The Nocturnal Gamblers

•••

*Anthony Asilo, Hiren Patel, Wen Hao Zhu, Akiva Ochoa

*Leader

https://codd.cs.gsu.edu/~aasilo1/WP/PW/01/index.html

User - Problem Statement / Solution + Gen Reqs

- Problem statement from user918374: “Let’s be honest, this pandemic sucks for everyone and travelling isn’t even the same anymore. I just wish I could go back to Las Vegas and go to a Casino. Oh how I’d love to play a Slot Machine!”

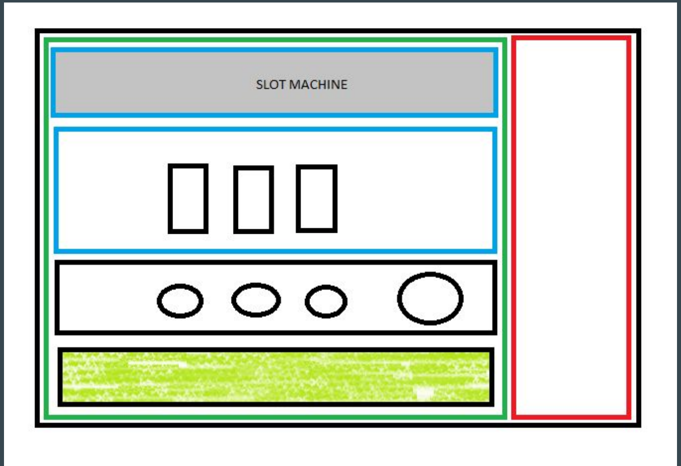
- Solution and General requirements: There are not inputs/outputs for our webpage due to the fact that we strictly used CSS and HTML only, unless you consider a link reference to the summary page an input and the page itself as an output.

Inspiration

- Caesars Palace is a famous casino in Las Vegas, Nevada.
- The <title> of index is Caesars Palace and we used a background image of the famous casino room.

Overview

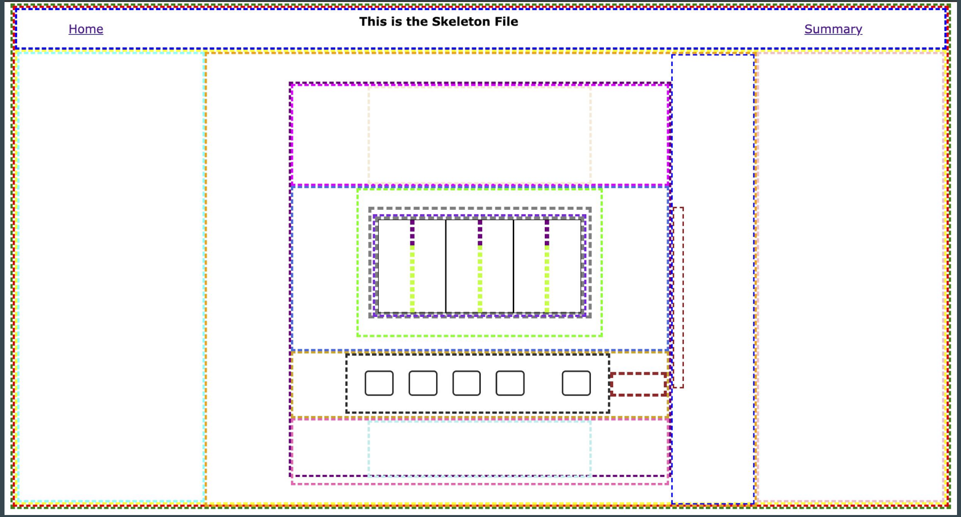
- Discord for Communication
- Decided on a Slot Machine
- Utilized Google Images for an idea
- Mockup by Akiva



User Interface (Skel)

- Created a skeleton of the Index to hold all div containers to match the image template
- Created side bars and header for easy to access information that is not overbearing.

JACKPOT
SLOT MACHINE
7
7
7
RESET
PAY TABLE
BET ONE
BET MAX
SPIN


Home

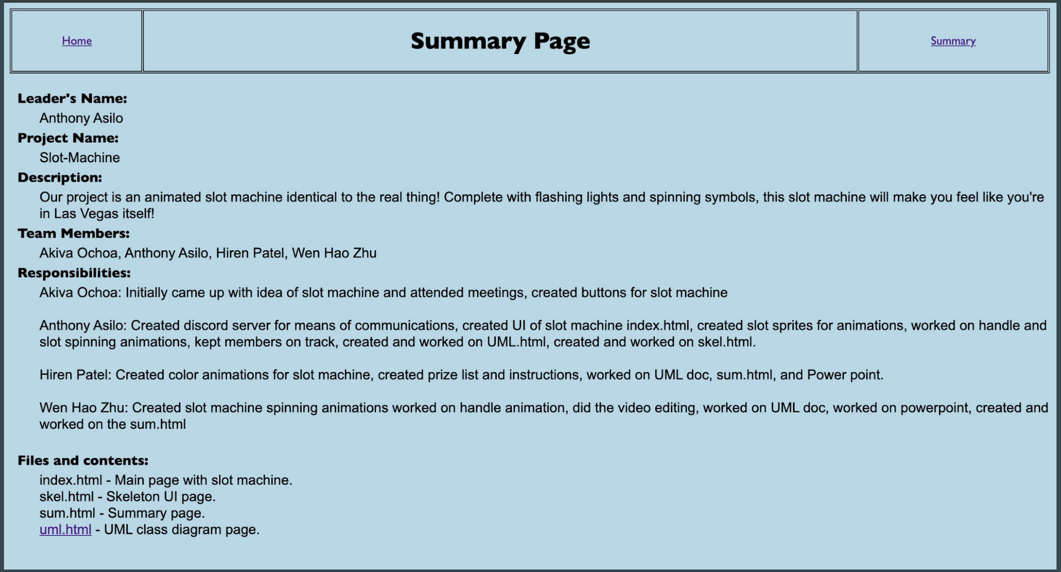
This is the Skeleton File

Summary

https://codd.cs.gsu.edu/~aasilo1/WP/PW/01/skel.html

User Interface (Summary)

- We wanted the summary page to be simple, clear and concise.
- Header consistent to all pages
- Link to UML



Home	**Summary Page**	Summary

Leader's Name:

Anthony Asilo

Project Name:

Slot-Machine

Description:

Our project is an animated slot machine identical to the real thing! Complete with flashing lights and spinning symbols, this slot machine will make you feel like you're in Las Vegas itself!

Team Members:

Akiva Ochoa, Anthony Asilo, Hiren Patel, Wen Hao Zhu

Responsibilities:

Akiva Ochoa: Initially came up with idea of slot machine and attended meetings, created buttons for slot machine

Anthony Asilo: Created discord server for means of communications, created UI of slot machine index.html, created slot sprites for animations, worked on handle and slot spinning animations, kept members on track, created and worked on UML.html, created and worked on skel.html.

Hiren Patel: Created color animations for slot machine, created prize list and instructions, worked on UML doc, sum.html, and Power point.

Wen Hao Zhu: Created slot machine spinning animations worked on handle animation, did the video editing, worked on UML doc, worked on powerpoint, created and worked on the sum.html

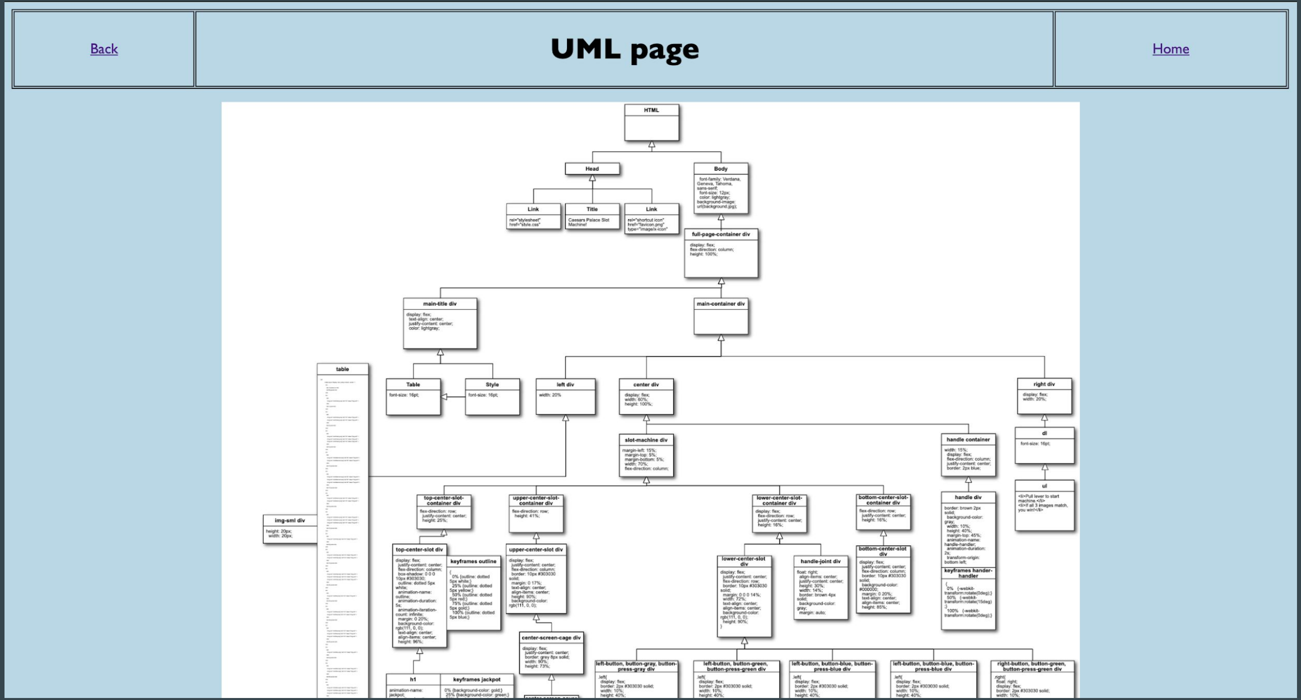
Files and contents:

index.html - Main page with slot machine.
skel.html - Skeleton UI page.
sum.html - Summary page.
uml.html - UML class diagram page.

https://codd.cs.gsu.edu/~aasilo1/WP/PW/01/sum.html

User Interface (UML)

- Design consistent with Summary Page



Back

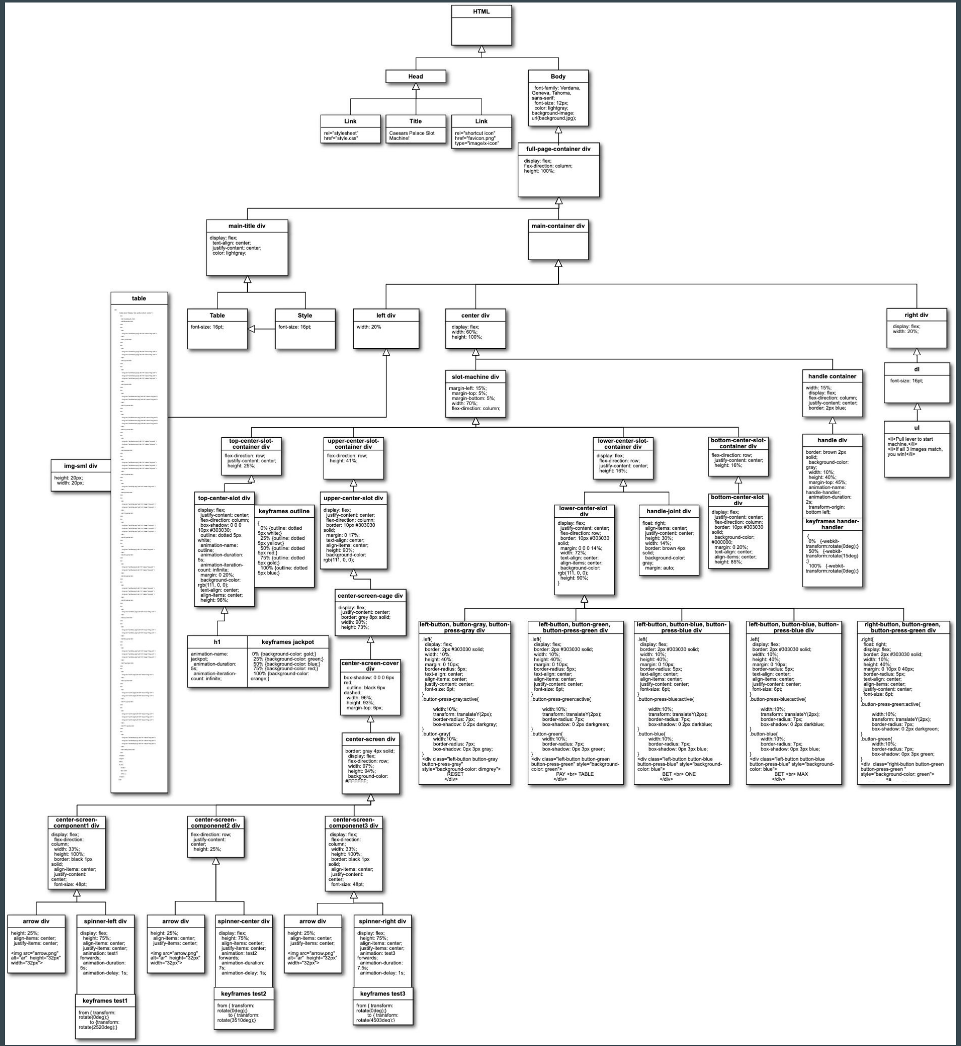
UML page

Home

https://codd.cs.gsu.edu/~aasilo1/WP/PW/01/uml.html

UML Class Design

- A CSS / HTML integrated diagram showing class hierarchy and inheritance of style and animation


HTML
Head
Link
rel="stylesheet" href="style.css"
Title
Caesars Palace Slot Machine!
Link
rel="shortcut icon" href="favicon.png" type="image/x-icon"
Body
font-family: Verdana, Geneva, Tahoma, sans-serif; font-size: 12px; color: lightgray; background-image: url(background.jpg);
full-page-container div
display: flex; flex-direction: column; height: 100%;
main-title div
display: flex; text-align: center; justify-content: center; color: lightgray;
main-container div
table
img-sml div
height: 20px; width: 20px;
Table
font-size: 16pt;
Style
font-size: 16pt;
left div
width: 20%
center div
display: flex; width: 60%; height: 100%;
right div
display: flex; width: 20%;
dl
font-size: 16pt;
ul
<li>Pull lever to start machine.</li> <li>If all 3 images match, you win!</li>
slot-machine div
margin-left: 15%; margin-top: 5%; margin-bottom: 5%; width: 70%; flex-direction: column;
handle container
width: 15%; display: flex; flex-direction: column; justify-content: center; border: 2px blue;
handle div
border: brown 2px solid; background-color: gray; width: 10%; height: 40%; margin-top: 45%; animation-name: handle-handler; animation-duration: 2s; transform-origin: bottom left;
keyframes hander-handler
{ 0% {-webkit-transform:rotate(0deg);} 50% {-webkit-transform:rotate(15deg);} 100% {-webkit-transform:rotate(0deg);}
top-center-slot-container div
flex-direction: row; justify-content: center; height: 25%;
top-center-slot div
display: flex; justify-content: center; flex-direction: column; box-shadow: 0 0 0 10px #303030; outline: dotted 5px white; animation-name: outline; animation-duration: 5s; animation-iteration-count: infinite; margin: 0 20%; background-color: rgb(111, 0, 0); text-align: center; align-items: center; height: 96%;
keyframes outline
{ 0% {outline: dotted 5px white;} 25% {outline: dotted 5px yellow;} 50% {outline: dotted 5px red;} 75% {outline: dotted 5px gold;} 100% {outline: dotted 5px blue;}
h1
animation-name: jackpot; animation-duration: 5s; animation-iteration-count: infinite;
keyframes jackpot
0% {background-color: gold;} 25% {background-color: green;} 50% {background-color: blue;} 75% {background-color: red;} 100% {background-color: orange;}
upper-center-slot-container div
flex-direction: row; height: 41%;
upper-center-slot div
display: flex; justify-content: center; flex-direction: column; border: 10px #303030 solid; margin: 0 17%; text-align: center; align-items: center; height: 90%; background-color: rgb(111, 0, 0);
center-screen-cage div
display: flex; justify-content: center; border: grey 8px solid; width: 90%; height: 73%;
center-screen-cover div
box-shadow: 0 0 0 6px red; outline: black 6px dashed; width: 96%; height: 93%; margin-top: 6px;
center-screen div
border: gray 4px solid; display: flex; flex-direction: row; width: 97%; height: 94%; background-color: #FFFFFF;
center-screen-component1 div
display: flex; flex-direction: column; width: 33%; height: 100%; border: black 1px solid; align-items: center; justify-content: center; font-size: 48pt;
center-screen-componenet2 div
flex-direction: row; justify-content: center; height: 25%;
center-screen-componenet3 div
display: flex; flex-direction: column; width: 33%; height: 100%; border: black 1px solid; align-items: center; justify-content: center; font-size: 48pt;
arrow div
height: 25%; align-items: center; justify-items: center; <img src="arrow.png" alt="ar" height="32px" width="32px">
spinner-left div
display: flex; height: 75%; align-items: center; justify-items: center; animation: test1 forwards; animation-duration: 5s; animation-delay: 1s;
keyframes test1
from { transform: rotate(0deg);} to {transform: rotate(2520deg);}
arrow div
height: 25%; align-items: center; justify-items: center; <img src="arrow.png" alt="ar" height="32px" width="32px">
spinner-center div
display: flex; height: 75%; align-items: center; justify-items: center; animation: test2 forwards; animation-duration: 7s; animation-delay: 1s;
keyframes test2
from { transform: rotate(0deg);} to { transform: rotate(3510deg);}
arrow div
height: 25%; align-items: center; justify-items: center; <img src="arrow.png" alt="ar" height="32px" width="32px">
spinner-right div
display: flex; height: 75%; align-items: center; justify-items: center; animation: test3 forwards; animation-duration: 7.5s; animation-delay: 1s;
keyframes test3
from { transform: rotate(0deg);} to { transform: rotate(4503deg);}
lower-center-slot-container div
display: flex; flex-direction: row; justify-content: center; height: 16%;
lower-center-slot div
display: flex; justify-content: center; flex-direction: row; border: 10px #303030 solid; margin: 0 0 0 14%; width: 72%; text-align: center; align-items: center; background-color: rgb(111, 0, 0); height: 90%; }
handle-joint div
float: right; align-items: center; justify-content: center; height: 30%; width: 14%; border: brown 4px solid; background-color: gray; margin: auto;
bottom-center-slot-container div
flex-direction: row; justify-content: center; height: 16%;
bottom-center-slot div
display: flex; justify-content: center; flex-direction: column; border: 10px #303030 solid; background-color: #000000; margin: 0 20%; text-align: center; align-items: center; height: 85%;
left-button, button-gray, button-press-gray div
.left{ display: flex; border: 2px #303030 solid; width: 10%; height: 40%; margin: 0 10px; border-radius: 5px; text-align: center; align-items: center; justify-content: center; font-size: 6pt; } .button-press-gray:active{ width:10%; transform: translateY(2px); border-radius: 7px; box-shadow: 0 2px darkgray; } .button-gray{ width:10%; border-radius: 7px; box-shadow: 0px 3px gray; } <div class="left-button button-gray button-press-gray" style="background-color: dimgrey"> RESET </div>
left-button, button-green, button-press-green div
.left{ display: flex; border: 2px #303030 solid; width: 10%; height: 40%; margin: 0 10px; border-radius: 5px; text-align: center; align-items: center; justify-content: center; font-size: 6pt; } .button-press-green:active{ width:10%; transform: translateY(2px); border-radius: 7px; box-shadow: 0 2px darkgreen; } .button-green{ width:10%; border-radius: 7px; box-shadow: 0px 3px green; } <div class="left-button button-green button-press-green" style="background-color: green"> PAY
 TABLE </div>
left-button, button-blue, button-press-blue div
.left{ display: flex; border: 2px #303030 solid; width: 10%; height: 40%; margin: 0 10px; border-radius: 5px; text-align: center; align-items: center; justify-content: center; font-size: 6pt; } .button-press-blue:active{ width:10%; transform: translateY(2px); border-radius: 7px; box-shadow: 0 2px darkblue; } .button-blue{ width:10%; border-radius: 7px; box-shadow: 0px 3px blue; } <div class="left-button button-blue button-press-blue" style="background-color: blue"> BET
 ONE </div>
left-button, button-blue, button-press-blue div
.left{ display: flex; border: 2px #303030 solid; width: 10%; height: 40%; margin: 0 10px; border-radius: 5px; text-align: center; align-items: center; justify-content: center; font-size: 6pt; } .button-press-blue:active{ width:10%; transform: translateY(2px); border-radius: 7px; box-shadow: 0 2px darkblue; } .button-blue{ width:10%; border-radius: 7px; box-shadow: 0px 3px blue; } <div class="left-button button-blue button-press-blue" style="background-color: blue"> BET
 MAX </div>
right-button, button-green, button-press-green div
.right{ float: right; display: flex; border: 2px #303030 solid; width: 10%; height: 40%; margin: 0 10px 0 40px; border-radius: 5px; text-align: center; align-items: center; justify-content: center; font-size: 6pt; } .button-press-green:active{ width:10%; transform: translateY(2px); border-radius: 7px; box-shadow: 0 2px darkgreen; } .button-green{ width:10%; border-radius: 7px; box-shadow: 0px 3px green; } <div class="right-button button-green button-press-green " style="background-color: green"> <a

https://codd.cs.gsu.edu/~aasilo1/WP/PW/01/uml.html

Key Design Features

- Slot machine
- Animations for the machine
- SPIN button
- Combos & Rewards
- Instructions
- Other Files (Skel, Sum, UML)

Design (Cont.)

We wanted our slot machine to create the feeling of being in an actual casino, so we felt that it was necessary to add instructions as well as a prize list.

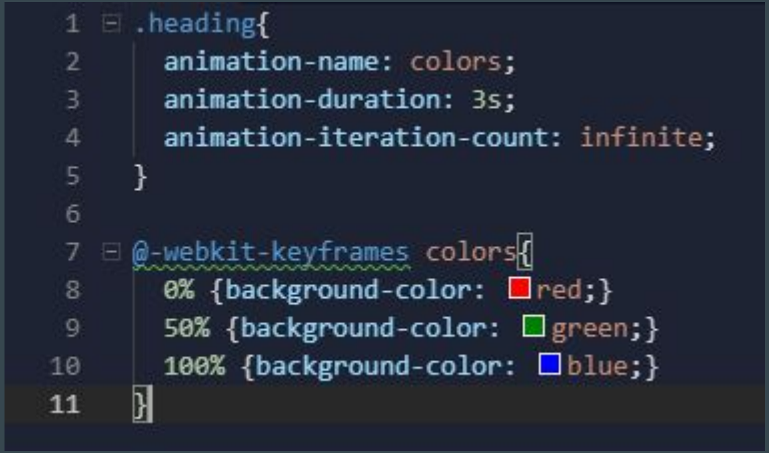
There are various possible combinations that one can obtain depending on the spin of the slot machine.

As a result, there are also many different rewards available due to the great amount of possibilities.

Design (Cont.)

Pseudo code using transformations, transitions, and animations:

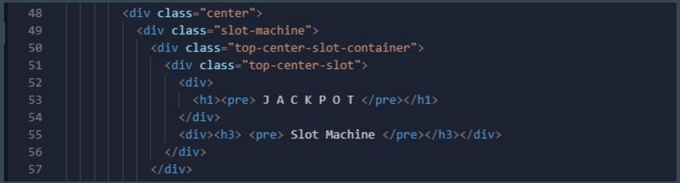
Here is a sample snippet of pseudocode for an animation that continuously changes colors of certain elements, which we implemented into our slot machine.



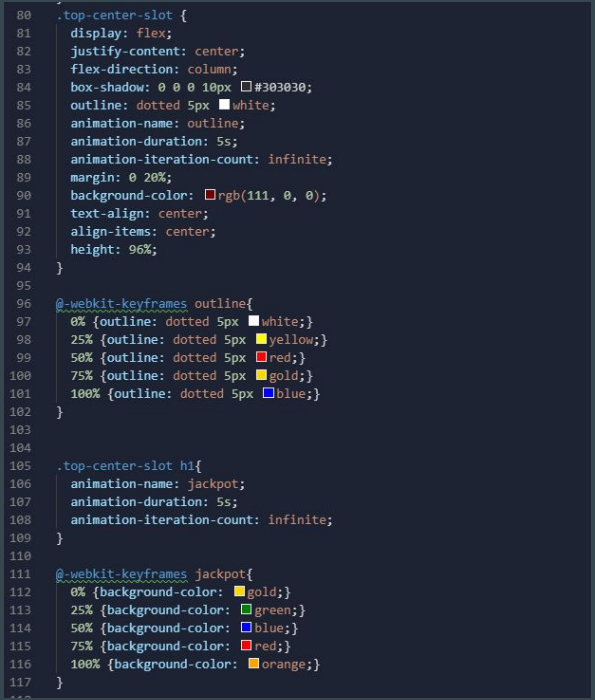
```
.heading{
    animation-name: colors;
    animation-duration: 3s;
    animation-iteration-count: infinite;
}

@-webkit-keyframes colors{
    0% {background-color: 🟥red;}
    50% {background-color: 🟩green;}
    100% {background-color: 🟦blue;}
}
```

Code from our webpage

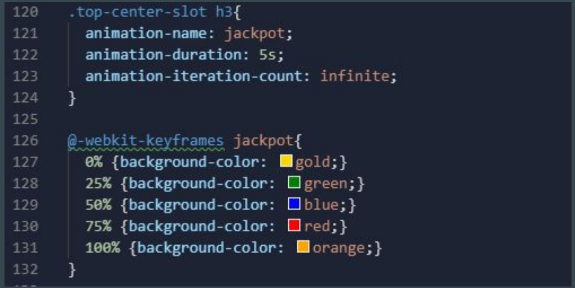
These are code snippets from our webpage, which represent the changing colors on the slot machine.



```html
<div class="center">
  <div class="slot-machine">
    <div class="top-center-slot-container">
      <div class="top-center-slot">
        <div>
          <h1><pre> J A C K P O T </pre></h1>
        </div>
        <div><h3> <pre> Slot Machine </pre></h3></div>
      </div>
    </div>
```



```css
.top-center-slot {
  display: flex;
  justify-content: center;
  flex-direction: column;
  box-shadow: 0 0 0 10px #303030;
  outline: dotted 5px white;
  animation-name: outline;
  animation-duration: 5s;
  animation-iteration-count: infinite;
  margin: 0 20%;
  background-color: rgb(111, 0, 0);
  text-align: center;
  align-items: center;
  height: 96%;
}

@-webkit-keyframes outline{
  0% {outline: dotted 5px white;}
  25% {outline: dotted 5px yellow;}
  50% {outline: dotted 5px red;}
  75% {outline: dotted 5px gold;}
  100% {outline: dotted 5px blue;}
}


.top-center-slot h1{
  animation-name: jackpot;
  animation-duration: 5s;
  animation-iteration-count: infinite;
}

@-webkit-keyframes jackpot{
  0% {background-color: gold;}
  25% {background-color: green;}
  50% {background-color: blue;}
  75% {background-color: red;}
  100% {background-color: orange;}
}
```

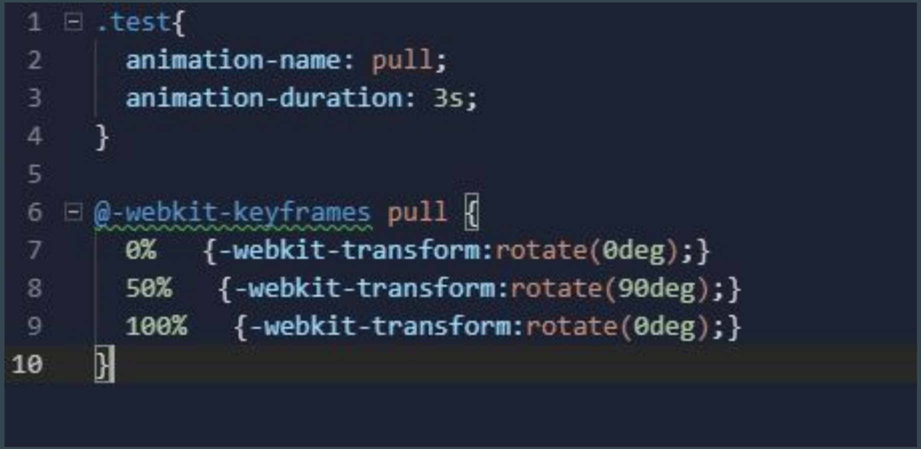


```css
.top-center-slot h3{
  animation-name: jackpot;
  animation-duration: 5s;
  animation-iteration-count: infinite;
}

@-webkit-keyframes jackpot{
  0% {background-color: gold;}
  25% {background-color: green;}
  50% {background-color: blue;}
  75% {background-color: red;}
  100% {background-color: orange;}
}
```

Design (Cont.)

Pseudo code using transformations, transitions, and animations:

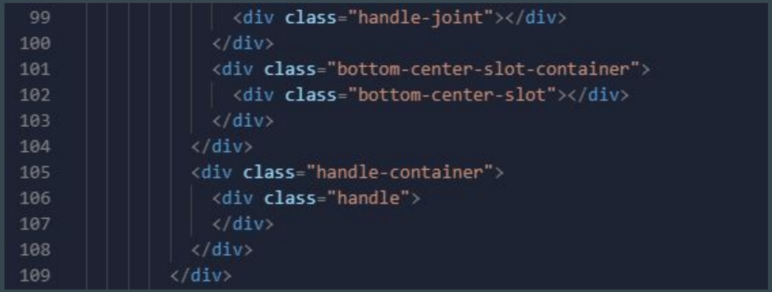
Here is a sample snippet of pseudocode for a pulling animation that we implemented into our slot machine.



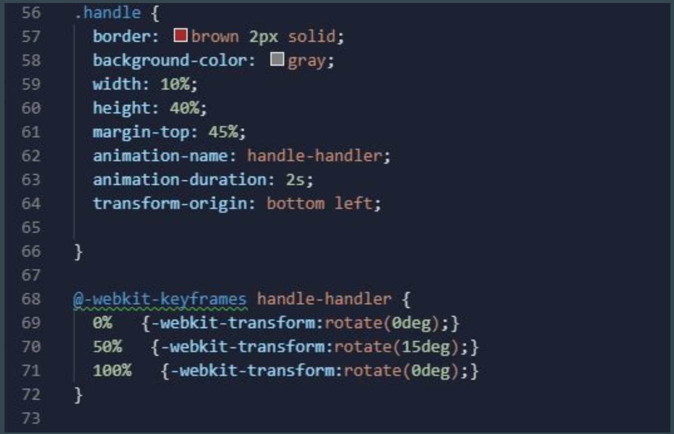
```
.test{
  animation-name: pull;
  animation-duration: 3s;
}

@-webkit-keyframes pull {
  0%   {-webkit-transform:rotate(0deg);}
  50%   {-webkit-transform:rotate(90deg);}
  100%   {-webkit-transform:rotate(0deg);}
}
```

Code from our Webpage

These are code snippets from our webpage, which represent the turning of the handle.



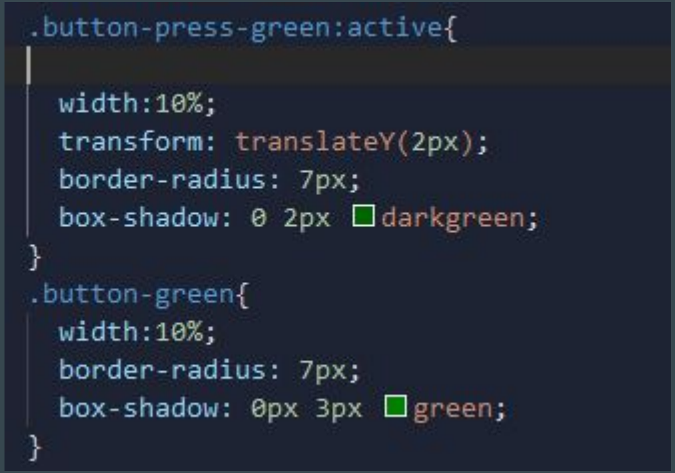
```html
      <div class="handle-joint"></div>
    </div>
    <div class="bottom-center-slot-container">
      <div class="bottom-center-slot"></div>
    </div>
  </div>
  <div class="handle-container">
    <div class="handle">
    </div>
  </div>
</div>
```



```css
.handle {
  border: brown 2px solid;
  background-color: gray;
  width: 10%;
  height: 40%;
  margin-top: 45%;
  animation-name: handle-handler;
  animation-duration: 2s;
  transform-origin: bottom left;

}

@-webkit-keyframes handle-handler {
  0%   {-webkit-transform:rotate(0deg);}
  50%   {-webkit-transform:rotate(15deg);}
  100%   {-webkit-transform:rotate(0deg);}
}

```

Design: Buttons (Static)

- The page buttons imitate a press using CSS’:
- Active selector
- Transform property
- Box Shadows

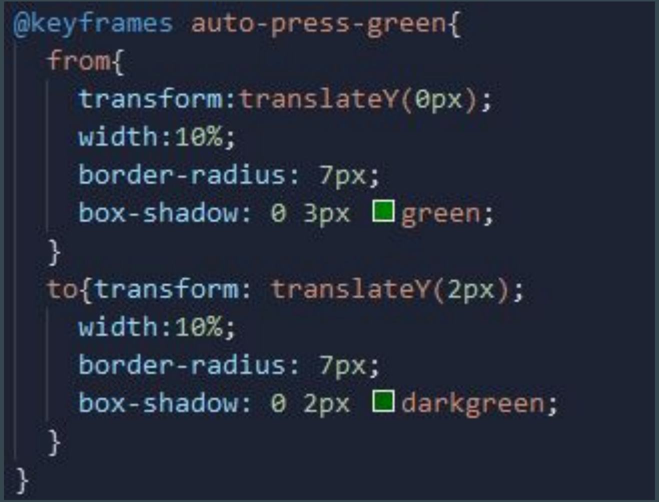


```
.button-press-green:active{

  width:10%;
  transform: translateY(2px);
  border-radius: 7px;
  box-shadow: 0 2px 🟩darkgreen;
}
.button-green{
  width:10%;
  border-radius: 7px;
  box-shadow: 0px 3px 🟩green;
}
```

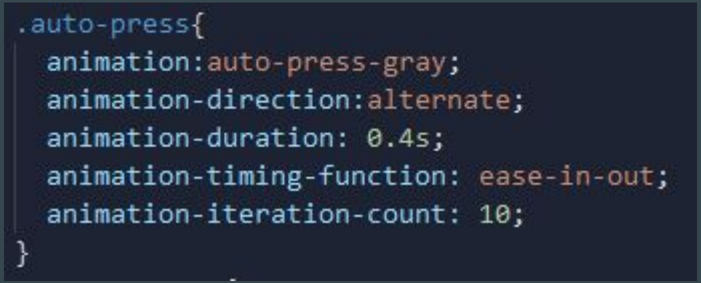
Design: Button (Dynamic)

The startup animation builds on the base button adding:

- Animation
- Transform Properties



```
@keyframes auto-press-green{
  from{
    transform:translateY(0px);
    width:10%;
    border-radius: 7px;
    box-shadow: 0 3px 🟩green;
  }
  to{transform: translateY(2px);
    width:10%;
    border-radius: 7px;
    box-shadow: 0 2px 🟩darkgreen;
  }
}
```

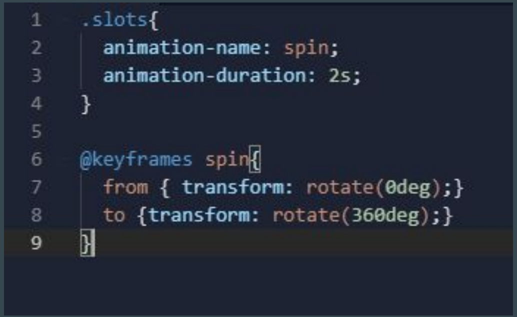


```
.auto-press{
  animation:auto-press-gray;
  animation-direction:alternate;
  animation-duration: 0.4s;
  animation-timing-function: ease-in-out;
  animation-iteration-count: 10;
}
```

Design: The Spinning Animation

Here is a sample snippet of pseudocode for a spinning animation that we implemented into our slot machine

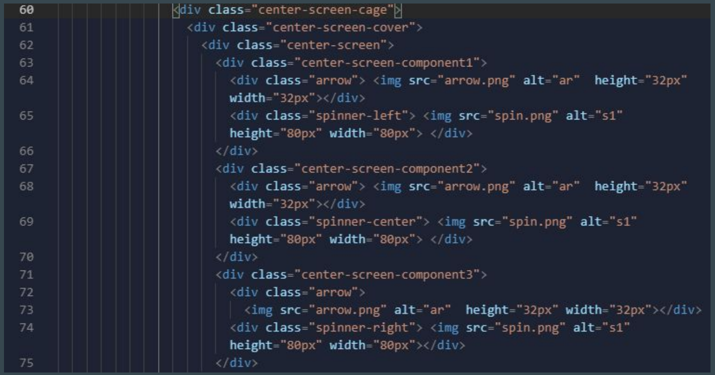
Rotation is done with the use of CSS transformations



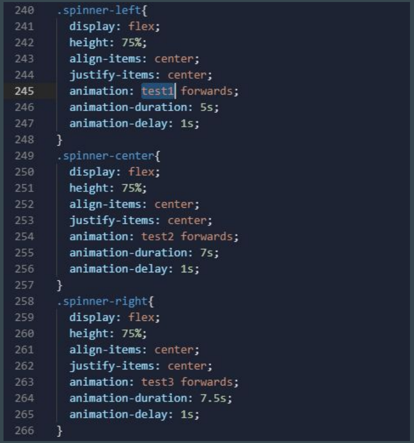
```
.slots{
  animation-name: spin;
  animation-duration: 2s;
}

@keyframes spin{
  from { transform: rotate(0deg);}
  to {transform: rotate(360deg);}
}
```

Code from our Webpage

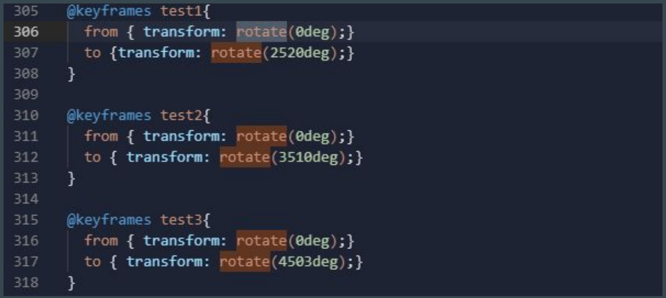
These are code snippets from our webpage, which represent the spinning animation for the slots.



```html
<div class="center-screen-cage">
  <div class="center-screen-cover">
    <div class="center-screen">
      <div class="center-screen-component1">
        <div class="arrow"> <img src="arrow.png" alt="ar"  height="32px"
        width="32px"></div>
        <div class="spinner-left"> <img src="spin.png" alt="s1"
        height="80px" width="80px"> </div>
      </div>
      <div class="center-screen-component2">
        <div class="arrow"> <img src="arrow.png" alt="ar"  height="32px"
        width="32px"></div>
        <div class="spinner-center"> <img src="spin.png" alt="s1"
        height="80px" width="80px"> </div>
      </div>
      <div class="center-screen-component3">
        <div class="arrow">
          <img src="arrow.png" alt="ar"  height="32px" width="32px"></div>
        <div class="spinner-right"> <img src="spin.png" alt="s1"
        height="80px" width="80px"></div>
      </div>
```



```css
.spinner-left{
  display: flex;
  height: 75%;
  align-items: center;
  justify-items: center;
  animation: test1 forwards;
  animation-duration: 5s;
  animation-delay: 1s;
}
.spinner-center{
  display: flex;
  height: 75%;
  align-items: center;
  justify-items: center;
  animation: test2 forwards;
  animation-duration: 7s;
  animation-delay: 1s;
}
.spinner-right{
  display: flex;
  height: 75%;
  align-items: center;
  justify-items: center;
  animation: test3 forwards;
  animation-duration: 7.5s;
  animation-delay: 1s;
}
```



```css
@keyframes test1{
  from { transform: rotate(0deg);}
  to {transform: rotate(2520deg);}
}

@keyframes test2{
  from { transform: rotate(0deg);}
  to { transform: rotate(3510deg);}
}

@keyframes test3{
  from { transform: rotate(0deg);}
  to { transform: rotate(4503deg);}
}
```

Testing

Test plan: To test changes in the website, we implemented version control to roll-back changes at any time if anything did not work. We implemented this with Github and used a site called Repl.it to collaborate together.

Bugs

Tracking and reporting bugs:

- One bug that we encountered when creating div elements was that we used the keyword “style” instead of “class”. As a result, our formatting was thrown off, and our stylesheet was not being implemented properly. However, we were able to quickly figure out what the issue was and fixed it accordingly.

Bugs fixed:

- Div elements (“class” instead of “style”)

Unresolved bugs:

- Some elements are not consistent for different resolutions

Reference

- Full UML Document