card->type == "wild") {
    return true;
  }
  $top_discard = end($GLOBALS["discard"]);
  if ($card->type == $top_discard->type or
      ($card->color == $top_discard->color and $card->color != null)) {
    return true;
  }
  return false;
}

/**
 * Prints the given error message with details about the error.
 * @param {string} $msg - Error message.
 * @param {object} $ex - Error object.
 */
function handle_error($msg, $ex) {
  header("Content-Type: text/plain");
  print ("{$msg}\n");
  print ("Error details: $ex \n");
  die();
}

/**
 * Obscures given hand by returning a hand with an equal number of back-facing
 * cards.
 * @param {array} $hand - Array of Card objects for hand.
 * @return {array} Array of obscured Card objects.
 */
function hide_cards($hand) {
  $hidden_hand = array();
  for ($i = 0; $i < count($hand); $i++) {
    array_push($hidden_hand, new Card("back", null, "Back"));
  }
  return $hidden_hand;
}

/**
 * Deals equally to each hand from given deck.
 * @param {array} $deck - Array of Card objects for deck.
 * @return {array} Array of two arrays of Card objects for each hand.
 */
function deal_hands(&$deck) {
  $deal_player = true;
  $players_hand = array();
  $opponents_hand = array();
  for ($i = 0; $i < 14; $i++) {
    if ($deal_player) {
      array_push($players_hand, array_pop($deck));
    } else {
      array_push($opponents_hand, array_pop($deck));
    }
    $deal_player = !$deal_player;
  }
  return [$players_hand, $opponents_hand];
}
```

```
  /**
   * Creates a deck of uno cards.
   * return {array} Array of Card objects representing the deck.
   */
  function create_deck() {
    $deck = array();
    for ($i = 0; $i < 15; $i++) {
      for ($j = 0; $j < 4; $j++) {
        if ($i < 13) {
          $name = ucfirst($GLOBALS["CARD_COLORS"][$j]) . " " . $GLOBALS["CARD_NAMES"][$i];
          if ($i > 0) {
            array_push($deck, new Card($GLOBALS["CARD_TYPE"][$i], $GLOBALS["CARD_COLORS"][$j], $name));
          }
          array_push($deck, new Card($GLOBALS["CARD_TYPE"][$i], $GLOBALS["CARD_COLORS"][$j], $name));
        } else {
          array_push($deck, new Card($GLOBALS["CARD_TYPE"][$i], null, $GLOBALS["CARD_NAMES"][$i]));
        }
      }
    }
    return $deck;
  }

  /* Card object for storing card information. */
  class Card {
    public $type;
    public $color;
    public $name;
    public $index;

    /**
     * Constructs Card object.
     * @param {string} $type - Card type.
     * @param {string} $color - Card color.
     * @param {string} $name - Card name.
     */
    public function __construct($type, $color, $name) {
      $this->type = $type;
      $this->color = $color;
      $this->name = $name;
    }
  }

  /* Card object for storing animation information. */
  class Animation {
    public $moveType;
    public $cardIndex;
    public $isPlayer;

    /**
     * Constructs Card object.
     * @param {string} $move_type - Animation type.
     * @param {integer} $card_index - Index of card in hand to move.
     * @param {boolean} $is_player - Whether or not animation is for the player.
     */
    public function __construct($move_type, $card_index, $is_player) {
      $this->moveType = $move_type;
      $this->cardIndex = (int)$card_index;
      $this->isPlayer = (bool)$is_player;
    }
  }
?>
```

favicon-16x16.png (808 bytes)

(binary file)

favicon-32x32.png (1418 bytes)

(binary file)

index.html (1908 bytes)

```
1. <!DOCTYPE html>
2. <!--
3. Name: Jack Venberg
```

```
4. Date: 05.14.18
5. Section: CSE 154 AH
6.
7. This is the index.html page for my Creative Project in which I have a Uno game
8. that connects to a custom-built Uno API. This page contains the two players
9. hands and the two card piles, as well as, information about each play.
10. -->
11.
12. <html lang="en">
13.   <head>
14.     <meta charset="UTF-8">
15.     <title>Uno</title>
16.     <link rel="icon" type="image/png" sizes="32x32" href="favicon-32x32.png">
17.     <link rel="icon" type="image/png" sizes="16x16" href="favicon-16x16.png">
18.     <link rel="stylesheet" type="text/css" href="main.css">
19.     <link rel="stylesheet" type="text/css" href="uno.css">
20.     <script src="main.js"></script>
21.     <script src="uno.js"></script>
22.   </head>
23.   <body>
24.     <div class="background-top"></div>
25.     <main>
26.       <article id="table">
27.         <h1>UNO Card Game</h1>
28.         <h2>Opponents Hand</h2>
29.         <section id="opponent" class="hand"></section>
30.
31.         <h2>Results</h2>
32.         <section id="results"></section>
33.
34.         <section id="deck" class="card-deck">
35.           <div id="discard" class="uno blank hidden"></div>
36.           <div id="pile" class="uno back hidden"></div>
37.           <div id="topPile" class="uno back hidden"></div>
38.         </section>
39.
40.         <section id="start-new" class="hidden">
41.           <h2>Start New Game</h2>
42.         </section>
43.
44.         <section id="color-picker" class="hidden">
45.           <h3>Pick a Color</h3>
46.           <div>
47.             <div class="color" id="red"></div>
48.             <div class="color" id="yellow"></div>
49.             <div class="color" id="green"></div>
50.             <div class="color" id="blue"></div>
51.           </div>
52.         </section>
53.
54.         <h2>Your Hand</h2>
55.         <section id="player" class="hand"></section>
56.       </article>
57.       <footer>
58.         <p>Created by Jack Venberg &copy; 2018</p>
59.       </footer>
60.     </main>
61.   </body>
62. </html>
```

main.css (1454 bytes)

```
1. /*
2. Name: Jack Venberg
3. Date: 05.14.18
4. Section: CSE 154 AH
5.
6. This is the main.css style for my Creative Project in which I have a Uno game
7. that connects to a custom-built uno API. This style is applied to all pages.
8. */
9.
10. * {
11.   margin: 0;
12. }
13.
14. html, body, button {
15.   font-family: helvetica, sans-serif;
16.   padding: 0;
```

```
  height: 100%;
  background-color: #E0E0E0;
}

header {
  margin-bottom: 10px;
}

h1 {
  font-size: 50px;
}

html, body, a, .nav-change > .active, .nav-change a:hover {
  text-decoration: none;
  color: #212121;
}

footer, nav, .background-top {
  width: 100%;
}

footer {
  background: #1976D2;
  line-height: 80px;
}

footer, nav a {
  text-align: center;
}

nav {
  z-index: 1;
  display: flex;
}

nav, .card {
  box-shadow: 1px 2px 5px rgba(0, 0, 0, 0.2);
}

nav a {
  padding: 20px 30px;
  font-size: 1.1em;
}

nav {
  transition: background-color 0.5s ease;
}

nav a:hover, .active {
  background-color: #E53935;
}

nav, .background-top {
  position: fixed;
}

.grow {
  transition: all .3s ease-in-out;
}

.grow:hover {
  transform: scale(1.1);
}

.nav-change {
  background-color: #D32F2F;
}

nav, .nav-change a:hover, .nav-change .active, main {
  background-color: #FAFAFA;
}

footer, .active, nav a:hover, .nav-change a {
  color: white;
}

.background-top {
  background-color: #EF5350;
  height: 55%;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
  top: 0;
```

```
98. }
99.
100. main {
101.   position: absolute;
102.   top: 100px;
103.   width: 70%;
104.   right: 15%;
105. }
```

main.js (939 bytes)

```
1. /*
2. Name: Jack Venberg
3. Date: 05.14.18
4. Section: CSE 154 AH
5.
6. This is the main.js script for my Creative Project in which I have a Uno game
7. that connects to a custom-built Uno API. This script contains universal
8. functions between all pages.
9. */
10.
11. "use strict";
12. (function() {
13.
14.   /** Called when pages scrolls. Changes nav color scheme. */
15.   window.onscroll = window.changeNavOnScroll;
16.
17.   /**
18.    * Returns the element that has the ID attribute with the specified value.
19.    * @param {string} id - element ID
20.    * @return {object} DOM object associated with id.
21.    */
22.   window.$ = function(id) {
23.     return document.getElementById(id);
24.   };
25.
26.   /**
27.    * Changes the color scheme of the nav bar when the nav bar crosses the card of
28.    * the page.
29.    */
30.   window.changeNavOnScroll = function() {
31.     if (window.pageYOffset > 120) {
32.       $('nav').classList.add("nav-change");
33.     } else {
34.       $('nav').classList.remove("nav-change");
35.     }
36.   };
37. })();
```

setup.sql (216 bytes)

```
1. DROP DATABASE IF EXISTS c9;
2. CREATE DATABASE c9;
3. USE c9;
4.
5. DROP TABLE IF EXISTS games;
6.
7. CREATE TABLE games (
8.     guid CHAR(13) PRIMARY KEY,
9.     playerHand TEXT,
10.     opponentHand TEXT,
11.     deck TEXT,
12.     discard TEXT
13. );
```

uno.css (2864 bytes)

```
1. /*
2. Name: Jack Venberg
3. Date: 05.14.18
```

```
 4. Section: CSE 154 AH
 5.
 6. This is the main.css style for my Creative Project in which I have a Uno game
 7. that connects to a custom-built Uno API. This script contains styling
 8. information for the index.html page.
 9. */
10.
11. main > article {
12.   padding-bottom: 10px;
13.   height: 70%;
14. }
15.
16. h1, h2, p, #color-picker {
17.   text-align: center;
18. }
19.
20. #color-picker .color {
21.   height: 40px;
22.   width: 40px;
23.   margin: 5px;
24. }
25.
26. #color-picker .color, #start-new {
27.   cursor: pointer;
28. }
29.
30. #color-picker > div, .hand, .card-deck {
31.   display: flex;
32.   justify-content: center;
33. }
34.
35. #red, #start-new {
36.   background-color: #F55;
37. }
38.
39. #yellow {
40.   background-color: #FA0;
41. }
42.
43. #green {
44.   background-color: #5A5;
45. }
46.
47. #blue {
48.   background-color: #55F;
49. }
50.
51. #start-new {
52.   box-shadow: 2px 2px 2px black;
53.   border-radius: 10px;
54.   margin: 10px auto;
55.   width: 200px;
56. }
57.
58. #start-new:active {
59.   box-shadow: inset 1px 1px 2px black;
60. }
61.
62. #start-new, h1 {
63.   color: white;
64. }
65.
66. h1 {
67.   font-size: 40px;
68.   margin-bottom: 20px;
69.   background-color: #1976D2;
70.   padding: 10px 0;
71. }
72.
73. h2, button {
74.   font-size: 30px;
75. }
76.
77. button {
78.   position: relative;
79.   margin: 0 auto;
80. }
81.
82. .uno, #topPile {
83.   transition: top 0.3s, left 0.3s;
84. }
```

```
85.
86. .uno {
87.   width: 242px;
88.   height: 362px;
89.   transform: scale(0.3);
90.   margin: calc(-363px * (1 - 0.3) / 2 + 5px) calc(((-243px * (1 - 0.3)) / 2) + 5px);
91.   background-image: url('UNO_deck.svg');
92.   background-repeat: no-repeat;
93.   background-size: 3644px 1814px;
94. }
95.
96. .blank {
97.   visibility: hidden;
98. }
99.
100. #topPile {
101.   right: 100px;
102.   position: absolute;
103. }
104.
105. .hand, .card-deck {
106.   padding: 10px;
107.   flex-wrap: wrap;
108.   min-height: 110px;
109. }
110.
111. .hidden {
112.   display: none;
113. }
114.
115. .num0 {
116.   background-position-x: calc(-243px * 0);
117. }
118.
119. .num1 {
120.   background-position-x: calc(-243px * 1);
121. }
122.
123. .num2 {
124.   background-position-x: calc(-243px * 2);
125. }
126.
127. .num3 {
128.   background-position-x: calc(-243px * 3);
129. }
130.
131. .num4 {
132.   background-position-x: calc(-243px * 4);
133. }
134.
135. .num5 {
136.   background-position-x: calc(-243px * 5);
137. }
138.
139. .num6 {
140.   background-position-x: calc(-243px * 6);
141. }
142.
143. .num7 {
144.   background-position-x: calc(-243px * 7);
145. }
146.
147. .num8 {
148.   background-position-x: calc(-243px * 8);
149. }
150.
151. .num9 {
152.   background-position-x: calc(-243px * 9);
153. }
154.
155. .skip {
156.   background-position-x: calc(-243px * 10);
157. }
158.
159. .reverse {
160.   background-position-x: calc(-243px * 11);
161. }
162.
163. .draw2 {
164.   background-position-x: calc(-243px * 12);
165. }
```

```
166.
167. .wild {
168.   background-position-y: calc(-363px * 4);
169.   background-position-x: calc(-243px * 13);
170. }
171.
172. .wildDraw4 {
173.   background-position-y: calc(-363px * 4);
174.   background-position-x: calc(-243px * 14);
175. }
176.
177. .back {
178.   background-position-y: calc(-363px * 4);
179.   background-position-x: calc(-243px * 12);
180. }
181.
182. .red {
183.   background-position-y: calc(-363px * 0);
184. }
185.
186. .yellow {
187.   background-position-y: calc(-363px * 1);
188. }
189.
190. .green {
191.   background-position-y: calc(-363px * 2);
192. }
193.
194. .blue {
195.   background-position-y: calc(-363px * 3);
196. }
```

uno.js (9990 bytes)

```
1. /*
2. Name: Jack Venberg
3. Date: 05.14.18
4. Section: CSE 154 AH
5.
6. This is the main.js script for my Creative Project in which I have a Uno game
7. that connects to a custom-built uno API. This script runs on index.html and
8. connects, makes calls to, and displays from the Uno API.
9. */
10.
11. "use strict";
12. (function() {
13.   const GAME_URL = "uno.php";
14.
15.   let guid; /* Game ID */
16.   let animations; /* Array of animation to be played */
17.   let canPlay; /* Boolean signifying whether or not player can play */
18.   let currentWildMove; /* Currently wild card move */
19.   let currentGameData; /* Current game data of whole game */
20.   let handSnapshots = []; /* Still snapshot of each hand */
21.
22.   /* Runs on page load. Adds onclick functionality and starts game. */
23.   window.onload = function() {
24.     $("pile").onclick = function() {
25.       playMove("new");
26.     };
27.     let colors = document.getElementsByClassName("color");
28.     for (let i = 0; i < colors.length; i++) {
29.       colors[i].onclick = wildMove;
30.     }
31.     $("topPile").addEventListener("transitionend", function(event) {
32.       if (event.propertyName === "top") {
33.         this.classList.add("hidden");
34.         this.style.top = null;
35.         this.style.left = null;
36.         $("new-card").classList.remove("blank");
37.         $("new-card").id = "";
38.         animateContinue();
39.       }
40.     });
41.     $("start-new").onclick = startGame;
42.     startGame();
43.   };
44.
```

```
45.   /* Starts game by fetching from API and displays all cards */
46.   function startGame() {
47.     canPlay = true;
48.     $("start-new").classList.add("hidden");
49.     guid = undefined;
50.     let data =  new FormData();
51.     data.append("startgame", true);
52.
53.     fetch(GAME_URL, {credentials: 'include', method: "POST", body: data})
54.       .then(checkStatus)
55.       .then(tryJSONParse)
56.       .then(function(gameData) {
57.         displayTable(gameData);
58.         $("discard").classList.remove("hidden");
59.         $("pile").classList.remove("hidden");
60.       })
61.       .catch(console.log);
62.   }
63.
64.   /**
65.    * Attempts to parse JSON response, otherwise prints to console the error
66.    * @param {object} - Promise response.
67.    * @return {object} - Parsed JSON object.
68.    */
69.   function tryJSONParse(response) {
70.     try {
71.       return JSON.parse(response);
72.     } catch (e) {
73.       console.log(response);
74.       return Promise.reject(new Error(response.status +
75.                                       ": " + response.statusText));
76.     }
77.   }
78.
79.   /**
80.    * Starts running animations from global animations array.
81.    * @param {gameData} - JSON object containing all game data.
82.    */
83.   function runAnimations(gameData) {
84.     canPlay = false;
85.     handSnapshots[0] = Array.from(document.querySelectorAll("#player .uno"));
86.     handSnapshots[1] = Array.from(document.querySelectorAll("#opponent .uno"));
87.     animations = gameData.animations;
88.     currentGameData = gameData;
89.     animateContinue();
90.   }
91.
92.   /* Runs single animation from global animations array. */
93.   function animateContinue() {
94.     if (animations.length > 0) {
95.       let animation = animations.shift();
96.       if (animation.moveType === "play") {
97.         let deck = animation.isPlayer ? handSnapshots[0] : handSnapshots[1];
98.         playToDiscard(deck[animation.cardIndex]);
99.       } else if (animation.moveType === "draw") {
100.        let deck = animation.isPlayer ? $("player") : $("opponent");
101.        drawCard(deck);
102.      } else if (animation.moveType === "won" || animation.moveType === "lost") {
103.        canPlay = false;
104.        $("start-new").classList.remove("hidden");
105.        displayTable(currentGameData);
106.      } else {
107.        animateContinue();
108.      }
109.    } else {
110.      displayTable(currentGameData);
111.      canPlay = true;
112.    }
113.  }
114.
115.  /**
116.   * Animates card moving to discard pile.
117.   * @param {object} card - Card DOM object to animate.
118.   */
119.  function playToDiscard(card) {
120.    card.addEventListener("transitionend", function(event) {
121.      if (event.propertyName === "top") {
122.        card.parentNode.removeChild(card);
123.        $("discard").className = card.className;
124.        animateContinue();
125.      }
```

```
126.      });
127.      flyTo(card, $("discard"));
128.    }
129.
130.    /**
131.     * Animates drawing card from deck to players hand.
132.     * @param {object} deck - Deck DOM object.
133.     */
134.    function drawCard(deck) {
135.      let newCard = document.createElement("div");
136.      newCard.id = "new-card";
137.      newCard.classList.add("blank");
138.      newCard.classList.add("uno");
139.      newCard.classList.add("back");
140.      deck.appendChild(newCard);
141.      if (deck.id === "player") {
142.        handSnapshots[0].push(newCard);
143.      } else {
144.        handSnapshots[1].push(newCard);
145.      }
146.
147.      let targetOffset = getOffset($("pile"));
148.      $("topPile").style.right = targetOffset.right + "px";
149.      $("topPile").classList.remove("hidden");
150.      flyTo($("topPile"), newCard);
151.    }
152.
153.    /**
154.     * Animates a card object flying to another card.
155.     * @param {object} card - DOM object to animate.
156.     * @param {object} target - DOM object to fly to.
157.     */
158.    function flyTo(card, target) {
159.      card.style.zIndex = 1;
160.      let cardOffset = getOffset(card);
161.      card.style.position = "absolute";
162.      card.style.top = cardOffset.top + "px";
163.      card.style.left = cardOffset.left + "px";
164.
165.      let targetOffset = getOffset(target);
166.      card.style.position = "absolute";
167.      card.style.top = targetOffset.top + "px";
168.      card.style.left = targetOffset.left + "px";
169.    }
170.
171.    /**
172.     * Returns the offset of a card from its parent container.
173.     * @param {object} card - DOM object to get offset.
174.     * @return {object} JSON object containing offsets.
175.     */
176.    function getOffset(card) {
177.      let childPos = card.getBoundingClientRect();
178.      let parentPos = $("table").getBoundingClientRect();
179.      return {
180.          top: childPos.top - parentPos.top,
181.          left: childPos.left - parentPos.left - 5,
182.          right: parentPos.right - childPos.right + 5
183.      };
184.    }
185.
186.    /**
187.     * Displays all card hands and piles on table based on given game data.
188.     * @param {object} gameData - JSON object containing game data.
189.     */
190.    function displayTable(gameData) {
191.      if (!guid) {
192.        guid = gameData.guid;
193.      }
194.      $("results").innerHTML = gameData.results;
195.      displayCards($("player"), gameData.playerHand);
196.      displayCards($("opponent"), gameData.opponentHand);
197.      displayDiscard(gameData.discard);
198.    }
199.
200.    /**
201.     * Displays the discard based on given discard JSON object.
202.     * @param {object} discard - JSON object of discard to display.
203.     */
204.    function displayDiscard(discard) {
205.      if (discard) {
206.        $("discard").className = "";
```

```
207.       $("discard").classList.add("uno");
208.       $("discard").classList.add(discard.type);
209.       if (discard.color) {
210.         $("discard").classList.add(discard.color);
211.       }
212.     } else {
213.       $("discard").classList.add("blank");
214.     }
215.   }
216.
217.   /**
218.    * Displays cards in given hand.
219.    * @param {object} hand - hand DOM object.
220.    * @param {array} cards - Array of card objects.
221.    */
222.   function displayCards(hand, cards) {
223.     while (hand.lastChild) {
224.       hand.removeChild(hand.lastChild);
225.     }
226.     for (let i = 0; i < cards.length; i++) {
227.       createCard(hand, cards[i]);
228.     }
229.   }
230.
231.   /**
232.    * Creates a given card in hand.
233.    * @param {object} hand - hand DOM object.
234.    * @param {object} cardData - JSON data of card.
235.    */
236.   function createCard(hand, cardData) {
237.       let card = document.createElement("div");
238.       card.classList.add("uno");
239.       if (cardData.color) {
240.         card.classList.add(cardData.color);
241.       }
242.       card.classList.add(cardData.type);
243.       if (hand.id === "player") {
244.         if (cardData.type === "wild" || cardData.type === "wildDraw4") {
245.           card.onclick = wildCheckPlayable;
246.         } else {
247.           card.onclick = function() {
248.             playMove(getIndex(this));
249.           };
250.         }
251.       }
252.       hand.appendChild(card);
253.   }
254.
255.   /* Checks if a wild card is playable */
256.   function wildCheckPlayable() {
257.     if (canPlay) {
258.       currentWildMove = getIndex(this);
259.       let data = new FormData();
260.       data.append("guid", guid);
261.       data.append("move", currentWildMove);
262.       data.append("checkplayable", true);
263.
264.       fetch(GAME_URL, {credentials: 'include', method: "POST", body: data})
265.         .then(checkStatus)
266.         .then(tryJSONParse)
267.         .then(wildChooseColor)
268.         .catch(console.log);
269.     } else {
270.       animateContinue();
271.     }
272.   }
273.
274.   /**
275.    * Shows wild card color chooser
276.    * @param {object} gameData - JSON object containing game data.
277.    */
278.   function wildChooseColor(gameData) {
279.     if (gameData.playable) {
280.       $("color-picker").classList.remove("hidden");
281.     }
282.     $("results").innerHTML = gameData.results;
283.   }
284.
285.   /* Plays wild card. */
286.   function wildMove() {
287.     playMove(currentWildMove, this.id);
```

```
288.     $("color-picker").classList.add("hidden");
289.   }
290.
291.   /**
292.    * Fetches given card move with given move index and color.
293.    * @param {integer} move - card index.
294.    * @param {string} color - card color.
295.    */
296.   function playMove(move, color) {
297.     if (canPlay) {
298.       let data = new FormData();
299.       data.append("guid", guid);
300.       data.append("move", move);
301.       if (color) {
302.         data.append("color", color);
303.       }
304.       fetch(GAME_URL, {credentials: 'include', method: "POST", body: data})
305.         .then(checkStatus)
306.         .then(tryJSONParse)
307.         .then(runAnimations)
308.         .catch(console.log);
309.     } else {
310.       animateContinue();
311.     }
312.   }
313.
314.   /* Returns index of element in parent */
315.   function getIndex(element) {
316.     return Array.from(element.parentNode.children).indexOf(element);
317.   }
318.
319.   /**
320.   * Checks the status response code.
321.   * @param {object} response - fetch response object.
322.   * @return {object} Returns a succesful promise if status is successful, and
323.   * returns a rejected promise otherwise.
324.   */
325.   function checkStatus(response) {
326.     if (response.status >= 200 && response.status < 300) {
327.       return response.text();
328.     } else {
329.       return Promise.reject(new Error(response.status +
330.                                       ": " + response.statusText));
331.     }
332.   }
333. })();
```

uno.php (14728 bytes)

```
1.  <?php
2.    /*
3.    Name: Jack Venberg
4.    Date: 05.14.18
5.    Section: CSE 154 AH
6.
7.    This is the uno.php script for my Creative Project in which I have a Uno game
8.    that connects to a custom-built uno API. This script runs the Uno API on
9.    cloud9 servers and handles all Uno game logic through POST calls.
10.   */
11.
12.   error_reporting(E_ALL);
13.   /* All card types, colors, and names */
14.   $CARD_TYPE = ["num0", "num1", "num2", "num3", "num4", "num5", "num6", "num7", "num8", "num9",
15.                 "skip", "reverse", "draw2", "wild", "wildDraw4"];
16.   $CARD_NAMES = ["Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine",
17.                  "Skip", "Reverse", "Draw Two", "Wild Card", "Wild Draw Four"];
18.   $CARD_COLORS = ["red", "yellow", "green", "blue"];
19.   $skip = false; /* Whether or not currently skipping opponent */
20.
21.   /* Database connection variables */
22.   $servername = "localhost";
23.   $username = "root";
24.   $password = "";
25.   $database = "c9";
26.
27.   /* Make a data source string that will be used in creating the PDO object */
28.   $ds = "mysql:host={$servername};dbname={$database};charset=utf8";
29.
```

```
30.  try {
31.    $db = new PDO($ds, $username, $password);
32.    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
33.  } catch (PDOException $ex) {
34.    handle_error("Can not connect to the database. Please try again later.", $ex);
35.  }
36.
37.  header("Content-Type: application/json");
38.  $animations = array(); /* Array for storing animation objects to be sent back */
39.  if (isset($_POST["startgame"]) and $_POST["startgame"]) {
40.    start_game();
41.  } else if (isset($_POST["guid"]) and isset($_POST["move"])) {
42.    if (isset($_POST["checkplayable"])) {
43.      check_if_playable();
44.    } else {
45.      play_move();
46.    }
47.  }
48.
49.  $db = null;
50.
51.  /* Starts game by setting up database and displaying initial values. */
52.  function start_game() {
53.    $GLOBALS["guid"] = uniqid();
54.    $GLOBALS["deck"] = create_deck();
55.
56.    shuffle($GLOBALS["deck"]);
57.    list($GLOBALS["players_hand"], $GLOBALS["opponents_hand"]) = deal_hands($GLOBALS["deck"]);
58.
59.    $game_data = array();
60.    $game_data["guid"] = $GLOBALS["guid"];
61.    $game_data["playerHand"] = $GLOBALS["players_hand"];
62.    $game_data["opponentHand"] = hide_cards($GLOBALS["opponents_hand"]);
63.    $game_data["results"] = "<p>Created new deck of cards.</p>";
64.    $game_data["animations"] = array();
65.    array_push($game_data["animations"], new Animation("new_deck", null, null));
66.    print(json_encode($game_data));
67.
68.    $stmt = $GLOBALS["db"]->prepare("INSERT INTO games (guid, playerHand, opponentHand, deck)
69.                                     VALUES (:guid, :player_hand, :opponent_hand, :deck)");
70.    $params = array("guid" => $GLOBALS["guid"],
71.                    "player_hand" => serialize($GLOBALS["players_hand"]),
72.                    "opponent_hand" => serialize($GLOBALS["opponents_hand"]),
73.                    "deck" => serialize($GLOBALS["deck"]));
74.    $stmt->execute($params);
75.  }
76.
77.  /**
78.   * Plays a move of a card depending on what "move" is set to in POST
79.   * request.
80.   */
81.  function play_move() {
82.    read_db();
83.
84.    $game_data = array();
85.    $move = $_POST["move"];
86.    if (is_numeric($move) and $move < count($GLOBALS["players_hand"]) and $move >= 0) {
87.      if (is_playable($GLOBALS["players_hand"][$move], $GLOBALS["players_hand"])) {
88.        $game_data["results"] = player_play($move);
89.
90.        if (count($GLOBALS["players_hand"]) == 0) {
91.          $game_data["results"] .= "<p>YOU WON!</p>";
92.          array_push($GLOBALS["animations"], new Animation("won", null, null));
93.        } else {
94.          $game_data["results"] .= opponent_play();
95.          if (count($GLOBALS["opponents_hand"]) == 0) {
96.            $game_data["results"] .= "<p>YOU LOST!</p>";
97.            array_push($GLOBALS["animations"], new Animation("lost", null, null));
98.          }
99.        }
100.     } else {
101.       $game_data["results"] = "<p>Card is unplayable.</p>";
102.     }
103.   } else if ($move == "new") {
104.     check_deck();
105.     array_push($GLOBALS["players_hand"], array_pop($GLOBALS["deck"]));
106.     array_push($GLOBALS["animations"], new Animation("draw", null, true));
107.     $game_data["results"] = "<p>You drew a card.</p>";
108.   }
109.
110.   $game_data["guid"] = $GLOBALS["guid"];
```

```
111.      $game_data["playerHand"] = $GLOBALS["players_hand"];
112.      $game_data["opponentHand"] = hide_cards($GLOBALS["opponents_hand"]);
113.      $game_data["discard"] = end($GLOBALS["discard"]);
114.      $game_data["animations"] = $GLOBALS["animations"];
115.      print(json_encode($game_data));
116.
117.      write_db();
118.    }
119.
120.    /**
121.     * Checks if a card specified by what "move" is set to in POST request.
122.     * Returns true if playable and false otherwise.
123.     * @return {boolean} Whether or not the move is playable.
124.     */
125.    function check_if_playable() {
126.      read_db();
127.
128.      $game_data = array();
129.      $game_data["guid"] = $GLOBALS["guid"];
130.      if (is_playable($GLOBALS["players_hand"][$_POST["move"]], $GLOBALS["players_hand"])) {
131.        $game_data["playable"] = true;
132.        $game_data["results"] = "<p>Please choose a color.</p>";
133.      } else {
134.        $game_data["playable"] = false;
135.        $game_data["results"] = "<p>Card is unplayable.</p>";
136.      }
137.
138.      print(json_encode($game_data));
139.
140.      write_db();
141.    }
142.
143.    /**
144.     * Plays a move by the player based on the given $move index.
145.     * @param {integer} $move - Index of card to play.
146.     */
147.    function player_play($move) {
148.      $played_card = play_card($move, $GLOBALS["players_hand"], $GLOBALS["opponents_hand"], true);
149.      return "<p>You played a " . $played_card->name . ".</p>";
150.    }
151.
152.    /* Plays a move for the player randomly choosing a playable card. */
153.    function opponent_play() {
154.      if (!$GLOBALS["skip"]) {
155.        $drew_count = 0;
156.        while(true) {
157.          $playable_moves = array();
158.          foreach ($GLOBALS["opponents_hand"] as $move=>$card) {
159.            if (is_playable($card, $GLOBALS["opponents_hand"])) {
160.              array_push($playable_moves, $move);
161.            }
162.          }
163.          if (count($playable_moves) == 0) {
164.            check_deck();
165.            array_push($GLOBALS["opponents_hand"], array_pop($GLOBALS["deck"]));
166.            array_push($GLOBALS["animations"], new Animation("draw", null, false));
167.            $drew_count++;
168.          } else {
169.            $move = $playable_moves[array_rand($playable_moves)];
170.            $played_card = play_card($move, $GLOBALS["opponents_hand"], $GLOBALS["players_hand"], false)
171.            $result = "<p>Opponent ";
172.            if ($drew_count > 0) {
173.              $result .= "drew " . $drew_count . " times and " ;
174.            }
175.            $result .= "played a " . $played_card->name . ".</p>";
176.            if ($played_card->type == "skip" || $played_card->type == "reverse") {
177.              $result .= "<p>Player was skipped</p>";
178.              $GLOBALS["skip"] = false;
179.              $result .= opponent_play();
180.            }
181.            return $result;
182.          }
183.        }
184.      } else {
185.        $GLOBALS["skip"] = false;
186.        return "<p>Opponent was skipped</p>";
187.      }
188.    }
189.
190.    /**
191.     * Plays a specific card speficied by the "$move" index. Takes card from given
```

```
192.     * "$hand" and will draw cards to "$otherHand".
193.     * @param {integer} $move - Index of card to play.
194.     * @param {array} $hand - Array of Card objects of hand to play from.
195.     * @param {array} $other_hand - Array of Card objects to draw to.
196.     * @param {boolean} $is_player - Whether of not move is for player.
197.     */
198.    function play_card($move, &$hand, &$other_hand, $is_player) {
199.      $played_card = $hand[$move];
200.      $played_card->index = $move;
201.      array_push($GLOBALS["animations"], new Animation("play", $move, $is_player));
202.      if ($played_card->type == "wild" or $played_card->type == "wildDraw4") {
203.        if ($is_player) {
204.          $played_card->color = $_POST["color"];
205.        } else {
206.          $played_card->color = $GLOBALS["CARD_COLORS"][array_rand($GLOBALS["CARD_COLORS"])];
207.        }
208.      }
209.      if ($played_card->type == "wildDraw4") {
210.        for ($i = 0; $i < 4; $i++) {
211.          check_deck();
212.          array_push($other_hand, array_pop($GLOBALS["deck"]));
213.          array_push($GLOBALS["animations"], new Animation("draw", null, !$is_player));
214.        }
215.      } else if ($played_card->type == "draw2") {
216.        for ($i = 0; $i < 2; $i++) {
217.          check_deck();
218.          array_push($other_hand, array_pop($GLOBALS["deck"]));
219.          array_push($GLOBALS["animations"], new Animation("draw", null, !$is_player));
220.        }
221.      } else if ($played_card->type == "skip" or $played_card->type == "reverse") {
222.        $GLOBALS["skip"] = true;
223.      }
224.      array_push($GLOBALS["discard"], $played_card);
225.      unset($hand[$move]);
226.      $hand = array_values($hand);
227.      return $played_card;
228.    }
229.
230.    /**
231.     * Reads from mySQL database based on what "guid" is set to in POST request
232.     * and stores data into global variables.
233.     */
234.    function read_db() {
235.      $stmt = $GLOBALS["db"]->prepare("SELECT * FROM games WHERE guid=:guid");
236.      $params = array("guid" => $_POST["guid"]);
237.      $stmt->execute($params);
238.      $row = $stmt->fetch();
239.
240.      $discard = isset($row["discard"]) ? unserialize($row["discard"]) : array();
241.      $GLOBALS["guid"] = $row["guid"];
242.      $GLOBALS["deck"] = unserialize($row["deck"]);
243.      $GLOBALS["players_hand"] = unserialize($row["playerHand"]);
244.      $GLOBALS["opponents_hand"] = unserialize($row["opponentHand"]);
245.      $GLOBALS["discard"] = $discard;
246.
247.    }
248.
249.    /* Writes to mySQL database based global table variables. */
250.    function write_db() {
251.      $stmt = $GLOBALS["db"]->prepare("UPDATE games SET playerHand=:playerHand,
252.                                       opponentHand=:opponentHand, deck=:deck, discard=:discard
253.                                       WHERE guid=:guid");
254.      $params = array("guid" => $GLOBALS["guid"],
255.                      "playerHand" => serialize($GLOBALS["players_hand"]),
256.                      "opponentHand" => serialize($GLOBALS["opponents_hand"]),
257.                      "deck" => serialize($GLOBALS["deck"]),
258.                      "discard" => serialize($GLOBALS["discard"]));
259.      $stmt->execute($params);
260.    }
261.
262.    /**
263.     * Checks to make sure deck has hards. If not, it takes discard pile and puts
264.     * discarded cards into deck.
265.     */
266.    function check_deck() {
267.      if (count($GLOBALS["deck"]) == 0) {
268.        $GLOBALS["deck"] = $GLOBALS["discard"];
269.        $GLOBALS["discard"] = array();
270.      }
271.    }
272.
```

```
273.   /**
274.    * Checks if a given card is playable from a given hand. Wild draw 4 is
275.    * playable only if no other cards are playable.
276.    * @param {object} $card - Card object that's checked for playablility.
277.    * @param {array} $hand - Array of Card objects to check from.
278.    * @return {boolean} Whether or not card is playable. True if yes, no otherwise.
279.    */
280.   function is_playable($card, $hand) {
281.     if ($card->type == "wildDraw4") {
282.       foreach($hand as $card_in_hand) {
283.         if ($card_in_hand->type != "wildDraw4" and is_playable_helper($card_in_hand)) {
284.           return false;
285.         }
286.       }
287.       return true;
288.     }
289.     return is_playable_helper($card);
290.   }
291.
292.   /**
293.    * Checks if a given card is playable.
294.    * @param {object} $card - Card object that's checked for playablility.
295.    * @return {boolean} Whether or not card is playable. True if yes, no otherwise.
296.    */
297.   function is_playable_helper($card) {
298.     if (!$GLOBALS["discard"]) {
299.       return true;
300.     }
301.     if ($card->type == "wild") {
302.       return true;
303.     }
304.     $top_discard = end($GLOBALS["discard"]);
305.     if ($card->type == $top_discard->type or
306.         ($card->color == $top_discard->color and $card->color != null)) {
307.       return true;
308.     }
309.     return false;
310.   }
311.
312.   /**
313.    * Prints the given error message with details about the error.
314.    * @param {string} $msg - Error message.
315.    * @param {object} $ex - Error object.
316.    */
317.   function handle_error($msg, $ex) {
318.     header("Content-Type: text/plain");
319.     print ("{$msg}\n");
320.     print ("Error details: $ex \n");
321.     die();
322.   }
323.
324.   /**
325.    * Obscures given hand by returning a hand with an equal number of back-facing
326.    * cards.
327.    * @param {array} $hand - Array of Card objects for hand.
328.    * @return {array} Array of obscured Card objects.
329.    */
330.   function hide_cards($hand) {
331.     $hidden_hand = array();
332.     for ($i = 0; $i < count($hand); $i++) {
333.       array_push($hidden_hand, new Card("back", null, "Back"));
334.     }
335.     return $hidden_hand;
336.   }
337.
338.   /**
339.    * Deals equally to each hand from given deck.
340.    * @param {array} $deck - Array of Card objects for deck.
341.    * @return {array} Array of two arrays of Card objects for each hand.
342.    */
343.   function deal_hands(&$deck) {
344.     $deal_player = true;
345.     $players_hand = array();
346.     $opponents_hand = array();
347.     for ($i = 0; $i < 14; $i++) {
348.       if ($deal_player) {
349.         array_push($players_hand, array_pop($deck));
350.       } else {
351.         array_push($opponents_hand, array_pop($deck));
352.       }
353.       $deal_player = !$deal_player;
```

```
354.     }
355.     return [$players_hand, $opponents_hand];
356.   }
357.
358.   /**
359.    * Creates a deck of uno cards.
360.    * return {array} Array of Card objects representing the deck.
361.    */
362.   function create_deck() {
363.     $deck = array();
364.     for ($i = 0; $i < 15; $i++) {
365.       for ($j = 0; $j < 4; $j++) {
366.         if ($i < 13) {
367.           $name = ucfirst($GLOBALS["CARD_COLORS"][$j]) . " " . $GLOBALS["CARD_NAMES"][$i];
368.           if ($i > 0) {
369.             array_push($deck, new Card($GLOBALS["CARD_TYPE"][$i], $GLOBALS["CARD_COLORS"][$j], $name))
370.           }
371.           array_push($deck, new Card($GLOBALS["CARD_TYPE"][$i], $GLOBALS["CARD_COLORS"][$j], $name));
372.         } else {
373.           array_push($deck, new Card($GLOBALS["CARD_TYPE"][$i], null, $GLOBALS["CARD_NAMES"][$i]));
374.         }
375.       }
376.     }
377.     return $deck;
378.   }
379.
380.   /* Card object for storing card information. */
381.   class Card {
382.     public $type;
383.     public $color;
384.     public $name;
385.     public $index;
386.
387.     /**
388.      * Constructs Card object.
389.      * @param {string} $type - Card type.
390.      * @param {string} $color - Card color.
391.      * @param {string} $name - Card name.
392.      */
393.     public function __construct($type, $color, $name) {
394.       $this->type = $type;
395.       $this->color = $color;
396.       $this->name = $name;
397.     }
398.   }
399.
400.   /* Card object for storing animation information. */
401.   class Animation {
402.     public $moveType;
403.     public $cardIndex;
404.     public $isPlayer;
405.
406.     /**
407.      * Constructs Card object.
408.      * @param {string} $move_type - Animation type.
409.      * @param {integer} $card_index - Index of card in hand to move.
410.      * @param {boolean} $is_player - Whether or not animation is for the player.
411.      */
412.     public function __construct($move_type, $card_index, $is_player) {
413.       $this->moveType = $move_type;
414.       $this->cardIndex = (int)$card_index;
415.       $this->isPlayer = (bool)$is_player;
416.     }
417.   }
418. ?>
```