

Uma abordagem fazendo uso de *serious games* para ensino de Sistemas Operacionais

Matheus dos Santos Luccas
 Escola de Engenharia de São Carlos
 - EESC - USP
 Instituto de Ciências Matemáticas e da Computação
 - ICMC - USP
 São Carlos, Brasil
 matsluccas@gmail.com

Julio Cezar Estrella
 Instituto de Ciências Matemáticas e da Computação
 - ICMC - USP
 São Carlos, Brasil
 jcezar@icmc.usp.br

Leonardo Tortoro Pereira
 Instituto de Ciências Matemáticas e da Computação
 - ICMC - USP
 São Carlos, Brasil
 leonardo.t.pereira13@gmail.com

Kalinka Regina Lucas Jaquie Castelo Branco
 Instituto de Ciências Matemáticas e da Computação
 - ICMC - USP
 São Carlos, Brasil
 kalinka@icmc.usp.br

Resumo—Este artigo apresenta o desenvolvimento e a aplicação de três jogos, de propósito educacional, projetados para servirem de ferramenta de auxílio ao ensino da disciplina de Sistemas Operacionais, que é ministrada em grande parte dos cursos de computação de nível superior. A forma adotada para que esses jogos fossem aplicados ao ensino dessa disciplina se justifica nos princípios do construtivismo. As mecânicas, jogabilidades e ambientações dos três jogos bem como os seus motivos de existência, justificados pelo propósito educacional, são apresentados com um esclarecimento e uma exploração dos tópicos da disciplina que regem a forma dos jogos. Um teste e estudo com alunos da Universidade de São Paulo, *campus* de São Carlos, envolvendo a aplicação desses três jogos foi realizado, e os resultados obtidos permitem concluir que o uso de jogos sinaliza para a melhora do processo de ensino e aprendizagem.

Palavras-chave: Jogos para Sistemas Operacionais, Escalonamento de Processos, *Race Conditions*, *Threads*, *Serious Games*.

I. INTRODUÇÃO

No âmbito educacional, inovar é visto como uma necessidade, pois as tecnologias que nos rodeiam estão constantemente em transformação. A informação muda sua estrutura e forma, mudando a maneira como são entendidas, interpretadas e compreendidas. Com isso, a simples mudança de turmas pode se tornar um desafio de adaptação, imediatamente relacionado à velocidade com a qual a tecnologia se transforma.

É sobre esse ideal, da necessidade constante de adaptação, que esse artigo se estabeleceu, explorando novas possibilidades da aplicação de *serious game* no âmbito do ensino superior em computação.

Serious game define-se como uma categoria de jogos eletrônicos com a específica ganância de consolidar a mestria da indústria de jogos e a possibilidade, por meio de simuladores, de servirem como ferramentas para o ensino

e habilitação [1]. Essa enriquecida ferramenta propõe um ambiente de caráter simulado e interativo, onde a construção do conhecimento é permitida por meio de aspectos lúdicos e multi motivadores dos jogos de entretenimento [2], muito populares nos tempos de hoje, representando em 2020 um mercado que move quase \$ 160 bilhões de dólares mundialmente^{1 2}.

Três jogos, *RaceConditions*, *Threadman* e *Escalonando*, foram desenvolvidos com essa perspectiva, aplicando uma particularidade que se fez regra a ser respeitada: a de que a estrutura, mecânica e jogabilidade carregassem inteiramente a mensagem e o ensino que se anseia aplicar, ou seja, os assuntos e tópicos a serem destacados estariam, em princípio, apresentados unicamente ao se jogar uma partida dos jogos, procurando não recorrer a informações escritas ou *quizes* que venham porventura mascarar atividades entediantes, prática esta comum e observada em diversas tentativas de aplicar o conceito de *serious game*.

Para avaliar os jogos foi realizada pesquisa com alunos que cursaram a disciplina na Universidade de São Paulo, *campus* de São Carlos, no ano de 2019, viabilizando uma análise qualitativa de opinião e eficiência, além de um levantamento de dados para possíveis novos projetos.

II. O ENSINO TEÓRICO DE SISTEMAS OPERACIONAIS EM NÍVEL SUPERIOR

A disciplina de Sistemas Operacionais I é oferecida pela maior parte dos cursos de graduação em Computação, representando conteúdo de relevante importância para a

¹Cabe destacar que todos os segmentos de jogos apresentaram um aumento no engajamento e nas receitas como resultado das medidas de isolamento social impostas pela COVID-19 em 2020.

²<https://newzoo.com/insights/articles/newzoo-games-market-numbers-revenues-and-audience-2020-2023>

formação dos alunos da área de Computação, sendo indicada e aconselhada pelo currículo de referência da SBC (Sociedade Brasileira de Computação) [3], da IEEE (*The Institute of Electrical and Electronics Engineers*)³ e da ACM (*Association for Computing Machinery*)⁴.

Em nível teórico, são apresentados na disciplina diversos conceitos (como por exemplo escalonamento de processos, *threads*, *deadlock*, região crítica, entre outros) de categoria abstrata, dos quais o aluno pode vir a ter maiores dificuldades em compreender. Para esses conceitos, uma ferramenta de simulação, exemplificação e visualização se apresenta como uma grande aliada, ao ambientar tais conteúdos dentro de um contexto de menor teor abstrato e mais literal [4].

Nos jogos desenvolvidos para a disciplina de Sistemas Operacionais, apresentados neste artigo, foram abordados dois cenários com foco no contexto educacional e de ensino:

- apresentar os jogos seguidos do material didático formal, oferecendo aos alunos um contexto anterior ao técnico e teórico, por exemplo, antes das aulas e;
- apresentar os jogos após o material didático formal, ou seja, após a aula, como forma de reforço, revisão e fixação do conteúdo. Consequentemente, oferecendo aos alunos um contexto posterior ao técnico e teórico.

Em ambos os cenários, monitora-se os princípios do construtivismo.

III. MÉTODOS E ESTRATÉGIAS NO DESENVOLVIMENTO

Para o desenvolvimento dos jogos, adotou-se o motor gráfico *Unity*, disponível gratuitamente para *Windows*, *Linux* e *Mac* e, de alta popularidade tanto para nível profissional em desenvolvimento de jogos, quanto para desenvolvedores iniciantes⁵.

A música e os efeitos sonoros foram desenvolvidos com a ferramenta *LMMS (Linux MultiMedia Studio)*, disponível gratuitamente para *Linux*, *Windows* e *Mac*. Trata-se de uma ferramenta intuitiva, eficiente e amigável para o seu propósito⁶.

As imagens, *sprites* e animações usadas nos jogos foram desenvolvidas por meio da ferramenta *GIMP (GNU Image Manipulation Program)*, uma ferramenta que permite a manipulação e criação de imagens de maneira simplificada e também disponíveis para os diferentes sistemas operacionais⁷.

IV. Gameplay E SUA ABSTRAÇÃO PARA COM A DISCIPLINA

Os jogos possuem versões compatíveis para os Sistemas Operacionais *Windows*, *Mac* e *Linux*, também contando com uma versão para dispositivos *Android*. Atualmente os

jogos se encontram disponíveis em versão para testes com amplo acesso⁸.

A. Escalonando

O jogo **Escalonando** se trata de um jogo de *puzzle* e de atenção, onde o jogador se coloca no papel de um escalonador (como o escalonador de processos de um Sistema Operacional), responsável por organizar pilhas de tarefas a serem processadas antes que sejam demasiadamente acumuladas.

Para evitar o acúmulo, o jogador terá que avaliar quais *processos* (no caso do jogo quais barras) são prioritários, quais estratégias usar e, aplicar o conceito de *preempção* (outro conceito importante dentro da disciplina).

No início da partida, o jogador se deparará com uma série de setores retangulares distribuídos lateralmente, conforme ilustrado na Fig. 1. Nesses setores, encontram-se pilhas de retângulos coloridos que representam, de forma abstrata, pilhas de processos dos quais o jogador, em papel de escalonador, deverá escalonar para que sejam reduzidas. O tamanho de cada processo varia e, o tempo total para completá-los depende desse tamanho e da sua velocidade particular de execução.

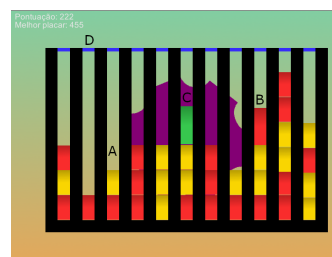


Fig. 1. Exemplo de cena do jogo: A - processo em estado **pronto**, B - processo em estado **bloqueado**, C - processo em estado **executando**, D - faixa de limite, se a pilha de processos a alcançar, o jogador poderá perder o jogo

As cores de cada processo indicam o seu estado: a cor vermelha indica um processo **bloqueado**, amarelo indica que está **pronto** para execução e verde indica um processo **executando**. Um processo indefinido é representado por sua ausência do campo de visão do jogador.

O jogador, portanto, deverá aplicar estratégias considerando as prioridades, levando em conta o tamanho e o tempo de execução de cada tarefa, além de considerar a *preempção* e o bloqueio de um processo em execução para gerenciar todos os processos e alcançar placares maiores de pontuação.

O jogo ainda permite que o jogador aplique em sua estratégia, por conta própria, alguns dos algoritmos de escalonamentos tratados na disciplina, tais como o *first-come, first served* ou o *shortest job first*.

B. Race Conditions

O jogo **Race Conditions** desafia o jogador com uma mecânica que exige ritmo e atenção constante. Foi

⁸<https://drive.google.com/drive/folders/1mpG6w-OYjmJKiyWMm-n2rjPOqJzhKA-g?usp=sharing>

³<https://www.computer.org/volunteering/boards-and-committees/professional-educational-activities/curricula>

⁴<https://www.acm.org/education/curricula-recommendations>

⁵available:<https://unity.com>

⁶<https://lmms.io>

⁷<https://www.gimp.org>

desenvolvido com o propósito de tratar de assuntos como a comunicação entre processos, regiões críticas, condições de corrida, exclusão mútua, espera ocupada, variáveis de travamento, estrita alternância e primitivas *sleep/wakeup*, também permitindo a discussão quanto ao funcionamento de monitores.

Para não encerrar a partida, o jogador deverá garantir que apenas um processo (representado pelos quadrados, como ilustrado na Fig. 2) ocupe a região crítica (representada pelo círculo), assim realizando o papel de comunicador.

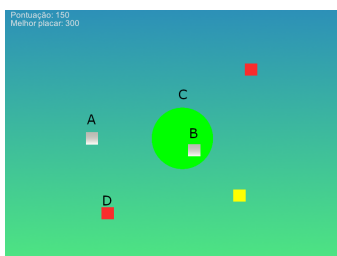


Fig. 2. Exemplo de cena do jogo: A - um processo se encaminhando em direção à região crítica, B - um processo dentro da região crítica, se algum outro processo acessar essa região antes que esse processo saia, a partida será encerrada, C - a região crítica, D - um processo que está esperando há muito tempo para acessar a região crítica, o jogador deverá tomar cuidado de priorizar o acesso.

O jogador também deverá garantir que nenhum processo fique em espera indefinidamente aguardando acesso à região crítica.

Para gerenciar a ocupação da região crítica, o jogador poderá escolher, antes do início da partida, uma dentre as três estratégias seguintes modos: variáveis *lock*, *sleep/wakeup* e *strict alternation*.

Alternando entre os modos, o jogador pode simular as três estratégias ensinadas em Sistemas Operacionais e observar suas peculiaridades, vantagens e desvantagens. Além disso, exercitar os diversos tópicos relacionados a comunicação de processos.

C. Threadman

O jogo *Threadman*, em sua jogabilidade, remete às plataformas mais clássicas. Aqui, o jogador controla a rota de uma *thread*, por meio das teclas *wasd*, ou do controle de toque do celular. Tal *thread* dividirá espaço com outras, todas colaborando para utilizar os recursos escassos e não violar a região crítica.

Neste jogo, o conteúdo instrucional incorporado foi o funcionamento das *threads* em processamento, os conceitos de semáforos, de região crítica e de gerenciamento de memória com partições fixas. Além disso, possibilita ao jogador adotar estratégias de gerenciamento de memória estudadas na disciplina de Sistemas Operacionais, tais como *first fit*, *best fit* e *worst fit*, simulando-as e observando suas características, vantagens e desvantagens.

Embora as *threads*, dentro do cenário do jogo, estejam programadas para não quebrar essas regras, o jogador deverá ter atenção para que a *thread* que está em seu controle não



Fig. 3. Exemplo de cena do jogo: A - *thread* controlada pelo jogador, B - uma das regiões críticas, C - ponto de dados que o jogador deve recolher, D - espaço de memória onde o jogador deve depositar os pontos, E - Exemplo de *threads* que compartilharão espaço com o jogador, F - ponto de controle dos espaços de memória

as viole. Portanto, ela deve apenas acessar a região crítica se esta já não se encontrar lotada. Caso alguma região crítica seja violada, a partida se encerra.

O objetivo do jogador é coletar um bloco de informação (iniciado de forma randômica no mapa do jogo), atravessando o mapa com regiões críticas em seu caminho para alcançá-lo, retornar ao ponto de início do jogo e depositar esse bloco em uma das partições de memória fixa.

V. RESULTADOS OBTIDOS

Para validar a efetividade dos jogos no aprendizado, realizou-se uma comparação entre duas turmas da disciplina Sistemas Operacionais I da Universidade de São Paulo, com pré-requisitos equivalentes e ministradas pelo mesmo docente. A primeira turma, denominada Turma A, contou com a participação de 38 alunos, foi ministrada no segundo semestre de 2018 e não foram apresentados os jogos nas aulas. A segunda turma, denominada Turma B, contou com a participação 39 alunos, foi conduzida no segundo semestre de 2019, sendo que os jogos foram utilizados como atividade complementar.

Considerando a ementa da disciplina, ao término da mesma o aluno deve ter conhecimentos iniciais sobre sistemas operacionais (Histórico), *threads*, *deadlock*, regiões críticas, condição de disputa, escalonamento de processos, gerenciamento de memória, paginação, *cache*, sistemas de arquivos, princípios de entrada e saída, entre outros. Na Tabela I são ilustrados os níveis de profundidade (Nível Profund.) considerados para todos os tópicos abordados durante o curso e como cada tópico foi avaliado. A coluna peso contém o fator de ajuste utilizado posteriormente para o cálculo dos níveis de conhecimento (Nível Conhec.), sendo que para o nível 1, o peso deve ser 0 pois o assunto não foi avaliado. A finalidade deste peso é reajustar o efeito de uma pergunta simples sobre um determinado tópico que poderia ser explorado de forma mais ampla [5].

Para a avaliação foram estipulados quatro níveis de habilidade adquirida, denominados níveis de conhecimento, sobre cada tópico abordado. Para cada estudante individualmente, por meio das avaliações, em uma escala de 0 a 10, foi atribuído um nível de conhecimento, considerando o fator de ajuste adequado. Na tabela, a notação ”| — ” indica que o valor à esquerda está incluído no intervalo considerado,

Tabela I
NÍVEIS DE PROFUNDIDADE DOS TÓPICOS.

Nível	Nome	Descrição	Peso
1	Superficial	O assunto foi introduzido e indicado ao aluno, de modo superficial, sem que seu conteúdo fosse cobrado.	0
2	Teórico	O assunto foi apresentado ao aluno teoricamente, durante o curso ou em disciplina pré-requisito. Conceitos teóricos foram avaliados sobre o assunto.	0,70
3	Prático	Foi exigido em atividade prática executada com orientação. Os conceitos teóricos e resultados da experiência foram avaliados.	0,85
4	Pleno	O assunto foi avaliado por meio de trabalho prático individual ou coletivo, bem como por meio de avaliação escrita.	1,00

enquanto o valor à direita não. Na Tabela II são ilustrados os níveis de conhecimento utilizados.

Tabela II
NÍVEIS DE CONHECIMENTO DOS TÓPICOS.

Nível	Nome	Descrição	Nota
1	Desconhecido	O aluno neste nível de conhecimento não consegue descrever a utilidade do tópico abordado.	0 – –2,5
2	Reconhecido	O aluno entende a finalidade do tópico abordado, mas não saberia como aplicar na prática.	2,5 – –5,5
3	Manutenível	O aluno não só sabe a finalidade, como é capaz de fazer pequenos e médios ajustes e adaptação para fazer uso do tópico abordado.	5,5 – –7,5
4	Crítico	O aluno não só é capaz de fazer ajustes e adaptações, como é capaz de implementar novas funcionalidades, realizar e desenvolver projetos e avaliar as soluções implementadas.	7,5 – – 10,0

Os resultados obtidos comparando o nível de profundidade e o nível de conhecimento são ilustrados na Tabela III. Para a classificação do nível de conhecimento foi considerada a média ponderada (Média Pond.) da turma baseado no nível de profundidade do tópico. Tópicos não avaliados, mas que constam da ementa do curso, aparecem com o valor "NA"(Não Avaliado). O desvio padrão (Desvio Pad.) é também apresentado como informação adicional.

A partir da observação dos dados da Tabela III é possível averiguar que o nível de profundidade dos temas abordados foi o mesmo para as turmas A e B. Com a utilização dos jogos como elemento de ensino complementar, é possível observar

Tabela III

RESULTADOS - COMPARATIVOS DA TURMA A E TURMA B EM NÍVEL DE PROFUNDIDADE E DE CONHECIMENTO DOS TÓPICOS.
NP - NÍVEL DE PROFUNDIDADE, MA - MÉDIA ARITMÉTICA, MP - MÉDIA PONDERADA, NC - NÍVEL DE CONHECIMENTO, DP - DESVIO PADRÃO, EP - ESCALONAMENTO DE PROCESSOS, D - DEADLOCK, M - MEMÓRIA, ES - ENTRADA/SAÍDA, HS - HISTÓRICO DE SISTEMAS, C - CONCEITOS, G - GERAL.

	Turma A					Turma B				
	NP	MA	MP	NC	DP	NP	MA	MP	NC	DP
EP	4	4,13	4,13	2	0,95	4	6,19	6,19	3	1,2
D	4	5,47	5,47	2	1,2	4	6,39	6,39	3	0,91
M	3	6,86	5,83	3	0,72	3	6,94	5,90	3	1,40
ES	2	4,38	3,07	2	1,19	2	4,64	3,24	2	1,15
HS	2	NA	NA	NA	NA	2	NA	NA	NA	NA
C	2	6,05	4,24	2	0,5	2	3,31	2,32	2	0,81
G	2	NA	NA	NA	NA	2	NA	NA	NA	NA

que a média da turma B foi maior que a média da turma A em todos os tópicos em que os jogos foram apresentados como atividade complementar, considerando que o mesmo nível de profundidade foi abordado em ambas as turmas.

VI. CONCLUSÃO

Neste artigo foram apresentados três jogos desenvolvidos para o ensino de Sistemas Operacionais. Os jogos foram bem recebidos pelos alunos, que se mostraram otimistas para a possibilidade de uma ferramenta deste teor auxiliando seu aprendizado.

Além da contribuição no desenvolvimento de Jogos educacionais, outra contribuição deste trabalho é demonstrar a efetividade do uso desses jogos como meio de aperfeiçoar o processo de ensino e aprendizagem em disciplinas de Sistemas Operacionais. É possível, observando os resultados obtidos, verificar que houve um ganho na efetividade do aprendizado quando comparadas a turma que não fez o uso dos jogos e a turma que o fez.

REFERÊNCIAS

- [1] B. Stokes, "Videogames have changed: time to consider 'serious games'?" *Development Education Journal*, vol. 11, p. 12, 01 2005.
- [2] L. Machado, R. Moraes, F. Nunes, and R. da Costa, "Serious games based on virtual reality in medical education." *Revista Brasileira de Educação Médica*, vol. 35, pp. 254–262, 01 2011.
- [3] A. F. Zorzo, D. Nunes, E. Matos, I. Steinmacher, J. Leite, R. M. Araujo, R. Correia, and S. Martins, *Referenciais de Formação para os Cursos de Graduação em Computação*. [Online]. Available: <https://www.sbc.org.br/documentos-da-sbc/send/127-educacao/1155-referenciais-de-formacao-para-cursos-de-graduacao-em-computacao-outubro-2017>
- [4] S. D. Sena, S. S. Schmiegelow, G. M. B. C. D. Prado, R. P. L. D. Sousa, and F. A. P. Fialho, "Aprendizagem baseada em jogos digitais: a contribuição dos jogos epistêmicos na geração de novos conhecimentos." *Renote*, vol. 14, no. 1, 2016.
- [5] P. H. Gurgel, E. F. Barbosa, and K. C. Branco, "A ferramenta netkit e a virtualização aplicada ao ensino e aprendizagem de redes de computadores," in *XXXII Congresso da Sociedade Brasileira de Computação (CSBC)*, vol. 7, 2012.