MATEMÁTICA APLICADA AOS GAMES

Uma abordagem teórica e prática para desenvolvedores

Cristiano Natal Tonéis

Editora Clube dos Autores

São Paulo
2015

Tonéis, Cristiano Natal
Matemática aplicada aos games: uma abordagem teórica e prática para desenvolvedores / Cristiano Natal Tonéis. São Paulo: ArteSam, 2015. 255p.

ISBN: 978-85-5697-054-1

1. Matemática aplicada - Programação. 2. Programação matemática - Games. 3. Teoria dos jogos - Programação. I. Título.

16/016 CDU 519.85

Índices para catálogo sistemático
1. Matemática 51
2. Programação matemática 519.85
3. Programação de computadores 004.42
4. Teoria dos jogos - Matemática 519.83
5. Games – Programação de computadores 519.85

(ficha elaborada por LVM – bibliotecária – CRB: 8/583)

Dedicatória

Dedico esse livro a todos que engendram os universos mágicos dos *games* nos quais podemos vivenciar histórias, aprender... Evoluir e reconstruir nosso mundo vivido.

"Como pode a matemática, sendo acima de tudo um produto do pensamento humano, independente da experiência, se adaptar tão admiravelmente à realidade objetiva?"
Albert Einstein

** INDICE **

1. Introdução.....9
1.1 Para que a matemática? E para que a matemática nos jogos digitais? 11
1.2 Os Modelos Matemáticos e os Modelos Computacionais.....13
1.3 Sistema Internacional: radianos ou graus? 16
1.3.1 Conversão entre unidades – graus e radianos 21
1.3.2 Ângulos positivos e negativos 23
2. O Plano Cartesiano 25
2.1 Plano e Espaço: Modelo Matemático e modelo computacional 27
2.2 Distância entre dois pontos no plano cartesiano.....35
2.3 Lei de Heron.....40
2.4 Ponto Médio 41
2.5 Coordenadas do Baricentro de um triângulo 42
2.6 Coordenadas do Incentro de um triângulo.....45
2.7 Coordenadas no R^3 ou 3D 46
2.8 A Distância entre pontos no espaço 3D ou R^3.....48
3. Estudo dos Vetores e os Jogos Digitais 50
3.1 Vetores: Tratamento geométrico 51
3.1.1 Operações com Vetores 51
3.1.2 Regra do paralelogramo e Regra do Polígono 51
3.1.3 Produto por escalar 53
3.2 Vetores: Tratamento algébrico 54
3.2.1 Um vetor representado por uma coordenada 54
3.2.2 Vetor no plano e no espaço em função dos versores dos eixos coordenados 56
3.2.3 Operações com vetores – álgebra 56
3.3 Produto interno ou produto escalar (*dot product*) 58
3.4 Ângulo entre dois vetores.....61
3.4.1 Exemplos de aplicação em *games* utilizando o conceito de vetores e suas operações.....65
3.5 "Vetor reflexão" no Plano 68

3.6 Paralelismo e ortogonalidade entre dois vetores 69
3.7 Produto vetorial ou produto externo (*cross product*) 71
3.7.1 Interpretação geométrica do produto vetorial.....72
3.7.2 Propriedade do Produto vetorial 72
3.8. Combinação linear de vetores 74
Observações Importantes 75
4. Matrizes e suas aplicações em jogos digitais.....76
4.1 Definição de matriz.....77
4.2 Tipos de Matrizes.....78
4.3 Operações com Matrizes 81
4.3.1 Igualdade de Matrizes.....81
4.3.2 Adição.....81
4.3.3 Subtração 82
4.3.4 Multiplicação por escalar.....83
4.3.5 Multiplicação de matrizes.....83
4.4 Aplicações nos jogos Digitais das operações matriciais 87
4.4.1 Simetria Axial.....87
4.4.2 Escala – Ampliando e reduzindo imagens.....88
4.4.3 Reflexão.....89
4.4.4 Translação.....90
4.4.5 Rotação 92
4.5 Comutatividade das transformações.....97
4.6 Combinando Transformações.....99
4.7 Outros tipos de Matrizes 104
5. Determinante de uma matriz e suas aplicações nos *games*.....112
5.1 Definição e Regra de Sarrus.....112
5.2 Teorema de Laplace 114
5.3 Condição de Alinhamento de Três Pontos 116
5.4 Sistemas de equações ou sistemas lineares 118
6. A trigonometria e os jogos digitais.....129
6.1 Uma breve revisão de funções trigonométricas.....129
6.2 Movimento circular: Equação paramétricas.....133
7. Breve estudo das Funções e suas possibilidades para os Jogos Digitais 137

7.1 Determinação da equação vetorial e equações paramétricas da reta 138
7.2 Outras aplicações para os *games*: aspectos computacionais ... 143
7.3 Função afim ou função linear 146
7.4 Equação geral e equação reduzida da reta 153
7.4.1 Casos especiais de jogos clássicos – *Space invaders* e similares 156
7.4.2 Casos especiais em jogos clássicos – *Asteroids* 159
7.5 Função quadrática: Gorilas, minhocas e aves furiosas! 161
7.5.1 Revisando e Aprendendo com os bravos! 168
7.6 Outros Recursos matemáticos para os jogos digitais 173
7.6.1 Funções compostas com valor absoluto 173
7.6.2 "Função Boomerang" 179
8. Introdução a Estatística para Jogos Digitais 182
8.1 Porcentagens 183
8.2 Jogando com as Probabilidades 186
9. Considerações finais 194
10. Atividades de Aplicação 196
Referências Bibliográficas 209
Anexos
Anexo 1: Código criado em html5- naves espaciais 215
Anexo 2: Algoritmo *pathfinding* A* 221
Anexo 3: Operações com vetores em C++ 228
Anexo 4: *Minigame* de carrinho de corrida (XNA) 231
Anexo 5: Guia de referência da utilização de vetores em *games* – problemas frequentes 238
Anexo 6: Rotação de uma imagem com ponto pivô (controles) 249
Anexo 7: Lançamento Oblíquo (código em XNA) 251
Anexo 8: Cálculo da equação do segundo grau 254

1. Introdução

Existiam para os gregos os conceitos de *Technê* e *Episteme*. *Technê* se refere à capacidade de produzir um objeto por meios racionais alcançando ainda à ideia de artesanato ou arte e também se encontra na raiz do que chamamos de *tecnologia.* A palavra tecnologia tem origem grega – *τεχνη* (*technê*): "ofício, artefato" e *λογια* (*logia*): "estudo" – sendo um termo empregado para descrever um conhecimento bem como as ferramentas, processos e os materiais criados ou utilizados a partir de tal conhecimento. Toda a construção de artefatos se torna uma arte criativa do ser humano, e se realiza por meio da combinação de conhecimento, prática e experimentação, ou seja, pela *Technê*. Enquanto a *episteme* se assemelha ao que conhecemos como conhecimento científico, em termos gregos representa o conhecimento fundamentado, contextualizado e verdadeiro (em termos lógicos).

A técnica para desenvolver ou criar *games* é uma área razoavelmente nova no Brasil. Em 2003, especificamente, surgiu o primeiro curso de jogos digitais no Brasil[1], oferecendo a seus ingressantes uma formação específica para criação e produção de *games*. Antes disso, esse mercado agregava profissionais de diferentes setores, sejam artistas ou da computação, como desenvolvedores de sistemas e cientistas da computação. Porém foi por meio da abertura de cursos específicos, que os jogos digitais também inauguraram um novo mercado de trabalho e de novas oportunidades para empresas na área de *games.*

[1] Oferecido pela Universidade Anhembi Morumbi, inicialmente como curso de Tecnologia em Jogos Digitai. Fonte: e-MEC. Disponível em <http://emec.mec.gov.br/>. Acesso em jan 2016.

Tenho experiência e vivência no universo dos *games*, como professor e pesquisador em Matemática Aplicada aos Jogos Digitais, em orientações de projetos de iniciação científica e produção de *games* protótipos, presenciei muitas das particularidades decorrentes do processo de produção de jogos digitais. Desde profissionais que se envolvem exclusivamente com a arte e *design* até futuros e promissores programadores para *games*.

Entre as diversas ciências que compõem o núcleo de um profissional em jogos digitais certamente a interdisciplinaridade é um fator comum e presente em todas. todo o processo criativo pode ser comprometido se o profissional em jogos não souber se comunicar com os responsáveis pela implementação ou programação (código do jogo). Não basta ser um exímio desenhista, ou roteirista, ou ainda um brilhante *level design,* torna-se necessário conhecer o processo em sua totalidade.

O mercado de produção de *games* exige dos profissionais conhecimentos que ultrapassem suas funções primárias, por exemplo: um artista gráfico também precisa conhecer a *game engine* para qual está criando, se necessário deve ser capaz de implementar seus cenários, pois assim compreende qual a melhor maneira de produzi-los para um determinado projeto e otimizar seu trabalho. Da mesma forma, um programador vai além de programar um único jogo, ao resolver um problema ele está resolvendo uma categoria de problemas e desse modo otimizando o processo de implementação, criando rotinas, funções ou procedimentos.

De fato, muitas funções e recursos computacionais estão disponíveis e prontos para o uso nas *game engines*, no entanto por mais "poderosa" ou completa que seja é necessário, durante o planejamento (projeto) de um jogo digital, determinar para qual plataforma o *game* será criado (consoles de videogames,

computadores, mobiles, redes sociais – *web*). Essa escolha guiará o processo criativo, desde a produção de cenários (arte e *design*) até a programação (implementação do jogo).

Computacionalmente se utilizam as chamadas bibliotecas, disponíveis para programação, e quando essas são carregadas para a memória do computador passam a ocupar um espaço referente a todos os recursos existentes nessa biblioteca. No entanto essa ação não é muito prática se, por exemplo, utilizarei somente uma pequena parte dos recursos da biblioteca, então seria mais adequado que eu criasse as minhas bibliotecas. Com isso a matemática aplicada aos jogos digitais emerge como uma nova aplicação da Matemática clássica (Geometria Analítica; Álgebra Linear; Geometria; Estatística; Cálculo Numérico) e também da Lógica assumindo um novo olhar sobre os processos computacionais que envolvem a produção de um jogo digital.

1.1 Para que a matemática? E para que a matemática nos jogos digitais?

Duas questões inquietantes, a primeira me acompanhou desde que ingressei no magistério público até o magistério superior em cursos de Tecnologia em Jogos Digitais e Ciências da Computação. Acredito que a segunda é uma consequência do modo como eu procurei pesquisar[2] a Matemática. Então para que a Matemática? E para que a Matemática nos *games*?

Realmente é muito difícil expressar em poucas palavras para que a matemática e toda minha reflexão não conseguiria expressar o significado da Matemática; porém posso falar um pouco de forma geral.

A pergunta parece simples, mas para respondê-la em primeira instância, afirmo que a Matemática pela Matemática

[2] Consequência de minhas pesquisas, de mestrado e doutorado, envolvendo matemática e jogos digitais.

encontra em si sua aplicação, ou seja, quem faz Matemática utiliza Matemática. No entanto, a Matemática se faz presente quando outras áreas do conhecimento necessitam resolver problemas e buscam modelos matemáticos produzindo aplicações para soluções de problemas específicos. Então o alcance da matemática não se restringe a ela própria! A Matemática auxilia na modelagem e previsão de fenômenos visando uma metodologia precisa e auxiliando a análise final de dados.

Para responder a segunda questão apresento dois aspectos igualmente importantes na produção de *games*:

1. Um jogo digital precisa ser implementado ou programado;
2. Um programador faz uso de modelos matemáticos, métodos numéricos e modelos computacionais para a implementação.

Assim, a Matemática nos jogos digitais visa fornecer modelos e métodos numéricos que auxiliem na produção de mecânicas e de *Level Designs* inovadores para o universo dos *games*.

Nesse livro desejo apresentar e utilizar os modelos matemáticos adequadamente a partir de escolhas computacionais ou *game engines*, oferecendo aos desenvolvedores (alunos de cursos de jogos, desenvolvedores *indie,* programação de *games, etc)* uma apresentação sintética de tópicos de suma importância para o planejamento e produção de jogos digitais. Trata-se de um tema muito amplo para o qual realizo um recorte prático evidenciando a impossibilidade de esgotá-lo.

1.2. Os Modelos Matemáticos e os Modelos Computacionais

Em 1936 Alan Turing[3] elaborou uma máquina, inicialmente teórica ou hipotética, que poderia executar processos mecânicos como uma pessoa, apresentando o que ficou conhecido como "Máquina de Turing".

Turing estava convencido de que operações de cálculo eram somente um dos tipos de sistemas formais que poderiam ser "imitados" pelos computadores. Em particular, Turing percebeu como as "tabelas de sua máquina teórica" poderiam se tornar elementos de uma poderosa gramática que as máquinas utilizariam para modificar suas próprias operações, ele se referia a criação de códigos ou linguagens de programação que seriam utilizadas para gerar todo tipo de ação pela máquina.

Parece-me que Turing realizou uma espécie de "previsão" a respeito dos códigos para computadores ou suas linguagens de programação e esta intima relação com a matemática. Com o avanço tecnológico foram desenvolvidas linguagens de programação de alto nível[4] que deixaram no passado os códigos de máquina, porém a necessidade da matemática para a programação ainda é uma realidade. Quando eu quero que a máquina resolva um problema é necessário identificar e utilizar modelos matemáticos e modelos computacionais.

Os modelos matemáticos são compostos por fórmulas, os teoremas, as teorias matemáticas que estão registradas na cultura acadêmica e escolar enquanto que os modelos computacionais formam as estruturas ligadas a máquina como o algoritmo e a

[3]A Tese de Allan Turing: TURING, Alan. *On Computable Numbers, With an Application to the Entscheidungs problem*, Proceedings of the London Mathematical Society, Series 2, vol. 42, 1936.

[4] As linguagens de alto nível são aquelas que possuem a sintaxe em inglês. As linguagens de baixo nível eram escritas em linguagem de máquina, como por exemplo, o *Assembler*.

lógica utilizada na programação (lógica booleana; lógica difusa ou *fuzzy; etc*); a linguagem de programação (*C; C++; C#, Java; Java Script*, etc) bem como ferramentas computacionais como, por exemplo, uma *game engine.* Nesse livro, ofereço uma reflexão concisa a respeito da matemática aplicada aos jogos digitais, ou seja, possibilidades de aplicações de modelos matemáticos a problemas que envolvem mecânicas e programação nos jogos digitais. No esquema a seguir (Figura 1) apresento um fluxograma básico para resolução de problemas por meio do computador:

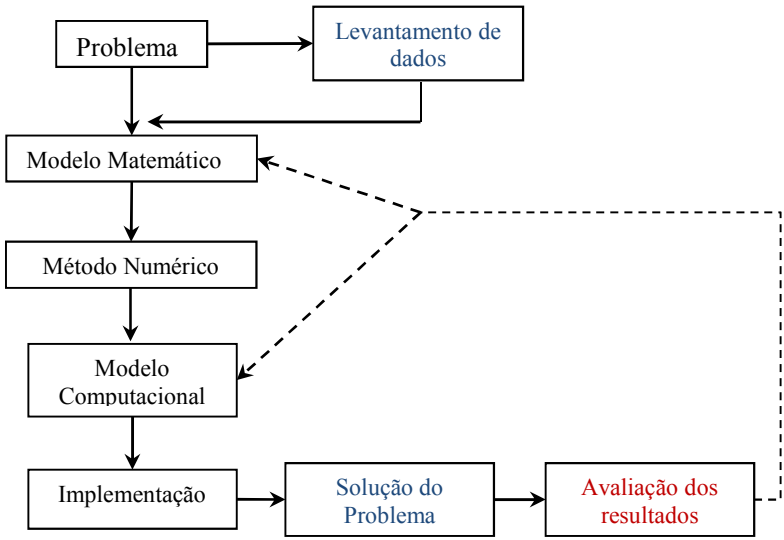


Fig. 1: Esquema para resolução de problemas em computação

Supondo que eu queira resolver um problema, por exemplo determinar as raízes de uma função quadrática:

$$y = x^2 - 5x + 6$$

Computacionalmente o que eu farei será resolver mais do que uma questão, não seria muito útil criar um programa que resolva apenas as raízes para essa função. É interessante que a máquina seja capaz de fornecer as raízes para qualquer função do segundo grau. Então meu problema é um exemplo que ao ser generalizado poderá resolver qualquer questão semelhante por meio do computador, ou seja, equações do tipo $y = ax^2 + bx + c$.

Minha equação $x^2 - 5x + 6 = 0$ passa a ser um exemplo do problema que desejo resolver. Primeiro busco um modelo matemático, neste momento entra em ação minha experiência com a matemática e os modelos que eu conheço. Recordo que posso resolver essa equação por meio de um algoritmo conhecido como a *fórmula de Bháskara*[5]. Certo, então escolhi meu modelo matemático, em seguida preciso decidir como vou implementar o cálculo para produzir a solução: a linguagem de programação e a estrutura lógica. Escolhi a linguagem C. Pronto! Agora que eu defini os modelos (matemático e computacional)[6] posso implementar[7] e depois testar a solução encontrada para o problema, nesse caso:

$x = 2$ ou $x = 3$.

Quando o programa está rodando eu consigo fazer com que a máquina resolva qualquer função do segundo grau que eu digite, então o que eu fiz foi resolver não apenas um problema,

[5] É interessante notar que esse nome é dado apenas pelos livros de matemática no Brasil. Este algoritmo para resolver equações do segundo grau é muito mais antigo que o registrado por Bháskara. De certo que os egípcios, os gregos sabiam como resolver problemas desse tipo, no entanto não possuíam a álgebra tão desenvolvida a ponto de escrever fórmulas com uso de letras.

[6] Além disso, ainda existem os métodos numéricos, ou seja, as condições de existência do problema, a precisão da máquina, critérios para tolerância de erros.

[7] No Anexo 8 deixo o código em C apresentando uma resolução.

mas toda uma categoria de problemas, pois o aplicativo funciona para encontrar as raízes de qualquer função do segundo grau ($y = ax^2+bx+c$).

Nesse processo a avaliação dos resultados é de suma importância, pois a partir dela sou capaz de verificar se os modelos escolhidos (matemáticos e computacionais) foram os melhores para resolver esse tipo de problema. No exemplo eu teria ainda outros modelos matemáticos disponíveis:

Métodos diretos	***Métodos iterativos***
"Fórmula de Bháskara"	*Método de Newton Raphson*
Fórmula de Cardano e Tartaglia	*Método da bissecção*

Esses modelos matemáticos, entre outros, também solucionam o meu problema das raízes da função quadrática, no entanto se eu não os conheço não poderia escolher ou avaliar os resultados encontrados.

Então a Matemática Aplicada aos Jogos Digitais procura fornecer modelos matemáticos que possam ampliar as soluções para problemas envolvidos na produção de jogos digitais – aspectos do *design* e da programação – com o auxílio também dos modelos computacionais.

O caminho para conhecer e compreender novos modelos matemáticos pode ser longo, mas também pode ser muito gratificante, pois iluminam os modelos computacionais.

1.3 Sistema Internacional[8]: Radianos ou Graus?

O Sistema Internacional de Medidas (*SI*) é uma convenção global de unidades de medidas que determina unidades

[8] *International System of Units*, as calculadoras científicas possuem a opção para operarem em graus (*degree*), em radianos (*rad*) ou grados (*gra ou gon*).

específicas para cada tipo de medição. Por exemplo, no *SI*, a unidade para medir áreas é o metro quarado (m^2).

Para medir ângulos utilizo os radianos, desse modo: 1,35 radianos podem ser representados como: 1,35 *rad*; 1,35 *rd* ; $1{,}35^{rad}$ ou ainda $1{,}35^{c}$. Lembre-se que computacionalmente é o ponto decimal "." no lugar da vírgula "," que se deve utilizar como separador decimal.

Mas, por que medir ângulos em radianos é melhor do que em graus? Essa é uma questão pertinente nesse trabalho, pois geralmente a matemática escolar não enfatiza a importância dos radianos e como consequência a medida de ângulos em graus (°) tornou-se amplamente difundida.

O sistema de medida em radianos[9] tem algumas vantagens em relação a outros sistemas por isso tornou-se uma unidade *default*[10] para o computador.

O sistema em radianos permite medir simultaneamente duas instâncias da circunferência: o ângulo central e o comprimento do arco gerado por esse ângulo (essa foi, provavelmente, a principal vantagem na medição em radianos).

Um radiano (*1rad*) é a medida de um raio (*radius*), em uma circunferência de raio unitário (figura 2).

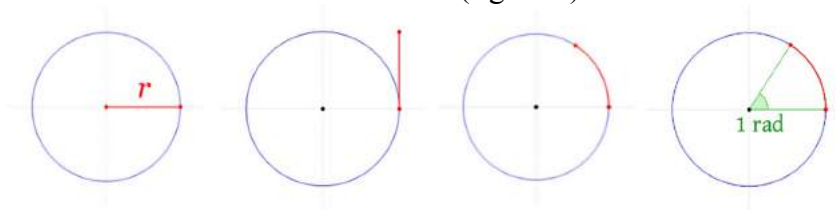


Fig. 2: Representação de *1 rad* na circunferência de raio unitário

[9] Math is Fun! *Radians*. Disponível em <https://www.mathsisfun.com/geometry/radians.html>. Acesso em jan 2016.

[10] Por *default*, é uma expressão computacional que significa como valor padrão.

Por isso uma semicircunferência possui um pouco mais que 3 *rad*, ou seja, o número que ficou conhecido como $\pi\, rad$ ($\pi = 3{,}1415927...$). Na figura 3 ilustrei uma volta completa na circunferência (2π) isso implica que o comprimento dessa circunferência (raio = 1) também é 2π . É muito comum, nos jogos digitais a utilização de valores truncados para expressar esse valor (uma volta = 6,28) porque o computador utiliza sempre valores numéricos[11] (reais) para representar um ângulo.

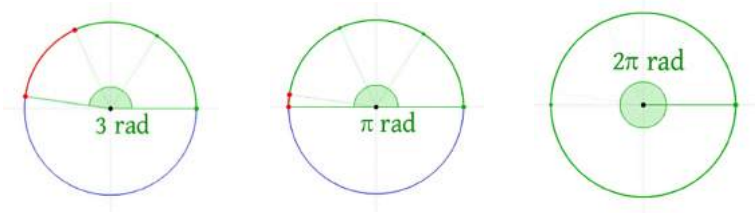


Fig. 3: Ilustração para π (semicircunferência) e 2π radianos.

Desse modo a fórmula do comprimento da circunferência de *raio r* (qualquer) é dada por: $C = 2.\pi.r$

[11] É importante sabermos que mesmo "aparentemente" operando com números decimais a base de operações do computador é binária ou hexadecimal. Desse modo sempre existirá um pequeno erro de conversão que é admitido pela máquina, por isso devemos nos preocupar com os processos de arredondamento e truncamento para não afetarem nosso projeto de modo prejudicial.

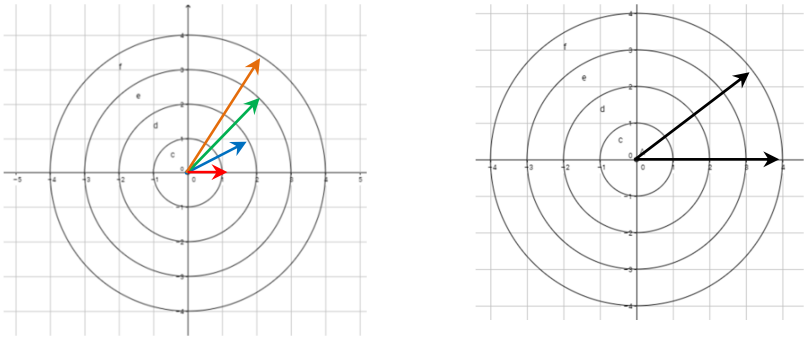


Fig. 4: À esquerda as circunferências com raios e comprimentos diferentes. À direita o mesmo ângulo define arcos de comprimentos diferentes devido a medida do raio. Imagens produzidas no *Geogebra*.

Exemplos

1) Determinar o comprimento da circunferência cujo raio mede 5 *pixels*.

$$C = 2.\pi.r$$

$$C = 2.\pi.5 = 10\pi$$

$$C \cong 31,4$$

2) Qual é o raio de uma circunferência que possui um comprimento de 157 *pixels*?

$$C = 2.\pi.r \Rightarrow r = \frac{C}{2.\pi}$$

$$r = \frac{157}{2.\pi} \cong 24,98$$

3) Calcular a medida de um arco de 30º em uma circunferência de 10 *pixels* de raio.

Solução: Em radianos teremos apenas que fazer o ângulo vezes o raio. Nesse caso transformamos 30° em radianos e depois aplicamos a fórmula.

$$l = \frac{\pi}{6}.10 \cong 5,24$$

As medidas em radianos não necessariamente precisam vir acompanhadas de um de seus símbolos, podemos simplesmente dizer:

$\theta = 1,3$ *ou* $\theta = 5$ *ou* $\theta = 2,72$

Quando eu expresso um ângulo de forma decimal então esse ângulo está em radianos. No processamento do computador e nas operações realizadas a máquina está lidando com constantes conversões entre valores decimais[12].

Veja o seguinte exemplo, quero localizar na circunferência o ângulo de 40°30' (quarenta graus e meio ou quarenta graus e trinta minutos)[13]. Trinta minutos é metade de um grau, pois sabemos que 41° > 40°30', porém sua localização e representação pela máquina (computador) passam a ser imprecisas, enquanto que, em radianos, escrevemos 0,7068584.

Outro exemplo, ao tratar do cálculo de ângulo entre vetores, o resultado será sempre um valor em radianos, ou seja, um número decimal.

Existem *funções em linguagens de programação*[14] que permitem a utilização de ângulos em graus, no entanto internamente está ocorrendo uma conversão e o número de operações será sempre maior do que imaginei. Então é útil

[12] Sabemos que os computadores operam com sistema binário ou hexadecimal, dependendo do cálculo. Desse modo a máquina está constantemente convertendo os valores entre diferentes bases numéricas (decimal, binário, hexadecimal).

[13] A medida de ângulos em graus pode ser fácil enquanto os valores são inteiros, quando apresentam valores não inteiros é necessário recordar que a base (em graus) é sexagesimal (60). Então está incorreto escrever 40°,5 (ou 40,5°) quando o que se deseja é quarenta graus e meio, logo escrevo: 40°30'.

[14] Torna-se muito importante conhecer a sintaxe relacionada às funções que pretendemos utilizar quando vamos programar. Cada linguagem de programação possui sua própria sintaxe.

conhecer e saber usar as medidas em radianos, as funções trigonométricas *(sinα, cosα, tgα)* entendem que o *ângulo α* de entrada é um valor em *radianos*.

1.3.1 Conversão entre Graus e radianos e radianos e Graus

Na figura 5, a seguir, faço uma comparação entre os dois sistemas para visualizar suas equivalências.

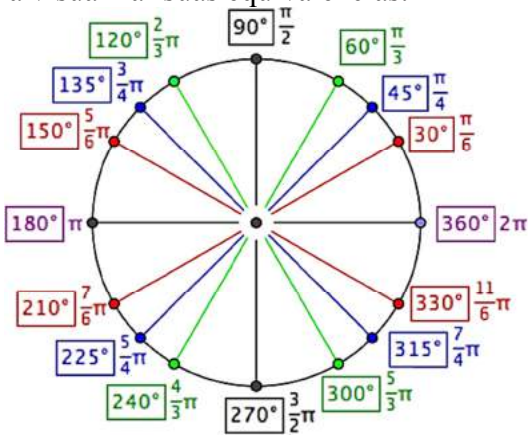


Fig. 5: Relações entre ângulos em radianos e graus.

O ângulo de $1rad \neq 1°$, pois $1rad \cong 57°17'$. Por meio de uma regra de três simples posso fazer a conversão entre os sistemas. Sendo que $\pi \equiv 180°$ (lê-se π é côngruo a 180°, ou seja, representam o mesmo giro), então:

Radianos para Graus

radianos *graus*

π _____ 180

0,523598 _____ x

$$x = \frac{180 * 0,523598}{\pi}$$

$x \cong 30$

Graus para radianos

radianos *graus*

π _____ 180

x _____ 60

$$x = \frac{60 * \pi}{180}$$

$x = 1,0471976$

Facilmente verifico que:

$$30° \equiv \frac{\pi}{6} \quad (\textit{figura 5}) \text{ e } \frac{\pi}{6} \cong 0,5235987$$

Foi desse modo que determinei que $40°30' \equiv 0,7068584$, ou seja:

radianos *graus*

π _____ 180

x _____ 40,5

(Para fins de cálculo utilize a notação decimal)

$$x = \frac{40,5 * \pi}{180}$$

$$x = 0,7068584$$

De maneira análoga posso converter radianos em graus:

radianos *graus*

π _____ 180

0,7068584 ___ x

$$x = \frac{180 * 0,7068584}{\pi}$$

$$x \cong 40,5$$

$$x \cong 40°30'$$

O valor 40,5 *não é uma notação válida para graus*, assim faço o seguinte ajuste:
40 + 0,5 = 40° (parte inteira) + 0,5 * 60 (parte dos minutos) = 40°30'
Na maioria das calculadoras científicas existe uma tecla que faz essa correção automaticamente, se no visor estiver 40,5 basta acionar a tecla da figura 6 e ela mostrará 40°30' em seu visor.



Fig. 6: Conversão para notação de graus com uso de uma calculadora científica.

1.3.2 Ângulos positivos e ângulos negativos

Para concluir essa explanação a respeito dos tipos de ângulos, por convenção[15], ângulos positivos são medidos no sentido anti-horário e ângulos negativos no sentido horário.

Os ângulos podem ser superiores a uma volta completa (superiores a 360° ou 2π). Nesses casos aconselho que sejam retiradas as voltas inteiras e se utilize o ângulo côngruo nas operações. Em uma animação isso não é problema, porém em caso de transformações[16] é aconselhável utilizar sempre um arco pertencente à primeira volta, sendo assim um ângulo compreendido entre 0 e 2π.

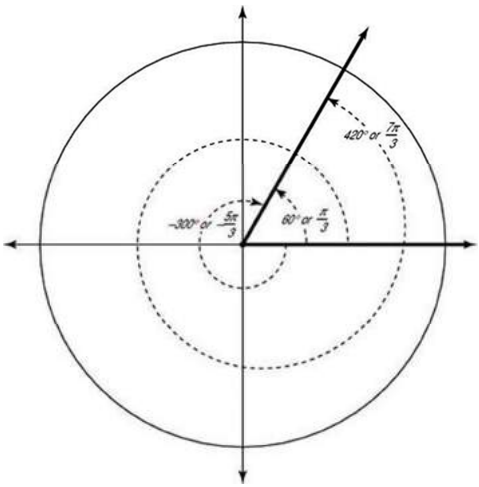


Fig 7: Medidas de ângulos positivos (sentido anti-horário) ou negativos (sentido horário).

A figura 7 ilustra a congruência entre os ângulos:

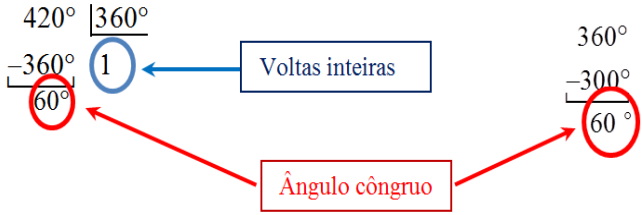
$$420° \equiv \frac{7\pi}{3} \equiv 60° \equiv \frac{\pi}{3} \equiv -300° \equiv -\frac{5\pi}{3}$$

[15] Matemática e computacional

[16] Veremos no item 4.4 as transformações que uma imagem pode sofrer e as operações matriciais ou vetoriais envolvidas.

Em caso de um movimento que decorre em 420° (e não se trata de uma animação) então, com é aconselhável que se utilize o menor arco correspondente, ou seja, o ângulo de 60°.

Por meio de um cálculo simples elimino as voltas inteiras. Veja dois exemplos um para o ângulo de 420° e o segundo para o ângulo de -300°:



Se os valores estiverem em radianos às operações serão as mesmas. O que fiz foi uma redução a arcos da primeira volta. Esse recurso é útil para movimentos que assumem mais de uma volta, pois o efeito final é equivalente ao arco reduzido, evitando cálculos e *tempo de processamento*.

Na seção de operações com matrizes/vetores apresento as transformações que uma imagem pode sofrer e no caso de rotações sucessivas o efeito final pode ser desastroso, pois a imagem pode acabar distorcida. Então é aconselhável reduzir para efeito de uma volta e quando se utilizar muitos movimentos de rotação criar animações, *cinematics*, *cutscene,* para o efeito.

Todas as ilustrações desse livro foram criadas com elementos gráficos ou compiladas do *web site* <http://www.gameart2d.com/freebies.html>, sob licença *Creative Common Zero (CC0) a.k.a Public Domain license*.

2. O Plano Cartesiano

Cogito ergo sum – Penso, logo existo – tal proposição resume o espírito de René Descartes (1596-1650), francês cujo *Discurso do Método*[17], inaugurou a Filosofia moderna. Em 1637, em uma época em que a força da razão como se conhece era muito mais do que incipiente, e em que textos filosóficos eram escritos em latim, voltados apenas para os doutores, Descartes publicou o *Discurso do Método* redigido em língua vulgar, isto é, o francês. Ele defendia o "uso público" da razão e escreveu o ensaio pensando em uma audiência ampla. Queria que a razão – este privilégio único dos seres humanos – fosse exatamente isso, um privilégio de todos os seres humanos.

Em homenagem a Descartes (que em latim era chamado de *Cartesius)* o modo de localizar objetos por meio de pontos, criado por ele, recebeu o nome de sistema cartesiano[18]. Em suas pesquisas procurou associar a geometria e a álgebra criando uma área totalmente nova para matemática: a Geometria Analítica. Dessa forma ele pode representar graficamente expressões algébricas.

O plano cartesiano é composto pela intersecção de duas retas orientadas perpendiculares. A orientação no sentido positivo das retas é representa por uma seta, como pode-se ver na figura a seguir. A reta vertical recebeu o nome de eixo das ordenadas (y) e a reta horizontal de eixo das abscissas (x). Na

[17] *La Géométrie* é um dos três ensaios prefaciados pelo *Discours de la Méthode pour bien conduire sa raison à chercher la verité dans les sciences* (Discurso sobre o Método para Bem Conduzir sua Razão a Procurar a Verdade nas Ciências). Descartes descreve esse tratado em *La Géométrie* (1637). A ideia de representar localizações por meio de pontos e coordenadas remonta aos gregos e ainda desde a descoberta do papiro de Ahmés (cerca de 1650 a.C.), que cálculos aritméticos podem ser atrelados a alguma interpretação geométrica.

[18] O Plano Cartesiano ou Plano ortogonal cartesiano.