BACK-END DE JOGOS EDUCACIONAIS

Maurício Scarelli Arantes Orientador: Prof. Dr. Wilson Massashiro Yonezawa

Sumário

- Introdução
- Fundamentação teórica
- Ferramentas
- Desenvolvimento
- Conclusão

Introdução

Introdução

- Ascensão da tecnologia educacional
- Desenvolvimento através do lúdico
- Videogames no aprendizado
- Impacto da COVID na educação

Justificativa

- Demanda por mais ferramentas de ensino
- Dificuldade em criar jogos multijogador em rede
- Poucas ferramentas de auxilio no desenvolvimento

Objetivos

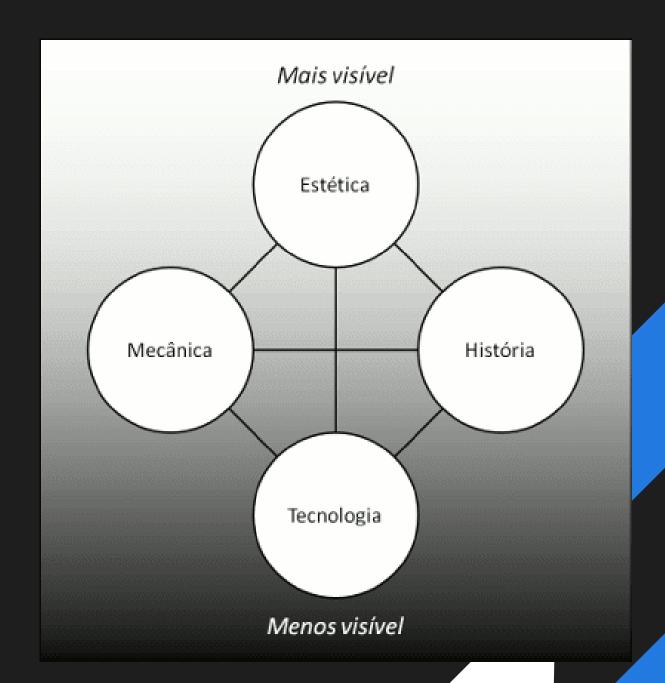
Desenvolver uma plataforma:

- Comporte diferentes jogos
- Simplifique a implementação de modos em rede
- Armazene os dados para análise do professor

Fundamentação Teórica

Jogos

- Definição de um jogo:
 - Visão acadêmica
 - Visão prática
- Características principais:
 - Mecânica
 - Narrativa
 - Estética
 - Tecnologia
- Diferença em relação a outras mídias



Jogos digitais

- Diferenças em relação a jogos não digitais
- Tecnologia como forma de interação
- Tecnologias úteis para o game designer:
 - Interatividade imediata
 - Manipulação de informação
 - Sistemas complexos automatizados
 - Comunicação via redes



Jogos digitais na educação

- Jogos e a aprendizagem
- Semelhanças com o processo de ensino





API

- Conjunto de rotinas e padrões
- Construção e integração de aplicações
- Abstração
- Código reutilizável

De onde surgiram as APIs?

- Bibliotecas de sub-rotinas
- Criação do EDSAC
- Primeiros programas triviais
- Tempo elevado na depuração de código
- Arquitetura de sub-rotinas



The library of tapes on which subroutines are punched is contained in the steel cabinet shown on the left. The operator is punching a program tape on keyboard perforator. By placing library tapes on the tapereader shown in the center of the photograph, the operator can copy them mechanically onto the tape she is preparing.

De onde surgiram as APIs?

- Primeira biblioteca de código reutilizável
- Ideias principais na criação de bibliotecas
- Surgimento do termo Application Program Interface
- Expansão para outras áreas da computação

PART TWO

SPECIFICATIONS OF EDSAC LIBRARY SUBROUTINES

Each subroutine is distinguished by a letter denoting its category, and a serial number within that category. The categories are as follows.

Category	Subject
A	Floating-point arithmetic
В	Arithmetical operations on complex numbers
C	Error diagnosis
D	Division
E	Exponentials
F	General subroutines relating to functions
G	Differential equations
L	Logarithms
M	Miscellaneous
N	Double-length arithmetic
P	Print and layout
Q	Quadrature
R	Read (i.e., Input)
S	nth root
T	Trigonometric functions
X	Complete programs
Z	Post-mortem routines

In the specifications on succeeding pages the following information is given in abbreviated form immediately beneath the title of each subroutine:

- 1. Type of subroutine, i.e., whether open, closed A, closed B, interpretive, or special.
- 2. Restriction on address of first order. If the word "even" appears it denotes that the first order must be placed in an even location. If no note appears it indicates that the location may be either odd or even.
- 3. Total number of storage locations occupied by the subroutine.
- 4. Addresses of any storage locations needed as temporary storage by the subroutine (other than 0D, which is used by the majority of subroutines).
- 5. Approximate operating time (not possible to state in all cases).

139

De onde surgiram as APIs?

- Criação do conceito de web APIs
- Web APIs no meio social
- Infraestrutura e computação em nuvem



Classificações de APIs

- Mecanismo de transporte
- Tipo de acesso
- Finalidade
- Arquitetura

APIs em jogos

- Presentes do baixo ao mais alto nível
- Softwares de desenvolvimento de jogos
- Exposição de dados de jogos ou partidas







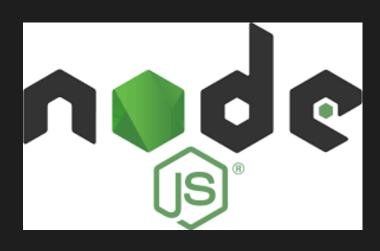
Representação digital de jogadas

- Jogadores interagem através de ações
- Forma de recriar e representar estado atual
- Estado completo:
 - Notação Forsyth-Edwards (FEN):
 - "1n2kb1r/2rb3p/3RPQ1p/1p3N2/8/8/1P3PPP/2q2RK1 w"
- Registro da jogada:
 - Notação Algébrica Padrão (SAN): "Qc1"



Ferramentas









Typescript

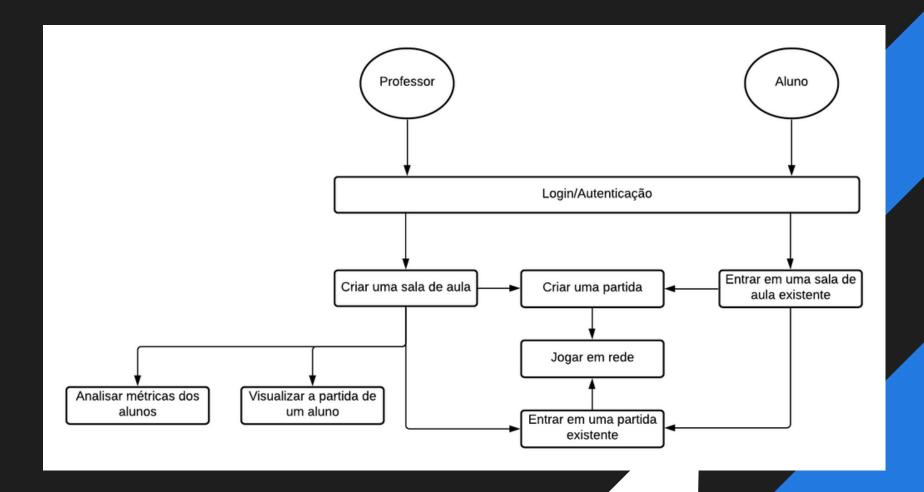


Socket.io

Desenvolvimento

Desenvolvimento

- Modelagem da aplicação
- Módulos de comunicação REST e event based
- Criação das funcionalidades



Conclusão

Conclusão e Trabalhos futuros

Conclusão

- Permitiu a criação de uma partida multijogador
- Armazena os dados das partidas
- Sistema funciona de maneira satisfatória, atingindo seus objetivos

Próximos passos

- Adição de cache
- Implementar geração de métricas
- Desenvolver um jogo modelo

Obrigado!