

Gesällprov superdesignad och rolig hemsida

Erik Mörsell 2024-10-26

ermo2181

Sammanfattning

Essensen och det övergripande syftet med programmet var att skapa en visuellt intressant, rolig och interaktiv webbsida som skulle bestå av fyra essentiella delar: ett bildspel, en modal med överraskningsmeddelanden, en gem-animation samt en informationssida där användaren hypotetiskt kan fylla i ett formulär. Målet med hemsidan är att den ska erbjuda en dynamiskt visuell upplevelse, väcka nyfikenhet och engagemang, samt ge bra feedback till användaren vid interaktion. Denna design syftar till att skapa en upplevelse som liknar att spela ett spel, där varje interaktion belönar användaren och gör sidan levande.

Framtagande:

Idé och Inspiration:

Min tanke var att skapa en hemsida som var visuellt intressant, bestod av högupplösta bilder och var både rolig och intuitiv för användaren. Inspirationen kom delvis från tipset "Förfinad superdesignad webbplats" på kurswebbsidan, vilket gav mig inspiration att använda mycket av mer förfinad CSS-styling genomgående på webbplatsen. Jag började fundera på olika funktionaliteter som kunde implementeras i programmet, såsom roliga knappar och interaktiva objekt. Jag ville att varje element på sidan skulle bjuda in till interaktion, vilket skulle skapa en mer engagerande användarupplevelse.

Materialinsamling och Grundstruktur:

Nästa steg var att samla material i form av bilder, färgscheman, typsnitt och andra stilelement. Jag påbörjade också byggandet av grunden för det första HTML-dokumentet och de andra dokumenten som hemsidan skulle bestå av inledningsvis, för att skapa en stabil grund för programmet. Layouten blev en viktig del av processen, jag beslutade mig för att ha ett bildspel på startsidan som skulle leda till nästa interaktiva del längre ner på sidan. Jag strävade efter ett "spellikt" tankesätt där användaren skulle kunna navigera genom sidan som om de spelade ett spel. Detta steg la grunden för hela programmet.

Design och Experimentering

I detta steg experimenterade jag med olika färger, typsnitt och animationer för att hitta en balans som kändes både estetiskt tilltalande och funktionellt. Jag stötte på problem med att få animationerna att känna smidiga och integrerade. Jag stötte även på problemet att vissa knappar inte var tillräckligt intuitiva och inbjudande i sin design. Jag tyckte även att det inte var tillräckligt tydligt för användaren vad som förväntades av den. För att lösa dessa problem så sökte jag på YouTube och återgick till tidigare uppgifter i kursen, även W3Schools och kursböckerna var en bra källa för informationshämtning. Genom trial and error samt dessa källor lyckades jag förbättra designen av knappar och effekter, vilket ledde till en mer harmonisk användarupplevelse.

Exempel på förändringar:

- Jag började med standardanimeringar men anpassade dem senare för att skapa en "swing" effekt som kändes mer dynamisk.
- Typsnittet ändrades från ett vanligt sans-serif till ett mer lekfullt typsnitt, vilket bättre passade med det övergripande temat för sidan.
- Knappen som innehåller en nedåtpil som ligger över bildspelet på startsidan var först statisk, sedan förbättrade jag användarvänligheten för knappen genom att animera den så att den blev mer intuitiv att klicka på.
- Knappen "click for a surprise" var först statisk(stilla) men efter användartester(min flickvän) så var det uppenbart att knappen inte var tillräckligt tydlig för användaren att det var den knappen man skulle klicka på. Varför jag valde att även animera den knappen för att öka tydligheten och intuiteten.
- Likaså "gemes" på sidan "gems" var statiska från början men ändrades sedan till att bli mer intuitivt klickvänliga för användaren.
- Ett annat exempel på en förändring eller förbättring som skedde under kodningens gång är hanteringen av modalen och hur den stängs. Ursprungligen kunde modalen endast stängas genom att klicka på stäng-knappen (close Button). Efter vidare utveckling och förbättrade jag funktionen för att också kunna stänga modalen genom att klicka utanför den, vilket gör användarupplevelsen smidigare.

```
// Stäng modalen när stäng-knappen klickas
closeButton.addEventListener('click', function() { modal.classList.remove('show');
// Remove class to hide modal
});
```

Lades till:

```
// Stäng modalen när man klickar utanför den
window.addEventListener('click', function(event) {
if (event.target === modal) { modal.classList.remove('show'); // Remove class to hide modal }
});
```

Denna förändring gör interaktionen mer intuitiv och ger användaren fler alternativ för att stänga modalen, vilket skapar en bättre användarupplevelse.

Finslipning och Testning

I detta steg började jag finslipa programmet. Jag implementerade mer feedback från min testanvändare och justerade knapparnas responsivitet och hastighet på animationerna. Under finslipningen och testningen förbättrade jag även koden genom formattering och kommentering av kod. Efter användartestet tog jag även bort animationerna för de sex Surprise-boxarna, i en tidigare fas av utvecklingen var dessa boxar animerade/rörliga, vilket av användartestet kunde läras var något otydligt för användaren. Det uppfattades som att det var dessa boxar man skulle trycka på. Detta blev missvisande då ideen är att användaren ska trycka på knappen "Click for a surprise". Dessa animationer togs bort. HoverAnimationerna för boxarna behölls. I detta slutgiltiga steg la jag även till sidan info.html, en sida som fungerar som en informativ sida för hela webbplatsen. Jag la till den för att säkerställa hemsidans komplexitet.

Slutlig Genomgång och Presentation

I det sista steget gick jag igenom hela programmet för att säkerställa att alla funktioner fungerade som de skulle och att användarupplevelsen var sammanhängande och smidig. Jag gjorde en sista justering av animationernas tidsinställningar och såg till att alla länkar och interaktiva element var fullt fungerande. Vid presentationen av hemsidan för min testanvändare var jag noga med att se till så att hemsidan fungerade och användes i praktiken av en användare på det sättet som jag som utvecklare hade förväntat mig. Därmed kände jag mig tillfredsställd över att den faktiska responsen från en "riktig" användare stämde överens med min vision och idé som utvecklare och skapare av hemsidan. Presentationen blev en framgång.

Form

Dokumenten index.html, surprises.html, gems.html och info.html är fyra html-dokument. Dokumentet styles.css är ett dokument för styling med CSS där all styling för programmet existerar. script.js är dokumentet där all javascript existerar.

HTML-dokumenten:

Dessa dokument följer en standard HTML5 struktur. Detta kan avläsas med kodraden `<!doctype html>` vilket specificerar att dokumentet följer HTML-5. `lang="en"` i `<html>`-taggen indikerar att sidans huvudspråk är engelska. `<meta charset="utf-8">` säkerställer att webbsidan kan hantera text med olika tecken och språk genom UTF-8-kodningen. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`, denna kodrad gör webbsidan responsiv genom att anpassa sidans bredd till enhetens skärmstorlek och ställa in den initiala zoomnivån till 100 %. `meta name="robots" content="noindex,nofollow"` instruerar sökmotorer att inte indexera sidan eller följa länkarna på den, vilket kan vara användbart.

Vidare följer html-dokumenten grundstrukturen för html5 med:

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Title</title>
</head>
<body>
</body>
</html>
```

Ett exempel på central HTML kod från startsidan omskrivet till pseudokod är koden som ligger inom main-blocket:

```
main-sektion med ID "next-section"
  header-sektion
    rubrik-nivå 1: "Welcome to this playground!"
    stycke-text: "Click the buttons below to go explore some fun stuff!"
    länk till "surprises.html" med klass "page-link"
```

länk-text: "Surprises😊"
länk till "gems.html" med klass "page-link"
länk-text: "Gems🥳"

Eftersom att detta program har en ökad komplexitet genom att den har flera dokument samt all CSS styling utanför html-dokumentet i ett separat css.styles dokument så behövs det en länk `<link rel="stylesheet" href="styles.css">` som används för att definiera utseendet på html element. Html har även en länk `<script src="script.js"></script>` som länkar till den externa filen script.js som ansvarar för dynamiska funktioner för programmet, som animationer och bildspel. Scriptet laddas in i slutet av sidan, vilket är en bra praxis eftersom det gör att sidan laddas snabbare.

Grundstrukturen för CSS dokumentet kan beskrivas enligt följande:

Det innehåller stilsättning för globala element som html och body, inställningar för bild och behållare, stilar för rubriker och text, knappdesign, sektioner och layout, modal och interaktioner, media queries, formulär och listor. Koden utformar en responsiv och visuellt tilltalande webbsida med en konsekvent stil, där olika komponenter som bilder, rubriker, knappar, och formulär är noggrant designade för att skapa en engagerande användarupplevelse.

Ett exempel på ett centralt kodavsnitt omskrivet till pseudokod från styles.css dokumentet är knapparna och hur de är utformade.

```
.scroll-button
display: block
width: 80px
height: 80px
background-color: orange-röd färg
color: vit färg
font-size: stor textstorlek
text-align: center (centrera texten horisontellt)
line-height: samma som höjd (vertikal centrerings av text)
border-radius: cirkulär form
position: absolut
bottom: 500px från botten
left: mitten på sidan
transform: flytta horisontellt så knappen är centrerad
text-decoration: ingen understrykning
transition: mjuk övergång för bakgrundsfärg, position, och skugga vid hover
animation: fade in och uppåt med repetition
```

```
.scroll-button:hover
background-color: mörkare orange-röd
transform: flytta uppåt och centrera horisontellt
box-shadow: applicera en lätt skugga runt knappen
```

```
.page-link
display: inline-block (för att lägga till padding)
```

```
background-color: orange-röd färg  
color: ljus rosa färg  
padding: stor inre marginal  
text-decoration: ingen understrykning  
font-size: större textstorlek  
font-family: Comic Sans MS eller liknande, sans-serif  
border-radius: rundade hörn  
margin: yttre marginal  
border: tjock gul kantlinje  
box-shadow: flera skuggor, inklusive en gul yttre glöd  
transition: mjuk övergång för bakgrundsfärg, position, skala, och skugga  
letter-spacing: något ökat bokstavsavstånd  
position: relativ  
z-index: överliggande lager
```

```
.page-link:hover  
background-color: något mörkare orange-röd  
transform: lyft uppåt och öka storlek  
box-shadow: fördjupad och utökad skugga, samt ytterligare glöd i orange-röd och gul
```

```
.action-button  
background-color: gul  
color: mörk orange-röd  
border: ingen kantlinje  
padding: stor inre marginal  
font-size: stor textstorlek  
border-radius: rundade hörn  
cursor: pekande hand vid hover  
box-shadow: lätt skugga runt knappen  
animation: puls som upprepas  
transition: mjuk övergång för bakgrundsfärg, skala och skugga
```

```
.action-button:hover  
background-color: mörk orange-röd  
color: vit  
transform: ökad storlek  
box-shadow: djupare och starkare skugga
```

Dessa knappar är avgörande för användarens navigering och interaktion med webbplatsen, vilket gör dem centrala i dokumentets funktionalitet och stil

PseudoKod för script.js:

```
// Program: ImageSlideshowAndModal
```

```
// Beskrivning: Hanterar ett bildspel, modal-funktionalitet och gem-animation på en webbsida

// Definiera en array med bildfilnamn
images = ["images/bild19.jpg", "images/bild18.jpg", "images/bild20.jpg"]

// Välj bildelementet för bildspelet
slideshowImage = hämta element med ID "slideshow"

// Initiera en räknare för att hålla reda på aktuellt bildindex
currentImageIndex = 0

// Funktion för att byta bild var 4:e sekund
function changelImage
    // Öka det aktuella bildindexet
    currentImageIndex = currentImageIndex + 1

    // Om indexet överskider antalet bilder
    if currentImageIndex >= längd av images then
        // Återställ indexet till 0
        currentImageIndex = 0
    end if

    // Byt bildkällan till nästa bild
    slideshowImage.src = images(currentImageIndex)
end function

// Ställ in ett intervall för att byta bild var 4:e sekund (4000 millisekunder)
starta timer(changelImage, 4000)

// Modal-funktionalitet för åtgärdsknappen
actionButton = hämta första element med klassen "action-button"
modal = hämta element med ID "customModal"
closeButton = hämta första element med klassen "close-button"
modalMessage = hämta element med ID "modalMessage"

// Om actionButton existerar
if actionButton != null then
    // Lägg till klick-händelse för åtgärdsknappen
    actionButton.addEventListener("click", function()
        // Definiera en array med överraskningsmeddelanden
        surprises = [
            "🎉 You've unlocked a secret!",
            "🎈 Here's a balloon!",
            "✨ You've found a star!",
            "🔥 It's getting hot!",
            "🌈 You're on a rainbow!",
            "☀ Shine bright!"
        ]
    
```

```

// Välj ett slumpmässigt överraskningsmeddelande
randomSurprise = välj slumpmässigt element från surprises

// Sätt meddelandet i modalen
modalMessage.textContent = randomSurprise

// Visa modalen
modal.classList.add("show")
end function)

// Stäng modalen när stäng-knappen klickas
closeButton.addEventListener("click", function()
    modal.classList.remove("show")
end function)

// Stäng modalen när man klickar utanför den
window.addEventListener("click", function(event)
    if event.target == modal then
        modal.classList.remove("show")
    end if
end function)
end if

// Lägg till klick-händelse för gem
gems = hämta alla element med klassen "gem"
for each gem in gems do
    gem.addEventListener("click", function()
        gem.classList.add("clicked") // Lägg till klass för animation
        vänta(900) // Justera för att matcha längden på animationen
        gem.classList.remove("clicked") // Ta bort klass efter animationslängden
    end function)
end for

```

Denna pseudokod för javascriptfilen beskriver den centrala funktionaliteten i dokumentet script.js.

Bildspel: Genom att använda en array och en räknare cyklar programmet genom bilder. Intervallinställningen gör att byten sker automatiskt.

Modal-funktionalitet: Med hjälp av event listeners kan användaren öppna och stänga modalen. Det finns en dynamisk del där ett meddelande slups fram för att skapa en överraskningseffekt.

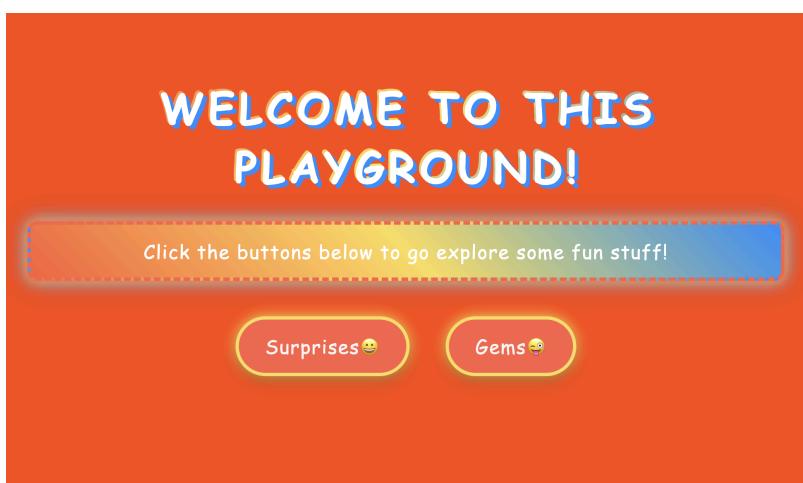
Gem-animation: Koden hanterar användarinteraktioner genom att lägga till och ta bort klasser som styr animationer, vilket gör användarupplevelsen mer engagerande.

Funktion

Startsida:

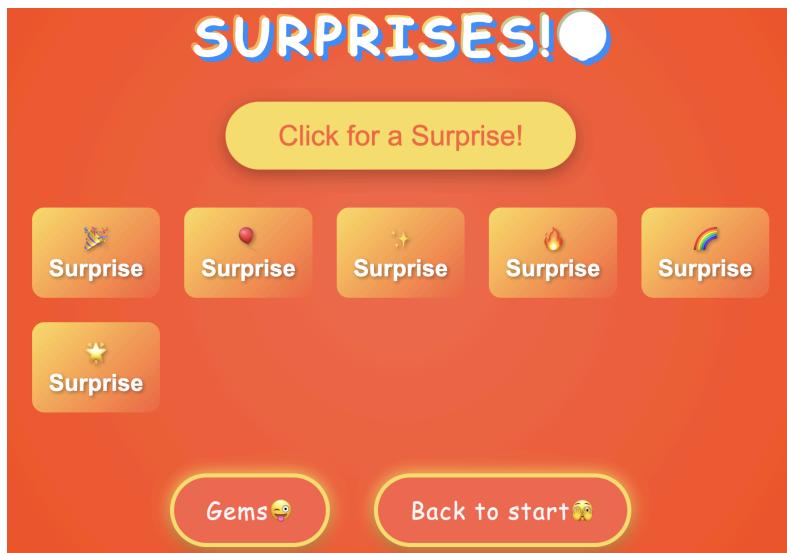


På startsidan möts användaren av ett bildspel som växlar mellan tre bilder. En animerad, interaktiv pil syns i mitten av skärmen för att hjälpa användaren att förstå att den ska tryckas på, att bläddra ner på sidan fungerar också.

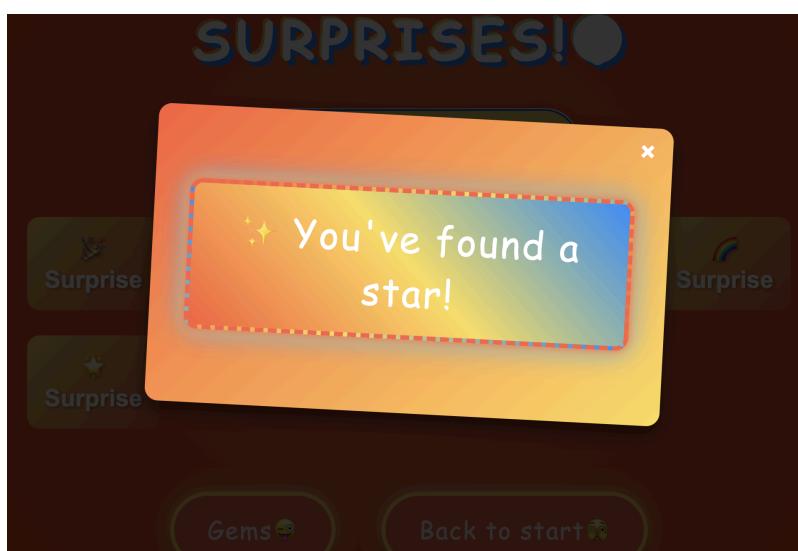


Efter en mjuk transition så kommer man ner till "nästa" del av startsidan, där två knappar visar sig. Användaren får välja vilken av knapparna man vill trycka på, om man vill komma till "Surprises" eller "Gems".

Surprises



Efter att ha tryckt på knappen Surprises kommer man till en ny sida. Här är knappen animerad för att visa för användaren vart man ska klicka för att få en "Surprise".



När knappen trycks på slumpas en överaskning av de sex existerande överaskningarna. Meddelandet visas som en modal som även den är animerad och stylad på ett roligt sätt. Användaren kan sedan antingen kryssa ner modalen alternativt trycka utanför den för att återgå till att kunna trycka på "click for a surprise"-knappen om och om igen.

Från denna sida kan användare välja att gå till nästa sida eller att gå tillbaka till startsidan.

Vid navigation till "Gems":



Kommer man till denna sida. Här kan användaren trycka på de olika gemen för att se "magin". Gemen animeras i en katapultliknande effekt för att sedan återgå till den ursprungliga läget. Här kan användaren trycka runt på gemen för en rolig och interaktiv upplevelse. Jag själv som utvecklare tycker att detta simpla, dock spelliknande gemit-spel är oväntat roligt och tillfredsställande. Man kan till exempel försöka ha så många olika gemen i "luften" samtidigt innan de är tillbaka. Vilket bidrar till ett något actionfyllt spelliknande äventyr.

Även här kan användaren välja att gå tillbaka till startsidan eller att prova "surprises" igen.

Footer

© 2024 Have Some Fun! | [More Info](#)

Längst ner på varje sida finns en footer med en länk till sidan "More info". Vid klick:

Info

[Back to start!](#)

This site was made by Erik Mörsell

This page was made for a few purposes

- Having fun
- Taking a break from studies
- Getting inspired!

Mainly for these benefits

1. Having fun!
2. Being happy!
3. Having a laugh!

Have Any Ideas for this website? Contact Us! (Example Form)

Name:

Email:

Message:

[Send Message](#)

Kommer användaren till en info-sida med information om sidan, olika listor med syftet och anledningen till varför sidan existerar. Det finns även ett formulär som användare kan fylla i för att ge feedback till skaparna av sidan. Formuläret skickas inte någonstans i denna version.