

UNIVERSIDADE REGIONAL DO NOROESTE DO ESTADO DO RIO
GRANDE DO SUL - UNIJUÍ
DEPARTAMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO
CAMPUS SANTA ROSA

COMPUTAÇÃO GRÁFICA
TRABALHO FINAL - ACUNIS

ALEX JUNG CELMER
GABRIEL C. ULLMANN

SANTA ROSA
2020

SUMÁRIO

1	TRABALHO FINAL - ACUNIS	2
1.1	Introdução	2
1.2	Proposta do jogo	2
1.3	Criação do jogo	2
1.3.1	Game design	3
1.3.2	Level design	4
1.3.3	Arte visual	5
1.3.4	Música	5
1.3.5	Desenvolvimento	5
1.3.6	Testes	6
1.4	Considerações finais	6
1.5	Apêndice	6
	 REFERÊNCIAS	 7

1 TRABALHO FINAL - ACUNIS

1.1 Introdução

Esse relatório descreve o processo de planejamento e desenvolvimento do jogo Acunis, criado para a disciplina de Computação Gráfica do curso de Ciência da Computação da Unijuí. O desenvolvimento do trabalho ocorreu entre os meses de abril e julho de 2020.

1.2 Proposta do jogo

Acunis é um simulador de jogo de sinuca que explore conceitos de Física. O jogador pode controlar várias forças e grandezas físicas, como a gravidade do campo de jogo, massa das bolinhas, força das tacadas, etc. Cada opção vem acompanhada de um breve texto que explica do ponto de vista científico as ações que o jogador pode tomar no ambiente virtual. Embora o foco do jogo seja "brincar" com a Física, o jogador poderá optar também por conduzir a partida de sinuca normalmente.



Figura 1 – Menu principal do jogo

O nome do jogo corresponde à palavra "sinuca" ao contrário e foi escolhido por passar uma imagem inusitada e descontraída. De forma análoga, os experimentos físicos que o jogador realiza dentro do ambiente do jogo podem levar a resultados inesperados e cômicos.

1.3 Criação do jogo

O jogo foi criado utilizando a engine Unity, que foi escolhida por ser de uso gratuito para projetos não-comerciais e por possuir uma comunidade ativa prestando suporte e produzindo conteúdo para os usuários (SHAH, 2017). A criação do jogo foi dividida em 6 partes nesse relatório: game design, level design, arte visual, música, desenvolvimento e testes. As atividades realizadas em cada fase serão explicadas com mais detalhes a seguir:

1.3.1 Game design

A fase de game design envolve o planejamento de como o jogo vai ser, quais serão os desafios apresentados ao jogador e quais possibilidades ele terá dentro do jogo. Segundo Sato (2010), "o game design visa propor problemas aos jogadores e, ao mesmo tempo, viabilizar ferramentas, ações e significados para a solução destes problemas.". Após avaliarmos as possibilidades oferecidas pela engine Unity em termos de controle de Física, definimos que o jogador poderia manipular as seguintes características:

a. Das bolinhas de sinuca:

- Massa
- Elasticidade

b. Do taco de sinuca:

- Força da tacada
- Direção/ângulo da tacada

c. Do ambiente de jogo:

- Resistência do ar
- Força da gravidade

d. Da mesa de sinuca:

- Atrito estático
- Atrito dinâmico
- Ângulo de inclinação da mesa
- Sentido de inclinação da mesa

O jogador pode alterar o comportamento da Física a qualquer momento da partida pressionando ESC, o que fará o jogo ser pausado e abrirá um menu com opções. Para continuar a partida, basta pressionar ESC novamente.

Quanto aos modos de jogo, decidimos por criar somente um *multiplayer* local no qual 2 jogadores alternariam rodadas utilizando o mesmo teclado/mouse. Tomamos essa decisão pois ambos membros do projeto possuem pouca experiência com a engine Unity, e portanto seria trabalhoso criar um bom modo *singleplayer* em tempo hábil para a entrega do trabalho. Isso

ocorre não por causa de complexidades nas mecânicas de jogo em si, mas sim pelo fato de que seria necessário desenvolver uma inteligência artificial capaz de jogar de forma relativamente balanceada contra o jogador humano.



Figura 2 – Opções de Física disponíveis no menu pausa

1.3.2 Level design

O level design consiste na criação dos desafios e cenários que o jogador verá ao longo do jogo. Segundo Farias et al. (2016) nessa etapa é realizado "um planejamento e detalhamento de tudo que fará parte do jogo, bem como as descrições das fases/níveis que compõem o mesmo". No caso do Acunis, há somente 1 cenário, que consiste em uma mesa de sinuca com 15 bolinhas coloridas e 1 branca. Embora hajam vários modos de se jogar sinuca, optamos pelo estilo de partida mais direto: o primeiro jogador a encaçapar 8 bolinhas vence. Não há limite de tempo.



Figura 3 – Uma partida de Acunis em curso

Os jogadores podem alterar a física da partida a qualquer momento através do menu pausa, que se abre ao pressionar a tecla ESC. Algumas mudanças tornam a partida impossível de jogar (por exemplo, jogar em gravidade zero). Contudo, mantivemos essas opções de forma que o jogador possa observar seus efeitos e entender sua relação com o estudo da Física. Cada opção vem acompanhada de uma breve descrição explicando o conceito do ponto de vista da Física,

acompanhada por referências bibliográficas que o jogador pode consultar posteriormente caso desejar.



Figura 4 – Explicação sobre gravidade acessível dentro do jogo

1.3.3 Arte visual

Os modelos 3D e texturas das bolinhas, mesa de sinuca e cenário foram obtidas gratuitamente através da Unity Asset Store. Os pacotes de assets utilizados foram: Free Barcade Asset Pack (FEROCIOUS, 2020) e Wooden Floor Pack (MIKELARG, 2020).

1.3.4 Música

A música foi composta utilizando o tom de Mi menor (Em) com escala pentatônica, tendo os batimentos por segundo em 180 (180BPM) e o compasso da música corre em 4x4. Foram gravados contra-baixo e guitarra, a guitarra possui várias faixas para divisão de base e solo, a faixa de bateria foi utilizada uma pronta sem direitos autorais que é utilizada para estudos.

1.3.5 Desenvolvimento

O desenvolvimento do jogo foi realizado utilizando a engine Unity, que fornece ferramentas para controle de vários sistemas comuns a diversos jogos como colisão, física, iluminação e construção de interfaces de usuário (UI). A lógica da aplicação foi criada através de scripts na linguagem C#, embora a engine suporte também Boo e UnityScript (UNITY, 2014). Escolhemos C# por conta familiaridade da linguagem com outras que os membros do grupo já dominam, por exemplo, Java e PHP. Foram criados scripts para:

- Controle de pontuação
- Checagem de condições de vitória
- Manipulação da física
- Controle de eventos da interface (ex: verificar se um botão foi clicado e realizar uma determinada ação)

1.3.6 Testes

O jogo foi testado a cada passo do desenvolvimento, mas também, de forma mais completa, após o término do desenvolvimento. O objetivo principal dos testes foi determinar se o jogo se comportava conforme o esperado e se havia problemas na parte visual. Não foram realizados testes com foco em avaliar a qualidade didática ou lúdica do jogo, passos que seriam cruciais caso o objetivo fosse tornar o game uma experiência completa e publicável.

1.4 Considerações finais

O trabalho ofereceu aos integrantes do grupo a oportunidade de ver e trabalhar em cada parte do processo de produção de um jogo eletrônico independente. Além disso, através da engine Unity foi possível colocar em prática vários conceitos de Computação Gráfica vistos em aula, como transformações e iluminação. É claro que, diferente das bibliotecas do OpenGL que foram utilizadas no início da disciplina, ao trabalhar com uma engine muita da matemática necessária para posicionar os objetos é abstraída pela ferramenta, o que torna o trabalho muito mais simples e ágil do ponto de vista do programador, embora ainda seja possível identificar os conceitos base por trás dessas ações.

Embora o game não tenha sido testado por uma ampla base de jogadores, acreditamos que ele seria uma ferramenta educacional útil para ensinar conceitos de Física de forma lúdica e interativa. Embora o jogo não tenha o conteúdo e organização de uma aula completa, poderia ser utilizado como material complementar para aulas do Ensino Médio, período no qual os estudantes aprendem sobre vários fenômenos físicos presentes no Acunis, como gravidade e atrito.

Temos ciência de que várias melhorias poderiam ser feitas, em especial na parte das interfaces e menus. Quanto às mecânicas de jogo, a única forma de avaliar seu valor didático e lúdico seria testando essa experiência com um maior número de jogadores. Contudo, visto que o trabalho não terá continuidade além da disciplina, essas são questões que ficam em aberto.

1.5 Apêndice

- [Executável para Windows \(32/64 bits\)](#)
- [Repositório do projeto no GitHub](#)
- [Vídeo de gameplay](#)

REFERÊNCIAS

FEROCIOUS, I. **Free Barcade Asset Pack**. 2020. Disponível em: <<https://assetstore.unity.com/packages/3d/props/free-barcade-asset-pack-123704>>.

HOUNSELL, M. F. e Isabela Gasparini e M. Jogos digitais educacionais para alfabetização matemática: Levantamento de habilidades e level design. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**, v. 27, n. 1, 2016. Disponível em: <<https://br-ie.org/pub/index.php/sbie/article/view/6724>>.

MIKELARG. **Wooden Floor Pack**. 2020. Disponível em: <<https://assetstore.unity.com/packages/2d/textures-materials/wood/wooden-floor-pack-31492>>.

SATO, A. K. O. **Game Design e Prototipagem: Conceitos e Aplicações ao Longo do Processo Projetual**. 2010. Disponível em: <http://www.sbgames.org/papers/sbgames10/artanddesign/Full_A&D_10.pdf>.

SHAH, V. **Reasons Why Unity3D Is So Much Popular In The Gaming Industry**. 2017. Disponível em: <<https://medium.com/@vivekshah.P/reasons-why-unity3d-is-so-much-popular-in-the-gaming-industry-705898a2a04>>.

UNITY. **Documentation, Unity scripting languages and you**. 2014. Disponível em: <<https://blogs.unity3d.com/2014/09/03/documentation-unity-scripting-languages-and-you/>>.