

Shaolin Shapes: Jogo Educacional de Geometria

Alauan Travain Darwich da
Rocha
alauantdr@gmail.com
Universidade Federal de Santa
Catarina
Brasil

Artur Frederico Heiber
arturfedericoheiber@gmail.com
Universidade Federal de Santa
Catarina
Brasil

Douglas Lopes Maltezo Saiter
douglaslmsaiter@gmail.com
Universidade Federal de Santa
Catarina
Brasil

Gabriela Guse Machado
gabriela_guse@gmail.com
Universidade Federal de Santa
Catarina
Brasil

Lucas Gabriel Barcaro
lucasbarcaro@gmail.com
Universidade Federal de Santa
Catarina
Brasil

Pedro Santos Pozza
pedrospozza@gmail.com
Universidade Federal de Santa
Catarina
Brasil

RESUMO

Este trabalho apresenta o desenvolvimento do jogo educativo "Shaolin Shapes", voltado ao ensino de geometria plana para crianças entre 8 e 12 anos. O jogo utiliza narrativa interativa, sensores de movimento e princípios pedagógicos para ensinar conceitos como ângulos, classificação de triângulos e simetria. A metodologia Agile-Stage-Gate foi empregada para organizar o processo de desenvolvimento. No artigo em questão são discutidos os requisitos, a arquitetura do sistema e os resultados preliminares da primeira fase do projeto.

KEYWORDS

Educação, Geometria, Jogos Educativos, Realidade Física, Sensor de Movimento, Ensino Fundamental

1 INTRODUÇÃO

No período da Segunda Sofística, a principal forma de ensino conhecida era a Paideia, que abrangia o ensino de música, matemática, ginástica, entre outros saberes. A palavra Paideia tem origem grega e significa "a formação da criança". Essa concepção educativa buscava preparar todas as crianças em três aspectos fundamentais: capacitar o indivíduo para garantir seu sustento, formar cidadãos conscientes de seus deveres e direitos em uma democracia, e incentivar o aprendizado contínuo ao longo da vida [1]. Dentre os saberes ensinados nesse contexto, este projeto busca focar na matemática, ciência que contribui não apenas para o desenvolvimento social, mas também para a formação do raciocínio lógico, da capacidade de resolver problemas e da atuação crítica no mundo [2].

A matemática é uma ciência que possui diversas áreas, dentre elas a geometria, da qual tem como objetivo o ensino das formas dos objetos e as suas relações, ajudando no desenvolvimento da manipulação de imagens mentais [10].

Tendo em vista o supracitado, foi realizado um brainstorming sobre como seria possível ajudar a promover um ensino melhor para este conteúdo, o resultado foi a idealização de um jogo que visa integrar diversão com aprendizado. Para concepção de tal, foi utilizada uma abordagem em diferentes frentes, envolvendo conhecimentos de software de baixo e alto nível, hardware, modelagem geométrica, design e pedagogia. Dada a multidisciplinaridade, foi utilizada a

metodologia híbrida agile stage-gate, que consiste em combinar a estrutura sequencial e controlada da metodologia de projeto stage-gate com a flexibilidade e entrega iterativa da metodologia Agile [3], [8].

A utilização de jogos educativos que envolvem interação física, como o uso de controles com sensores de movimento, contribui significativamente para a aprendizagem ao integrar habilidades motoras ao processo cognitivo. Segundo a [5], jogos permitem que os alunos assimilem conceitos de forma significativa, principalmente quando inseridos em contextos práticos e dinâmicos. Ademais, quando o corpo é envolvido na experiência de aprendizagem, como no caso de jogos que exigem gestos e movimentos coordenados, há um aumento da memorização e da compreensão dos conteúdos [9]. Essa abordagem favorece também o desenvolvimento de habilidades interdisciplinares e engajamento, promovendo uma experiência lúdica que estimula tanto o raciocínio quanto a coordenação motora [7] [11].

2 OBJETIVO

O objetivo do trabalho é a realização de um projeto que engloba diversas áreas do conhecimento de engenharia mecatrônica, promovendo a integração de conceitos já abordados durante a graduação. O projeto em questão se trata de um videojogo educacional voltado para o público infantil - de 8 a 12 anos - que ensina tópicos relacionados à geometria elementar de uma maneira interessante e divertida.

O jogo deve ilustrar didaticamente os conteúdos mencionados em adequação com a matriz curricular vigente para o ensino fundamental. O propósito é construir o conhecimento a respeito de geometria de maneira gradual usando uma narrativa que chame a atenção do jogador. Também é realizada uma avaliação contínua do aluno ao passo que ele avança os níveis pré-estabelecidos, sendo possível aferir a qualidade do ensino que está sendo ofertado.

Além do quesito pedagógico, o videojogo deve apresentar uma jogabilidade ergonômica e confortável aliada a um desempenho de software e hardware satisfatório, de modo a tornar a experiência agradável para o usuário. O trabalho também visa aprimorar a capacidade de trabalho em grupo dos alunos. O uso de princípios clássicos de desenvolvimento de projeto, como práticas de metodologia ágil, foram utilizadas objetivando o aumento da produtividade e da coesão entre os setores envolvidos no trabalho.

3 METODOLOGIA

Neste capítulo, será apresentada a concepção do jogo educativo, bem como o método de organização aplicado pela equipe.

3.1 Inspiração

O software de alto nível consiste em uma adaptação do jogo Fruit Ninja. A dinâmica do jogo original utiliza figuras de frutas que surgem na tela de forma ascendente, simulando um lançamento para o alto, seguidas por sua queda e desaparecimento. O jogador deve realizar um movimento de "corte" nas frutas enquanto elas estão visíveis, utilizando a tela sensível ao toque (touchscreen). Nesta adaptação, as frutas foram substituídas por figuras geométricas planas, de diferentes formatos. A ação de corte será executada por meio de um hardware externo, desenvolvido em forma de controle de videogame, inspirado no controle do Nintendo Wii. Este dispositivo é equipado com um giroscópio, responsável pela captação dos movimentos realizados pelo jogador. Esses movimentos serão utilizados para guiar o cursor na tela, permitindo que o usuário interaja com os elementos do jogo por meio de gestos físicos.

3.2 O Jogo

O jogo inicia com uma breve narrativa, na qual um robô ensina kung fu ao jogador, que deve acertar os objetos lançados como forma de treino. Em seguida, o jogador é inserido em uma jornada para completar as fases propostas, cada uma com o objetivo de reforçar o aprendizado de conceitos relacionados à geometria plana. À medida que o jogador avança, novos conteúdos, com complexidade crescente, são introduzidos. No total, o jogo é composto por cinco fases, com foco nos seguintes tópicos:

- (1) Classificação de triângulos: equiláteros, isósceles, escalenos e retângulos;
- (2) Identificação dos tipos de ângulos internos: agudo, reto e obtuso;
- (3) Análise do número de arestas de uma figura geométrica com base na quantidade de lados;
- (4) Reconhecimento de simetria ou assimetria nas formas geométricas;
- (5) Diferenciação entre formas geométricas côncavas e convexas.

3.3 Método de organização aplicado

Para a realização deste trabalho, a equipe foi dividida em três setores: software de alto nível; eletrônica e software de baixo nível; e modelagem física. Essa divisão permitiu que os integrantes se concentrassem em suas respectivas tarefas e habilidades. No entanto, para garantir que o produto final fosse coeso e que as partes se integrassem adequadamente, um dos membros ficou responsável pela gestão da integração entre os setores, promovendo a comunicação contínua entre as áreas para viabilizar a criação de um jogo funcional.

Com o objetivo de manter a organização durante a execução das atividades, este projeto adotou uma metodologia ágil, a qual permite o acompanhamento das tarefas semanais pela equipe, além de oferecer maior flexibilidade na execução. Essa abordagem mostrou-se adequada ao contexto do grupo, composto exclusivamente por

estudantes de engenharia, cujas rotinas demandam uma gestão adaptável e eficiente.

O framework aplicado neste projeto foi o Scrum, que, segundo [3], é composta por três elementos principais: papéis, artefatos e ferramentas. Esses elementos contribuem significativamente para o senso de pertencimento da equipe, aumento da motivação, melhora na comunicação e compartilhamento do conhecimento.

- Sprints com tempo determinado, que organizam o trabalho em ciclos curtos e produtivos;
- Reuniões diárias, adaptadas para uma frequência semanal, considerando a disponibilidade da equipe;
- Reuniões de retrospectiva (retrospective meetings), nas quais se avaliam os avanços e possíveis melhorias do processo.

As ferramentas utilizadas foram:

- Backlogs do produto e do sprint, que registram as tarefas a serem desenvolvidas;
- Quadro Scrum, que visualiza o andamento das atividades;
- Gráficos de burndown, utilizados para acompanhar o progresso dos sprints.

Quanto aos papéis, a metodologia define:

- O Product Owner, responsável por estabelecer as prioridades do projeto — função assumida pelos professores orientadores;
- O Scrum Master, que garante a aplicação da metodologia e a remoção de impedimentos;
- A Equipe de desenvolvimento, responsável pela execução das tarefas técnicas.

Para viabilizar a aplicação da metodologia Scrum, foi criado um grupo no site GitHub, utilizado como repositório central dos códigos, imagens e textos produzidos pela equipe. Além disso, utilizou-se a funcionalidade de projetos da plataforma, que permite a construção de quadros Scrum e o gerenciamento visual dos sprints atribuídos a cada membro.

Como ilustrado na Figura 1, o quadro foi organizado em quatro colunas: (i) To do (a fazer); (ii) In progress (em andamento); (iii) In review (em revisão); e (iv) Done (concluído). Cada tarefa exibe não apenas o seu título, mas também o setor responsável, a prioridade e o tamanho estimado, o que facilita o controle e o acompanhamento do progresso das atividades pela equipe.



Figure 1: Quadro Scrum da equipe no GitHub Projects, com visualização das tarefas organizadas por estágio, prioridade, setor e estimativa de esforço. Elaborado pelos autores.

4 RESULTADOS PRELIMINARES E DISCUSSÃO

O projeto atingiu a construção do protótipo do jogo com fase introdutória, menu funcional e conexão entre software e hardware por

Bluetooth. A ergonomia e precisão do controle foram validadas em protoboard.

4.1 Software de Alto Nível

O desenvolvimento foi realizado utilizando a biblioteca Pygame, que facilita a manipulação gráfica e o gerenciamento de eventos, como a movimentação de objetos e a detecção de cliques, sendo ideal para aplicações de jogos interativos. Inspirado na mecânica do jogo Fruit Ninja, o objetivo inicial foi gerar objetos geométricos na tela, atribuir movimento a eles e implementar a verificação de cliques do usuário para simular o impacto dos objetos pelo clique. A partir dessa base, o projeto evoluiu por meio da programação orientada a objetos, com a criação de diferentes classes e estruturas de controle de estado, organizando o código de maneira modular. Essa estrutura favorece a legibilidade, manutenção e futura expansão. Até o momento, as seguintes funcionalidades já foram implementadas com sucesso:

- Menu de seleção de fases: Navegação entre fases com controle via mouse, sendo possível acessar o tutorial (fase básica) já desenvolvida.
- Identidade visual com botões em pixel art.
- Aparência visual diferenciada por fase: Com estilos distintos para cada etapa do jogo.
- Movimento dinâmico das formas geométricas pela tela.
- Animações visuais ao serem cortadas proporcionando um feedback imediato ao jogador.
- Remoção de formas após o corte e geração de novas para manter a dinâmica do jogo.
- Contador de cortes corretos por tipo de forma para monitor o progresso do jogador.
- Contador de erros para registrar cortes incorretos.

Em paralelo, o repositório do projeto no GitHub já disponibiliza vídeos demonstrativos do jogo em funcionamento, permitindo uma visualização clara da situação atual do desenvolvimento.

Atualmente, o sistema contabiliza os cortes corretos e incorretos, mas será necessário implementar a lógica que, ao atingir um número limite de erros, redirecione o jogador a uma tela explicativa da fase e reinicie o nível. Essa abordagem visa reforçar o aprendizado por repetição, com uma experiência mais educativa. Da mesma forma, ao atingir o número correto de acertos para completar a fase, o jogador será redirecionado automaticamente ao menu de fases para iniciar a próxima etapa do jogo.

Em conjunto com o desenvolvimento dos próximos passos do software, está prevista a integração do controle físico desenvolvido na protoboard com o jogo, o que permitirá testes mais precisos da experiência final que será oferecida aos usuários.

Essa integração entre software e hardware marca o início da fase de testes conjuntos com sensores e botões, consolidando o caráter multidisciplinar do projeto e aproximando a aplicação do cenário real de uso.

1

4.2 Software de Baixo Nível e Hardware

Neste momento o hardware do projeto se encontra em período inicial de testes isolados, sem a integração com o software de alto nível. Os testes são realizados em um protótipo montado sobre a placa de ensaio, com todos os componentes eletrônicos já incorporados.

Os experimentos realizados constataram sucesso na obtenção de dados do giroscópio e na comunicação bilateral via Bluetooth com o ESP32, sendo possível transmitir dados de posição em tempo real para o software de baixo nível, além de receber comandos para acionamento do motor de vibração.

Ainda está sendo avaliado maneiras de melhorar a responsividade do sistema, aumentando a taxa de dados enviada ao software de baixo nível para um acoplamento mais fluido entre jogo e controlador, mesmo que resultados satisfatórios já tenham sido apresentados.

Em seguida, será realizado o transporte dos componentes eletrônicos para a placa de circuito impresso definitiva, possibilitando o encapsulamento no modelo mecânico que está sendo desenvolvido. Objetiva-se também integrar o hardware com o software de alto nível para iniciar os testes e validar o acoplamento entre as partes do projeto.

Nesse ínterim, será necessário avaliar o desempenho global do sistema quando exposto às condições de computação paralela entre o software de alto e baixo nível. Espera-se uma ligeira queda de performance visível ao usuário, que terá de ser tratada.

4.3 Modelagem Geométrica

No presente instante, foram adquiridas modelagens geométricas dos componentes levantados pelo setor de eletrônica por meio de sites de compartilhamento de projetos open-source como o grabcad. Aqueles que não possuem modelagem disponível, foi adquirido o componente físico e, por meio de ferramentas de medição como o paquímetro, modeladas em CAD.

Para o projeto preliminar, foi desenvolvido uma carcaça de plástico inicialmente projetada para construção via manufatura aditiva, como impressão 3D, baseada em um controle de videogame já existente, o Nintendo Wii. Esta modelagem passou por três grandes mudanças: inicialmente seriam utilizadas duas células de li-ion 18650 para alimentação dos componentes eletrônicos, porém ao prototipar o primeiro controle, foi observado que ficaria significativamente maior que o esperado, portanto, a partir de reuniões de sincronização entre as equipes, foi acordado a mudança para uma bateria de 9V que, por cálculo, se mostrou suficiente para alimentação. Com esta mudança, houve uma reformulação da modelagem do controle para melhor adaptar aos componentes. A terceira mudança foi em relação à ergonomia, pois, além de ser funcional, o controle deve ser de fácil utilização por uma criança de 8 a 12 anos.

Para mudanças futuras, é esperado que tenha uma mudança no formato externo e uma estetização relacionada ao tema do jogo — personagem que o jogador controla ser um praticante de kung fu — e adicionar sistemas de segurança para que o controle não seja derrubado ou quebrado.

¹Repositórios do projeto: <https://github.com/orgs/Geometrianinja/repositories>

5 REQUISITOS E ARQUITETURA

5.1 Requisitos Funcionais

- O jogo deve apresentar conteúdos relacionados à geometria de forma interativa e lúdica;
- Um tutorial educativo deve ser incluído para ensinar o respectivo conteúdo;
- O nível de dificuldade das fases deve ser adequado ao público-alvo infantil;
- O sistema deve ser responsivo, com baixa latência entre o movimento do controle e a resposta na tela;
- O controle deve ser capaz de movimentar o cursor dentro do jogo de maneira precisa;
- O controle deve possuir botões que permitam a interação do jogador com o jogo;
- O sistema deve permitir reiniciar fases ou retornar ao menu principal a qualquer momento;

5.2 Requisitos Não Funcionais

- A narrativa do jogo deve incluir uma pequena história, com o objetivo de criar um vínculo emocional com a criança e tornar o aprendizado mais envolvente;
- O controle deve conter uma corda de segurança para evitar quedas durante o uso;
- O design do controle deve ser anatômico, garantindo conforto no manuseio;
- A interface do jogo deve ser simples, intuitiva e de fácil navegação;
- LEDs devem ser incorporados ao controle para sinalizar estados (como conexão ativa, erro, etc.);
- Os textos e instruções do jogo devem usar linguagem apropriada para o público infantil.
- O jogo deve fornecer feedback sonoro a cada ação correta

5.3 Arquitetura preliminar

Integração entre módulos de hardware (ESP32, giroscópio), software (motor de jogo) e design do controle com impressão 3D.

5.3.1 Software de Baixo Nível e Hardware. O ESP32 atua como o núcleo do sistema, sendo responsável pelo processamento dos dados recebidos do giroscópio MPU6050, dos botões de pressão (push buttons), além do controle de saída para o motor de vibração e da comunicação via Bluetooth, tanto para envio quanto recebimento de dados.

O MPU6050 é um giroscópio com acelerômetro embutido, utilizado para capturar a orientação do controle em termos de roll, pitch e yaw, ou seja, os ângulos de rotação nos três eixos espaciais. Esses dados são fundamentais para determinar a direção e o movimento realizados pelo jogador durante a jogabilidade.

Como forma de retorno tátil, é utilizado um motor de vibração do tipo Vibracall, que é ativado para fornecer um feedback sensorial ao usuário, reforçando os estímulos do jogo por meio de vibrações. Para que o ESP32 consiga controlar esse motor de maneira eficiente, é empregado um transistor BC-558, que atua como um interruptor amplificador, permitindo que uma corrente maior seja fornecida ao motor a partir de um sinal de controle de baixa corrente vindo do microcontrolador.

Para proteger o circuito e garantir que a corrente enviada à base do transistor seja adequada, é utilizado um resistor de 1kΩ. Este resistor tem como função limitar a corrente de base, prevenindo danos ao transistor e ao ESP32, além de garantir o funcionamento correto da chave de controle.

Esse fluxo de funções pode ser observado na Figura 2.

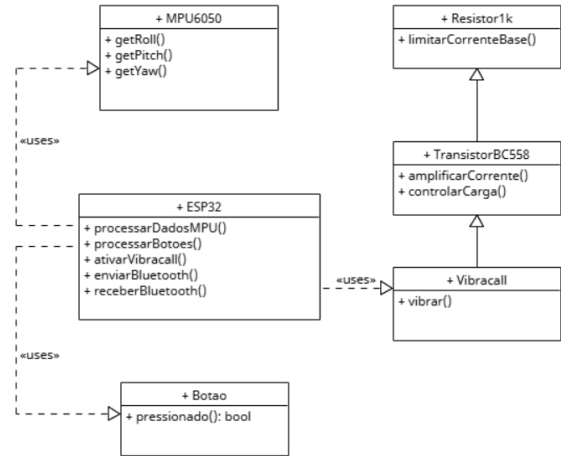


Figure 2: Diagrama de classes das funções. Elaborado pelos autores.

O circuito eletrônico montado sobre a placa de ensaio pode ser visualizado na Figura 3.

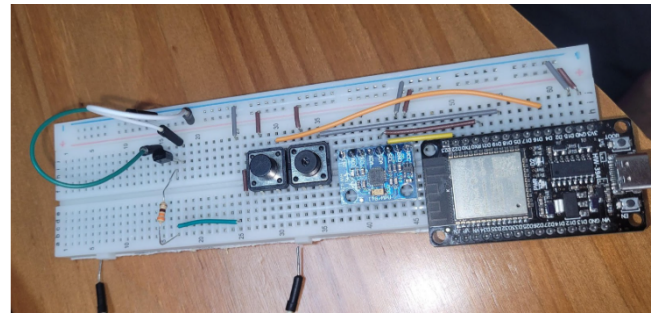


Figure 3: Circuito Eletrônico sobre placa de ensaio. Elaborado pelos autores.

A arquitetura modular adotada no firmware para o ESP32 segue princípios fundamentais de sistemas embarcados em tempo real, priorizando a responsividade, o isolamento funcional e a eficiência na comunicação assíncrona. Até o momento, o projeto se estende em três módulos - main.cpp, button.cpp e mpu.cpp - que administram funções específicas. Tal prática promove a coesão e divisão de responsabilidades, características intrínsecas em desenvolvimentos direcionados e sistemas embarcados[6].

```

1 funcao setup()
2     configurar GPIOs
3     iniciar comunicacao Bluetooth (SerialBT)
  
```

```

4      chamar add_button() para configurar botoes
5      chamar add_mpu() para iniciar sensor MPU6050
6      criar tarefa bt_listener()
7      criar tarefa wait_for_question_mark()
8
9  funcao loop()
10     alternar LED no GPIO 2 para indicar
        funcionamento
11     delay(tempo curto)
12
13 tarefa bt_listener()
14     enquanto true
15         se receber dado via Bluetooth
16             se dado == 2
17                 atualizar timestamp da ultima
                    comunicacao
18                 ativar motor de vibracao
19             senao
20                 processar outro comando recebido
21             se tempo desde ultimo byte 2 > 2 segundos
22                 reiniciar processo de conexao (voltar
                    a aguardar "?")
23
24 funcao mpu_callback(yaw, pitch, roll)
25     enviar dados via SerialBT
26
27 funcao wait_for_question_mark()
28     enquanto caractere recebido != "?"
29         aguardar
30     iniciar comunicacao definitiva

```

Listing 1: Pseudocódigo de inicializacao e tarefas.

No arquivo main.cpp, a função setup() executa a configuração dos periféricos e cria tarefas FreeRTOS responsáveis por processar eventos paralelos. Essa abordagem, comumente utilizada em aplicações microprocessadas, particiona o comportamento do sistema em tarefas concomitantes, diminuindo a latência da aplicação.

A tarefa bt_listener, executada sobre FreeRTOS, se trata de uma função reativa baseada em eventos. Ela monitora continuamente os dados recebidos via Bluetooth e atua sobre o motor de vibração caso seja constatado o comando de ativação. Além disso, ela vigia o estado da comunicação com uma política de timeout, prevenindo travamentos em caso de perda de comunicação. Esse tipo de controle temporal é essencial em sistemas embarcados, conforme abordado por [4], onde o tempo não é apenas uma variável, mas uma dimensão funcional do sistema.

```

1 funcao add_button(pino, rise_callback,
        fall_callback)
2     configurar GPIO como entrada com interrupcao
3     associar bt_isr_handler ao pino
4
5 funcao bt_isr_handler()
6     desabilitar interrupcao no pino
7     criar tarefa bt_dbounce(pino, rise_callback,
        fall_callback)
8
9 tarefa bt_dbounce(pino, rise_callback,
        fall_callback)
10     esperar tempo de debounce
11     ler estado do botao

```

```

12     se estado for pressionado
13         chamar fall_callback()
14     senao
15         chamar rise_callback()
16     reabilitar interrupcao no pino

```

Listing 2: Pseudocódigo para configuracao e tratamento de botoes com debounce.

O módulo button.cpp implementa lógica de tratamento de botões físicos com foco em resposta determinística e tratamento de ruído. Utiliza-se interrupções para capturar eventos com latência mínima (bt_isr_handler) e uma tarefa separada para aplicar debounce via software (bt_dbounce). A estrutura de callback implementada permite que o comportamento associado a eventos de pressionamento e soltura seja configurado dinamicamente, promovendo reutilização e flexibilidade.

```

1 funcao add_mpu(data_ready_callback)
2     iniciar sensor MPU6050 com biblioteca DMP
3     configurar offsets do giroscopio
4     criar tarefa mpuTask(data_ready_callback)
5
6 tarefa mpuTask(callback)
7     enquanto true
8         se novos dados disponiveis no FIFO do
            MPU6050
9             ler dados brutos
10            calcular yaw, pitch e roll
11            chamar callback(yaw, pitch, roll)
12            delay(tempo de amostragem)

```

Listing 3: Pseudocódigo para inicializacao e processamento de dados do sensor MPU6050.

No mpu.cpp, o sensor MPU6050 é inicializado e seus dados são tratados através da biblioteca MPU6050_6Axis_MotionApps612, que utiliza o DMP (Digital Motion Processor) interno para realizar o pré-processamento dos dados de movimento. A tarefa mpuTask é responsável pela leitura dos dados e cálculo dos valores de roll, pitch e yaw, enviando-os ao sistema via callback. A calibração do dispositivo é realizada manualmente a fim de evitar variações nos dados entre as reinicializações.

5.3.2 Software de alto nível. O software de alto nível é o que controla o jogo em questão, ele está sendo desenvolvido em linguagem python, e conta com o auxílio dos recursos disponibilizados através da biblioteca pygames. Com o intuito de otimizar a programação, o conceito de modularização está sendo aplicado, que consiste em separar em diferentes arquivos cada funcionalidade do jogo, de forma que a equipe possa atuar de maneira interdependente no desenvolvimento. Abaixo está uma descrição dos módulos desenvolvidos até o momento, para cada módulo em questão é apresentado um diagrama de classes que corresponde a arquitetura de código desenvolvida, bem como suas respectivas funções.

A figura 4 apresenta o diagrama das classes responsáveis por gerar as diferentes formas geométricas que surgem na tela durante o decorrer do jogo. Com um módulo dedicado, a classe Forma possui os atributos necessários para gerar qualquer tipo de figura geométrica na tela. Partindo dessa configuração, as subclasses herdam tais atributos, e contém uma lista de pontos, distribuídos de tal

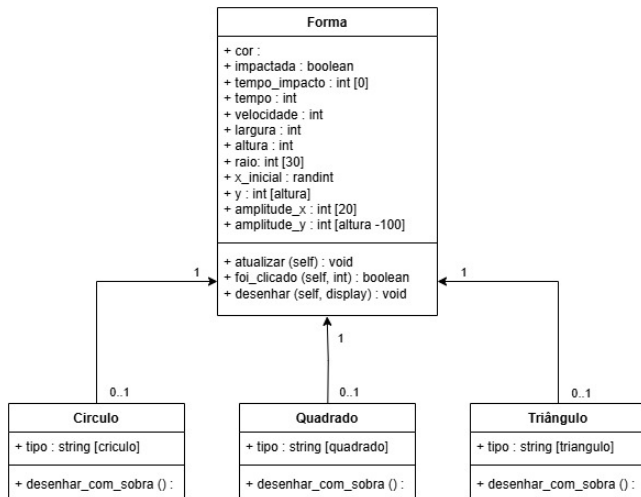


Figure 4: Módulo Formas

a forma a gerar na tela a figura geométrica de nome correspondete da subclasse.

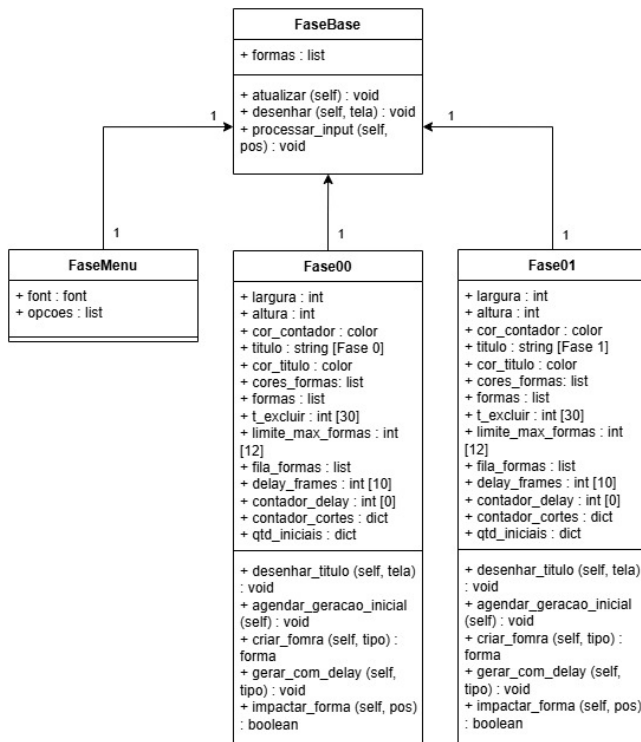


Figure 5: Módulo Fases

A figura 5 apresenta uma diagrama das classes que gerenciam as fases do jogo. Nesse sentido, seguindo as boas práticas de programação para desenvolvimento de jogos e softwares de alto nível, aqui também foi utilizado o conceito de herança, no qual temos uma fase modular inicial, denominada como FaseBase. Na sequência,

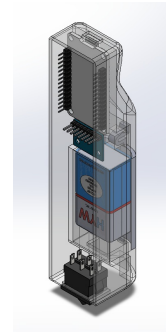


Figure 6: Protipação em CAD do esquemático externo do controle.

uma herança dessa classe é a FaseMenu, que tem como função no código apresentar a lista de fases disponíveis para serem acessadas. Assim também, as outras duas classes utilizam dessa herança são as Fase00 e Fase01, das quais cada uma possui o seu conjunto de atributos e métodos que garante a jogabilidade conforme descrito na seção de metodologia.

A modelagem do controle foi feita de forma que tenha espaço para todos os componentes eletrônicos integrados em uma placa de circuito impresso e, concomitantemente, fosse ergonômica, a carcaça do controle pode ser vista na figura 6.

6 PRODUTO MÍNIMO VIÁVEL (MVP)

6.1 Primeira Fase do Projeto

Nesta primeira fase do projeto, o objetivo é desenvolver uma versão inicial funcional do sistema, focando na estrutura básica de software, hardware e na modelagem dos componentes. A seguir, estão descritos os itens contemplados nesta etapa:

6.1.1 Software (Alto Nível).

- Menu Inicial: Desenvolvimento de uma interface de entrada que permita ao usuário iniciar o jogo, acessar opções e sair da aplicação.
- Menu de Fases: Criação de uma tela para seleção das fases disponíveis no jogo.
- Integração entre Menus: Implementação da navegação entre o menu inicial e o menu de fases, garantindo fluidez e usabilidade.
- Protótipo da Fase 0: Construção do protótipo de uma fase introdutória com funcionalidades básicas para testes iniciais da mecânica do jogo.

6.1.2 Software (Baixo Nível).

- Controle do Ponteiro na Tela: Implementação da movimentação de um ponteiro virtual com base em dados provenientes de um giroscópio.
- Código para o ESP32: Desenvolvimento do firmware responsável pela leitura do sensor e envio de comandos ao jogo.
- Comunicação via Bluetooth: Estabelecimento de uma conexão Bluetooth entre o controle (com ESP32) e o sistema de jogo, permitindo a troca de dados *wireless*.

6.1.3 3. Hardware.

- Integração dos Componentes: Montagem de um protótipo funcional utilizando protoboard, visando validar a comunicação entre os componentes eletrônicos e o microcontrolador.

6.1.4 Modelagem.

- Modelagem dos Componentes Eletrônicos: Representação dos espaços ocupados pelos circuitos utilizados na montagem do controle.
- Modelagem Física do Controle: Criação de um protótipo do controle com base em parâmetros ergonômicos.
- Ergonomia: Considerações de design inspiradas no controle do Nintendo Wii, com foco em conforto, manuseabilidade e adequação ao público-alvo.

6.2 Segunda Fase do Projeto

A segunda fase do projeto tem como foco a integração dos módulos desenvolvidos na primeira etapa, bem como a revisão e aprimoramento do protótipo físico. O objetivo é consolidar a estrutura do jogo e do controle, promovendo melhorias tanto no funcionamento quanto na usabilidade do sistema.

6.2.1 Software (Alto Nível).

- Tutorial: Desenvolvimento de um tutorial introdutório que apresenta o universo do jogo, suas mecânicas e os objetivos propostos ao jogador.
- Integração com Controle: Adaptação do código do jogo para permitir o controle total do cursor por meio do dispositivo desenvolvido, garantindo a funcionalidade esperada durante as fases.

6.2.2 Software (Baixo Nível).

- Otimização da Comunicação com o Computador: Revisão e aprimoramento do protocolo de comunicação Bluetooth, visando maior estabilidade e menor latência na transmissão dos dados.
- Melhoria na Aquisição dos Dados: Refinamento da lógica de coleta de dados pelos sensores, de modo a garantir precisão e responsividade durante a jogabilidade.

6.2.3 Hardware.

- Modelagem da Placa de Circuito (Fenolite): Elaboração do layout da placa de circuito impresso (PCI) em software de CAD, com posicionamento adequado dos componentes eletrônicos, visando a futura montagem final do controle.

6.2.4 Modelagem.

- Revisão do Design do Controle: Ajustes na modelagem física do controle, considerando o encaixe adequado dos componentes eletrônicos, como sensores, ESP32 e alimentação.
- Adequação Estética ao Tema do Jogo: Modificação visual do controle para refletir a identidade visual do jogo, promovendo maior imersão e coerência temática.

6.3 Terceira Fase do Projeto

A terceira fase do projeto contempla a finalização do jogo com uma versão funcional e jogável, acompanhada de um controle

físico pronto para uso. O objetivo principal é garantir que qualquer usuário consiga interagir com o jogo de forma intuitiva e fluida, permitindo sua apresentação como um produto funcional.

6.3.1 Software (Alto Nível).

- Tutorial da Fase 01: Desenvolvimento de um tutorial explicativo específico para a Fase 01, incluindo a introdução narrativa correspondente a esse estágio do jogo.
- Implementação da Fase 01: Finalização de todos os elementos que compõem a Fase 01, garantindo jogabilidade estável e completa, sem necessidade de grandes ajustes posteriores.

6.3.2 Software (Baixo Nível).

- Sincronização: Ajuste da taxa de comunicação via Bluetooth para que esteja alinhada com a taxa de atualização do jogo, assegurando resposta em tempo real e boa experiência do usuário.

6.3.3 Hardware.

- Placa de Circuito: Conclusão do projeto da placa de fenolite e envio para fabricação, visando a montagem final do controle físico com circuito dedicado.

6.3.4 Modelagem.

- Impressão 3D: Produção do corpo do controle utilizando impressão 3D, considerando o design final aprovado e a acomodação dos componentes eletrônicos.

REFERENCES

- [1] Mortimer J. Adler. 1982. *The Paideia Proposal: An Educational Manifesto*. Macmillan Publishing Company, New York. <https://files.eric.ed.gov/fulltext/ED455213.pdf> Acesso em: 1 maio. 2025.
- [2] Brasil, Ministério da Educação. 2018. Base Nacional Comum Curricular. https://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518-versaofinal_site.pdf. Acesso em: 1 maio 2025.
- [3] Robert G. Cooper. 2016. Agile-Stage-Gate Hybrids: The Next Stage for Product Development. *Research-Technology Management* (2016).
- [4] Rômulo Silva de Oliveira. 2013. *Fundamentos de Sistemas de Tempo Real* (2 ed.). LTC, Rio de Janeiro.
- [5] Educação Pública. 2023. *Uso de jogos didáticos no processo ensino-aprendizagem*. <https://educacaopublica.cecierj.edu.br/artigos/19/23/uso-de-jogos-didaticos-no-processo-ensino-aprendizagem> Acesso em: 23 abr. 2025.
- [6] José Henrique Kleinschmidt. 2016. *Sistemas Embarcados: Fundamentos e Aplicações* (1 ed.). LTC, Rio de Janeiro.
- [7] Manga High. 2022. *Os benefícios da aprendizagem baseada em jogos*. <https://www.mangahigh.com/en-us/blog/2022-03-25-os-benef%C3%ADcios-da-aprendizagem-baseada-em-jogos> Acesso em: 23 abr. 2025.
- [8] A. S. Marques. [n. d.]. Desenvolvimento de ferramenta para remoção da espécie invasora coral-sol. p.32–36, 2025.
- [9] PEPSIC. 2019. Jogos e brincadeiras na escola: contribuições para o desenvolvimento infantil. *Psicologia Escolar e Educacional* 23, 1 (2019). https://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1415-69542019000100005 Acesso em: 23 abr. 2025.
- [10] Odalea Aparecida Viana. 2005. *O componente espacial da habilidade matemática de alunos do ensino médio e as relações com o desempenho escolar e as atitudes em relação à matemática e à geometria*. Ph.D. Dissertation. Universidade Estadual de Campinas, Faculdade de Educação, Campinas, SP. https://hdl.handle.net/20.500.12733/1601193_279p. Acesso em: 1 mai. 2025.
- [11] ÁRVORE. 2022. *Jogos na educação: veja como utilizar o recurso em sala de aula*. <https://www.arvore.com.br/blog/jogos-na-educacao> Acesso em: 23 abr. 2025.