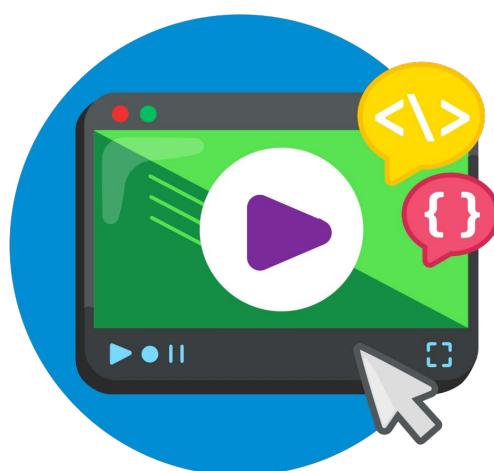




MATERIAL DO PROFESSOR

Lógica de programação:
criando arte interativa com p5.js





1

Introdução

O Material do Professor é um instrumento que oferece o suporte e o embasamento necessários para a condução dos estudantes nesta jornada de aprendizagem.



Aqui, você encontrará orientações pedagógicas para o trabalho com cada unidade, bem como os objetivos de aprendizagem, os conteúdos e habilidades, as estratégias didáticas utilizadas na construção da unidade e, por fim, materiais aula a aula que servirão de apoio ao seu planejamento e durante o processo avaliativo dos estudantes. Utilize o Material do Professor como guia durante o planejamento das aulas, adaptando as estratégias de acordo com as necessidades e interesses dos seus estudantes.

Este material é digital e pode sofrer atualizações periodicamente. Certifique-se de verificar regularmente a plataforma para baixar as versões mais recentes e garantir que você tenha acesso às informações mais atualizadas e relevantes para suas aulas.

2

Conteúdo programático

Olá, professor(a)!

Nesta unidade, os estudantes serão apresentados à ferramenta p5.js. Para isso, serão guiados a construir dois projetos artísticos. O primeiro, *Monalisa digital*, consiste na criação de um rosto utilizando formas geométricas, com base no quadro Monalisa. Já no segundo projeto, os estudantes farão uma transposição da brincadeira popular de quente e frio. Isso será feito por meio da criação de um ponto oculto na tela, que deverá ser encontrado por meio de dicas, assim como fazemos na brincadeira popular. Para as dicas, utilizaremos os conceitos presentes no teorema de Pitágoras.

Temática da unidade: pensamento criativo e computacional

Ferramentas utilizadas: p5.js

Formato de entrega do projeto: código no editor p5.js

Quantidade de aulas: 8

Objetivos de aprendizagem:

- Compreender que o p5.js é uma biblioteca de criação de projetos criativos;
 - Localizar as coordenadas X e Y na tela para posicionar elementos geométricos no canvas do p5.js;
 - Aplicar a função *map()* para relacionar as posições do mouse com limites específicos em um projeto criativo;
 - Utilizar estruturas condicionais na criação de um algoritmo;
 - Compreender que a menor distância entre dois pontos é calculada a partir do teorema de Pitágoras.
-

Habilidades:

- Itinerário Formativo

(EMIFMAT04) Reconhecer produtos e/ou processos criativos por meio de fruição, vivências e reflexão crítica na produção do conhecimento matemático e sua aplicação no desenvolvimento de processos tecnológicos diversos.

- BNCC - Computação

(EM13CO02) Explorar e construir a solução de problemas por meio de refinamentos, utilizando diversos níveis de abstração desde a especificação até a implementação.

(EM13CO18) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas e usando diversos artefatos computacionais.

2

Conteúdo programático

AULA

CONTEÚDO

OBJETIVO DE APRENDIZAGEM

1

- Fundamentos de programação em JavaScript utilizando a biblioteca p5.js;
- Utilização de funções básicas, como *setup()* e *draw()*;
- Manipulação de coordenadas em formas geométricas no p5.js.

1. Compreender os conceitos básicos de funções em JavaScript;
2. Aplicar comandos básicos de desenho no P5.js, como *createCanvas()* e *circle()*;
3. Modificar parâmetros de comandos em p5.js para ajustar a posição e o tamanho de formas geométricas.

2

- Recuperação e gerenciamento de projetos no editor p5.js;
- Uso de comandos de desenho (*line()* e *triangle()*) para criar formas geométricas específicas;
- Manipulação das coordenadas X e Y para posicionar elementos em um plano cartesiano.

1. Aplicar comandos de desenho básicos, como *line()* e *triangle()* no p5.js;
2. Criar novas formas e elementos visuais (ex.: boca, nariz e orelhas) na personagem, utilizando as coordenadas do plano cartesiano;
3. Relembrar o processo de login e recuperação de projetos anteriores no editor p5.js.

3

- Cores predefinidas e hexadecimal no p5.js;
- Utilização de outros comandos para criar mais formas geométricas;
- Comentários para documentação de código.

1. Aplicar conceitos de cores em código usando o P5.js;
2. Criar novas formas e elementos visuais (ex.: sobrancelhas e pupilas) na personagem, utilizando as coordenadas do plano cartesiano;
3. Documentar o código com comentários claros e organizados.

4

- Comando *map()* para limitar o movimento de elementos na tela;
- Conceito de variáveis em programação;
- Práticas de declaração de variáveis e organização de código para evitar erros em animações.

1. Aplicar conceitos de mapeamento para restringir a movimentação de elementos gráficos na tela;
2. Analisar a lógica de posicionamento e ajuste de variáveis para criar interações visuais precisas;
3. Criar uma animação interativa na qual elementos gráficos seguem a posição do cursor do mouse.

AULA

CONTEÚDO

OBJETIVO DE APRENDIZAGEM

5

- Utilização do comando `random()` para sortear um valor aleatório;
- Diferenciação entre números inteiros e decimais;
- Corrigir bugs a partir de pequenos testes.

1. Aplicar conceitos de coordenadas aleatórias para posicionar elementos na tela de um projeto;
2. Compreender a diferença entre números inteiros e decimais em operações de comparação;
3. Testar e depurar o código utilizando `console.log()`.

6

- Uso do operador lógico (`e`);
- Introdução ao conceito de distância em um plano cartesiano;
- Aplicação do Teorema de Pitágoras.

1. Aplicar operadores lógicos para combinar múltiplas condições em uma estrutura condicional;
2. Compreender o conceito de distância e sua aplicação em projetos interativos;
3. Implementar o Teorema de Pitágoras para calcular distâncias em um plano cartesiano.

7

- Ajuste do tamanho de um círculo baseado na distância do cursor ao ponto oculto;
- Uso da função `dist()` no p5.js;
- Comparação de métodos de cálculo de distância, incluindo o Teorema de Pitágoras e a função `dist()`.

1. Compreender o uso da função `dist()` no p5.js para calcular distâncias entre dois pontos;
2. Aplicar o conceito de distância para ajustar o comportamento visual de elementos interativos;
3. Associar diferentes abordagens para calcular distâncias.

8

- Identificação e remoção de código redundante ou desnecessário;
- Uso do comando `constrain` no p5.js;
- Ajuste da dificuldade de um jogo interativo.

1. Revisar o código para identificar e remover variáveis e comandos desnecessários;
2. Aplicar o comando `constrain` para limitar o movimento de um objeto dentro de um plano cartesiano;
3. Modificar a dificuldade de um projeto interativo por meio da modificação de parâmetros de distância.

3

Orientações pedagógicas

Pensando em prepará-lo(a) para a condução das aulas, construímos uma formação sobre esta unidade com o objetivo de fornecer mais insumos para sua prática docente. Você pode acessá-la na plataforma da Start.

Para apoiar o desenvolvimento desta unidade, estimule a familiarização com o p5.js, incentivando a exploração de comandos básicos de desenho e das coordenadas do plano cartesiano. Para o projeto *Monalisa*, apresente o contexto da obra, destacando sua história e características marcantes, como o famoso olhar que parece seguir o observador. Mostre uma versão inicial do projeto para inspirar os estudantes a personalizarem suas criações, utilizando formas geométricas para desenhar o rosto e os olhos.

Ao longo do desenvolvimento do projeto, apresente o uso de cores e animações, ensinando a aplicação de funções como *fill()* e incentivando o uso do site [**Paleta de cores**](#) para criar combinações harmônicas. Oriente os estudantes a documentarem o código e experimentarem diferentes ordens de comandos, ajudando-os a compreender como isso impacta o resultado visual. Explique o conceito de variáveis e sua aplicação em animações dinâmicas, como o movimento dos olhos seguindo o cursor do mouse. Reforce o entendimento utilizando analogias simples, como a ideia de um estacionamento, para ilustrar o armazenamento de valores.

Na segunda etapa da unidade, com o projeto *Quente e frio*, inicie contextualizando a brincadeira tradicional e conecte-a ao conceito de números aleatórios para calcular distâncias e criar interações. Explique o funcionamento da função *random()* e a aplicação do Teorema de Pitágoras para medir a proximidade entre a personagem e o ponto escondido. Reforce a importância da lógica condicional, mostrando como ajustar a dificuldade do jogo com variações de parâmetros. Proponha desafios práticos para que os estudantes testem seus códigos, aprimorando a jogabilidade.

Finalize a unidade promovendo o compartilhamento dos projetos na plataforma, permitindo que os estudantes apresentem suas criações e recebam feedbacks dos colegas. Proponha uma reflexão coletiva sobre as estratégias utilizadas e as dificuldades enfrentadas durante o desenvolvimento dos projetos. Encoraje melhorias, como ajustes na jogabilidade ou inclusão de elementos criativos. Essas atividades consolidam os aprendizados, estimulam o pensamento crítico e reforçam a importância da colaboração no processo de programação.

4

Materiais de apoio

Para auxiliar na condução das aulas desta unidade, disponibilizamos materiais de apoio aula a aula para professores e estudantes. Esses materiais podem ser acessados na plataforma Start, no campo “Material de apoio” de cada aula.

4.1 Professores

O **guião do professor** possui o passo a passo descritivo da aula, bem como os comandos necessários para a construção do projeto. Ele pode ser utilizado para o planejamento e preparação da aula, contando com direcionamentos pedagógicos, dicas para condução das aulas, explicação de conceitos de programação e soluções para possíveis dificuldades que os estudantes possam ter. Ao final de cada guia, há uma explicação para o desafio da aula, contando com a expectativa de resolução por parte dos estudantes.

O **gabarito do professor** possui a resolução comentada dos exercícios que os estudantes devem realizar em cada aula. Essas atividades funcionam como um complemento à prática de programação pela construção de projetos, fornecendo uma devolutiva rápida ao professor sobre a proficiência dos estudantes.

4.2 Estudantes

Os **slides dos estudantes** contêm um passo a passo descritivo, aula a aula, dos comandos que os estudantes deverão realizar para a construção do projeto da unidade. Por se tratar de um direcionamento da aula, ele pode ser disponibilizado individualmente ou projetado para a turma. Ao final de cada slide, há um desafio para aprofundar os conteúdos que foram trabalhados naquela aula, sendo uma ferramenta importante para a prática da programação e para fornecer uma atividade extra para aqueles estudantes que finalizarem mais rapidamente a construção dos códigos.

As **listas de exercícios dos estudantes** são um conjunto de atividades de fixação dos conteúdos que foram trabalhados em aula. É esperado que, a cada aula, os estudantes realizem os exercícios e tirem eventuais dúvidas sobre eles com o professor.

4

Materiais de apoio

4.3 Rubrica de avaliação

Por fim, a rubrica é uma valiosa ferramenta para auxiliar na correção dos projetos que são submetidos pelos estudantes. Elas possuem quatro características a serem avaliadas de acordo com cada projeto da unidade e, para cada característica, há a descrição de quatro níveis de proficiência em que o estudante pode se encaixar. Dessa forma, ao avaliar um projeto, o professor consegue analisar o que é esperado em cada nível, fazendo com que sua correção seja baseada em critérios claros e que o feedback de desenvolvimento para o estudante seja melhor direcionado. Essas rubricas também podem ser acessadas no painel de gestão.

Acesse aqui a rubrica dos projetos:

[**Projeto 01 - Monalisa**](#)

[**Projeto 02 - Quente e frio**](#)



Minhas anotações

Utilize o seu editor de PDF para realizar anotações neste espaço.



start

by alura



Contato:

Suporte

@startbyalura

www.startalura.com.br