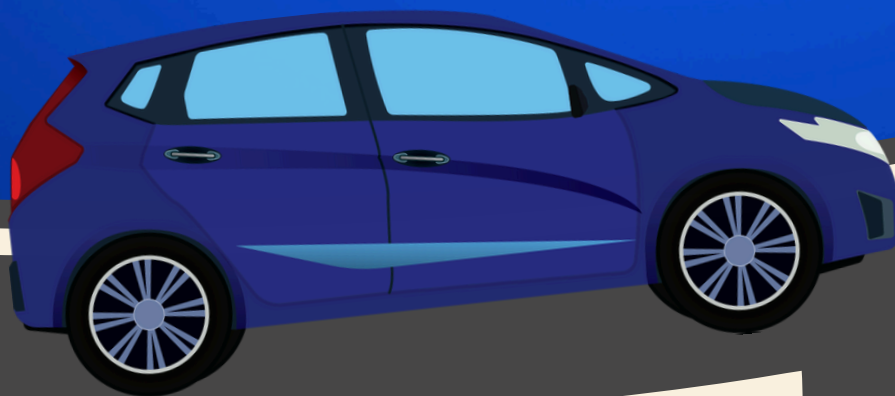


REFLEXES: CARS

Creado por
ROJAS ZAMORA
MAURICIO

DISEÑO Y
MAQUETACION
MAURICIO CASTILLO

TRADUCIDO POR
ALEXANDRA DIAZ



“Reflexes: Cars” Game

Introduction

The CSS code for the "Reflexes: Cars" game defines the entire visual part of the interface: colors, sizes, element distribution, simple animations, and responsive design. It is located within the main file, `index.html`, inside the `<style>` tag. This CSS manages to create the game environment without using external images, only with gradients, shadows, and shapes generated by code.

The style aims for a modern and technological look, featuring dark background colors, soft light tones, and yellow details to highlight the player's actions.

Global Color Variables (:root)

```
:root {  
  --bg: #0f1724;  
  --road: #2a2f36;  
  --lane: #3a3f46;  
  --accent: #ffcc00;  
  --text: #e6eef8;  
}
```

These are custom CSS variables that define the main colors used throughout the game.

Advantage: They allow for maintaining a consistent palette and easily modifying the design without changing every line of code.

- --bg: Main color for the general background (dark blue)
 - --road: Color of the road where the cars appear.
 - --lane: Color of the lanes.
- --accent: Accent color (bright yellow) used for buttons and scores.
 - --text: Light color for text.

General Configuration

```
* { box-sizing: border-box; }
```

It makes the total size of each element include the border and padding. It facilitates the precise alignment of objects.

```
body {  
  margin: 0;  
  font-family: Inter, Segoe UI, Roboto, "Helvetica Neue", Arial;  
  background: linear-gradient(180deg, var(--bg), #071122);  
  color: var(--text);  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  height: 100vh;  
}
```

- It removes the browser's default margins.
- It uses a modern typography (Inter or similar)
- It creates a vertical background gradient, simulating depth.
- It centers all content horizontally and vertically using flexbox.
- The height of the visible area is 100% of the screen height (100vh)

Main Game Container (.wrap)

```
.wrap {  
  width: 960px;  
  max-width: 96vw;  
  background: linear-gradient(180deg, #08132266, #081322aa);  
  padding: 18px;  
  border-radius: 12px;  
  box-shadow: 0 10px 30px rgba(2,6,23,.6);  
  display: grid;  
  grid-template-columns: 1fr 340px;  
  gap: 16px;  
}
```

- It is the main frame that contains the game area and the sidebar.
- It uses CSS Grid to divide the space into two columns:
 - 1.The game zone.
 - 2. The side panel (information, options, buttons).
- box-shadow adds a diffuse shadow to give an elevation effect.
- border-radius softens the edges.
- max-width: 96vw prevents the design from overflowing on small screens.

Game Area (.game-area)

```
.game-area {  
  background: var(--road);  
  border-radius: 10px;  
  position: relative;  
  overflow: hidden;  
  padding: 12px;  
}
```

- It represents the board where the action takes place.
- The color is the same as the road.
- overflow: hidden prevents the cars from leaving the visible area.
- position: relative allows elements to be placed in absolute positions within the area.

Road and Lanes (.road and .lane)

```
.road {  
  background: repeating-linear-gradient(180deg, #1b2229 0 40px, #23292f 40px 80px);  
  height: 640px;  
  max-height: 78vh;  
  border-radius: 8px;  
  position: relative;  
  display: flex;  
  align-items: flex-start;  
  justify-content: center;  
}
```

- It uses a repeating gradient that simulates the pavement with horizontal lines.
- The height is adjusted to 78% of the visible height.
- display: flex organizes the lanes equally within the road.

```
.lane {  
  flex: 1;  
  margin: 0 10px;  
  position: relative;  
}
```

- Each .lane represents a lane where the cars will appear.
- Each .lane represents a lane where the cars will appear.

```
.lane:before {  
  content: '';  
  position: absolute;  
  left: 50%;  
  transform: translateX(-50%);  
  top: 0;  
  bottom: 0;  
  width: 6px;  
  background: linear-gradient(180deg, #dbe7ff66, #dbe7ff22);  
  opacity: .08;  
}
```

- It creates a faint central line inside each lane, simulating road markings.
- It uses a pseudo-element (:before) that does not require additional HTML.

Information Panel and Buttons

```
.hud {
  display: flex;
  gap: 8px;
  align-items: center;
  justify-content: space-between;
  margin-bottom: 10px;
}
```

- It defines the top bar with time, score, and live information.
 - It uses flex to distribute the panels evenly.

```
.panel {
  background: rgba(255, 255, 255, 0.03);
  padding: 8px 12px;
  border-radius: 8px;
  font-weight: 600;
}
```

- It designs the boxes where data is displayed (time, score, etc.)
 - It features a semi-transparent background, rounded borders, and bolder text.

```
button {
  background: var(--accent);
  border: none;
  color: #072022;
  padding: 10px 12px;
  font-weight: 700;
  border-radius: 8px;
  cursor: pointer;
}
```

- Main style for the buttons (Start, Pause, Restart)
 - Yellow background (accent color)
 - "Hand" type cursor indicates that it is clickable.

```
button.ghost {
  background: transparent;
  border: 1px solid rgba(255,255,255,0.06);
  color: var(--text);
}
```

- "Transparent" variant for secondary buttons.
- It uses a border and light text to differentiate functions like Pause or Share.

Sidebar (.sidebar)

```
.sidebar {
  padding: 6px;
  display: flex;
  flex-direction: column;
  gap: 12px;
}
```

- ● It contains the description, game settings, and credits.
- It uses a vertical layout with space between blocks.

```
h1 {
  font-size: 18px;
  margin: 0 0 6px 0;
}
p.small {
  margin: 0;
  font-size: 13px;
  color: #b6c6d8;
}
```

- It defines the game title and explanatory text with a light tone and moderate size.
- It is ideal for comfortable reading on small screens.

Visual Elements for Cars

```
.car {  
  width: 72px;  
  height: 44px;  
  border-radius: 8px;  
  position: absolute;  
  left: 50%;  
  transform: translateX(-50%);  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  font-weight: 700;  
}
```

- Each .car is a colored rectangle that represents a vehicle.
- It is centered in its lane using transform: translateX(-50%)
- position: absolute allows them to be moved freely in the game area.

Car Colors

```
.car.red { background: linear-gradient(180deg, #ff4757, #d63b48); }  
.car.blue { background: linear-gradient(180deg, #3b82f6, #2563eb); }  
.car.green { background: linear-gradient(180deg, #34d399, #10b981); }  
.car.yellow { background: linear-gradient(180deg, #ffd166, #ffb74d); }
```

Each color identifies a different car using vertical gradients that give a sense of volume.

```
.car .wheel {  
  position: absolute;  
  border-radius: 50%;  
  bottom: -6px;  
  width: 14px;  
  height: 14px;  
  background: #111;  
}
```

- It creates the vehicle's wheels.
- They are small dark circles placed at the bottom of the car.

Car Colors

```
.car.red { background: linear-gradient(180deg, #ff4757, #d63b48); }  
.car.blue { background: linear-gradient(180deg, #3b82f6, #2563eb); }  
.car.green { background: linear-gradient(180deg, #34d399, #10b981); }  
.car.yellow { background: linear-gradient(180deg, #ffd166, #ffb74d); }
```

Each color identifies a different car using vertical gradients that give a sense of volume.

```
.car .wheel {  
  position: absolute;  
  border-radius: 50%;  
  bottom: -6px;  
  width: 14px;  
  height: 14px;  
  background: #111;  
}
```

- It creates the vehicle's wheels.
- They are small dark circles placed at the bottom of the car.

Text and Small Elements.

```
.small { font-size: 12px; color: #b9cbdc; }  
.footer {  
  display: flex;  
  gap: 8px;  
  align-items: center;  
  justify-content: space-between;  
}
```

- .small: used for secondary texts or descriptions.
- .footer: organizes the game credits and the share button at the bottom of the sidebar.

Small Screen Adaptation (Media Query)

```
@media (max-width: 900px) {  
  .wrap {  
    grid-template-columns: 1fr;  
    max-width: 98vw;  
  }  
  .sidebar {  
    order: 2;  
  }  
}
```

- It allows the game to adapt to mobile devices.
- It changes the column structure to a single column (the panel goes below the game area)
- It improves the experience on small screens like tablets or smartphones.

Conclusion

The CSS code for the “Reflexes: Cars” game fulfills a dual function:

- Attractive visual design without the need for external images.
- Adaptive and functional organization that maintains game clarity and performance.

Thanks to its use of flexbox, grid, CSS variables, and gradients, it achieves a modern, responsive, and entirely code-generated style.