



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e
Computação Científica

Paulo Nelson de Oliveira

Criação de um Aplicativo de Realidade Aumentada para o Ensino da Geometria

CAMPINAS
2019

Paulo Nelson de Oliveira

**Criação de um Aplicativo de Realidade
Aumentada para o Ensino da Geometria**

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do título de Mestre.

Orientador: Roberto Andreani

Este exemplar corresponde à versão final da dissertação defendida pelo aluno Paulo Nelson de Oliveira e orientada pelo Prof. Dr. Roberto Andreani.

CAMPINAS
2019

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

OL4c Oliveira, Paulo Nelson de, 1987-
Criação de um aplicativo de realidade aumentada para o ensino da geometria / Paulo Nelson de Oliveira. – Campinas, SP : [s.n.], 2019.

Orientador: Roberto Andreani.
Dissertação (mestrado profissional) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Realidade aumentada. 2. Geometria - Estudo e ensino. I. Andreani, Roberto, 1961-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Creation of an augmented reality application for teaching geometry

Palavras-chave em inglês:

Augmented reality

Geometry - Study and teaching

Área de concentração: Matemática em Rede Nacional

Titulação: Mestre

Banca examinadora:

Roberto Andreani [Orientador]

Claudina Izepe Rodrigues

Cintya Wink de Oliveira Benedito

Data de defesa: 13-12-2019

Programa de Pós-Graduação: Matemática em Rede Nacional

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: 0000-0002-4818-7068

- Currículo Lattes do autor: https://www.cnpq.br/cvlattesweb/pkg_login.pr

Dissertação de Mestrado Profissional defendida em 13 de dezembro de 2019 e aprovada pela banca examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). ROBERTO ANDREANI

Prof(a). Dr(a). CLAUDINA IZEPE RODRIGUES

Prof(a). Dr(a). CINTYA WINK DE OLIVEIRA BENEDITO

A Ata da Defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós-Graduação do Instituto de Matemática, Estatística e Computação Científica.

Este trabalho é dedicado a minha amada esposa Karoline Meiriele Alves Oliveira que me colocou neste caminho, e que com certeza sem ela nada disso estaria se concretizando. Dedico também a meus amados pais Nelson de Oliveira e Teresa de Oliveira que sempre me apoiaram durante minha trajetória.

Agradecimentos

À Deus por me dar forças e convicção para concluir as tarefas que me confia.

À toda minha família pelo apoio.

Aos amigos do Profmat 2017: Roberto, Jonas, Reginaldo, Tiago e Deborah pelo incentivo e grande ajuda durante todo o curso.

Ao professor Doutor Roberto Andreani, exemplo de pessoa, agradeço pela disponibilidade, paciência e sabedoria com que me orientou neste trabalho.

Aos professores do IMECC pelos conhecimentos compartilhados, em especial à professora Doutora Claudina pelo exemplo que levo para a vida.

Aos professores membros da banca examinadora pela disponibilidade e atenção dispensados ao meu trabalho.

Ao Colégio Delta Nobre, em especial aos diretores Alexandre Lahni e Antônio Luiz por me darem a oportunidade de estar neste caminho.

Aos colegas da EMEB Professora Cleusa e EMEB Humberto Brasi por disponibilizarem a estrutura necessária para o desenvolvimento do projeto.

Aos alunos que prontamente aceitaram a pesquisa.

O que foi, isso é o que há de ser; e o que se fez, isso se fará; de modo que nada há de novo debaixo do sol.

Eclesiastes 1:9

Resumo

A tecnologia está mudando o mundo e conseqüentemente a educação, o que induz o professor a se adaptar mesmo que de forma parcial. Este trabalho tem como objetivo principal apresentar ferramentas necessárias para o professor juntamente com o aluno criarem um aplicativo em sala. Durante a elaboração desta dissertação foi criado um aplicativo de realidade aumentada denominado *Solid Planning*, que tem a finalidade de auxiliar a visualização dos alunos à respeito da geometria espacial. Para a criação será fundamental a utilização de três ferramentas, o primeiro é um programa para criar jogos chamado *Unity*, o segundo é um programa de modelagem chamado *Blender*, e por fim um kit de desenvolvimento de *software* chamado *Vuforia*. Como o aplicativo citado acima tem a finalidade de trabalhar com figuras em três dimensões, este trabalho dará uma base ao aluno em *Geometria Espacial*, mas não apenas isso, aqui o aluno também terá um contato básico com a programação ao trabalhar com o *Unity*, e uma base de modelagem ao trabalhar com o *Blender*. Como a ideia é que o aluno e o professor criem juntos o aplicativo, toda construção será estruturada em planos de aulas. Este trabalho será finalizado com relatos e comentários dos alunos a respeito do aplicativo criado.

Palavra-chave: Aplicativo, Realidade Aumentada, Geometria

Abstract

Technology is changing the world and consequently education, which induces the teacher to adapt even partially. This work has as main objective to present necessary tools for the teacher along with the student to create a classroom application. During the elaboration of this dissertation was created an augmented reality application called *Solid Planning*, which has the purpose of helping the students visualization about the spatial geometry. It will be fundamental to use three tools, the first is a program to create games called *Unity*, the second is a modeling program called *Blender*, and finally a *software* development kit called *Vuforia*. Since the above application is intended for working with three-dimensional figures, this work will give the student a foundation in spatial geometry, but not only that, here the student will also have a basic contact with programming when working with *Unity*, and a modeling base when working with *Blender*. As the idea is that the student and teacher create the application together, all construction will be structured in lesson plans. This paper will be completed with reports and comments from students regarding the application created.

Keyword: Application, Augmented Reality, Geometry

Lista de Figuras

1	Círculo inscrito e circunscrito	17
2	Reta numérica	18
3	Plano cartesiano	19
4	Ângulos adjacentes	19
5	Ângulos suplementares	19
6	Ângulos opostos pelo vértice	20
7	Prisma	22
8	Elementos do prisma	22
9	Planificação de um prisma	23
10	Volumes <i>Cavalieri</i>	24
11	Princípio de <i>Cavalieri</i>	24
12	Cilindro	25
13	Planificação de um cilindro	26
14	Pirâmide	26
15	Pirâmide planificada	27
16	Cone	28
17	Revolução	28
18	Planificação Cone	29
19	Gráfico de utilização do <i>Unity</i>	29
20	Tela de <i>download</i> do <i>Unity</i>	30
21	<i>Layout Unity</i> vazio	31
22	<i>Layout Unity</i> com objeto	32
23	Ferramentas de alteração de objetos	33
24	Cubo sólido	33
25	Cubo formado por 6 objetos	34
26	Variáveis na linguagem C#	35
27	Pasta <i>Scripts</i>	37
28	Interface do <i>Visual Studio</i>	37
29	Rotação no centro da face	39
30	Abertura ilimitada de uma face	39
31	Abertura em 90° de uma face	42
32	Planificação finalizada no <i>Unity</i>	43
33	Imagem de ativação	44
34	Realidade Aumentada - Planificação Cubo	46
35	<i>Fita de Mobius</i>	47
36	Plano no <i>Software Blender</i>	48
37	Construção <i>Fita de Mobius</i>	48
38	<i>Fita de Mobius</i> pré-pronta	48
39	<i>Fita de Mobius</i> finalizada	49
40	Cilindros no <i>Blender</i>	50
41	Anel no <i>Blender</i>	50
42	<i>Garrafa de Klein</i> em construção	51
43	<i>Garrafa de Klein</i> finalizada	51
44	Diagrama de navegação	52
45	Tela inicial do aplicativo	54
46	Fórmulas de quadriláteros	55

47	Fórmulas de triângulos	55
48	Fórmulas <i>Geometria Espacial</i>	56
49	Planificação cubo	56
50	Planificação prisma	57
51	Planificação pirâmide	57
52	<i>Garrafa de Klein</i>	58
53	<i>Fita de Mobius</i>	58
54	Resumo das respostas do questionário	59
55	Questionário	63

Lista de Abreviaturas e Siglas

SBM Sociedade Brasileira de Matemática

IMPA Instituto Nacional de Matemática Pura e Aplicada

PCN Parâmetros Curriculares Nacionais

PROFMAT Mestrado Profissional de Matemática

APP Aplicativo

Sumário

1	Introdução	14
2	A importância do estudo da <i>Geometria</i> no ensino básico	17
2.1	Histórico	17
2.2	O auxílio de um <i>Software</i> de acordo com os Parâmetros Curriculares Nacionais e as novas tecnologias	21
3	Planificação	22
3.1	Planificação de um prisma	22
3.2	Planificação de um cilindro	25
3.3	Planificação de uma pirâmide	26
3.4	Planificação de um cone	27
4	O <i>Software Unity</i>	29
4.1	A Utilização do <i>Unity</i>	29
4.2	<i>Microsoft Visual Studio</i>	30
4.3	<i>Layout</i> do <i>Unity</i>	31
5	Criação do Aplicativo <i>Solid Planning</i>	32
5.1	A criação das faces do cubo	32
5.2	Programando a abertura da <i>Face1</i>	34
5.3	Limitando a abertura da <i>Face1</i> em 90°	39
5.4	Planificação finalizada	42
5.5	Criando a realidade aumentada no computador com o auxílio do <i>Vuforia</i>	43
5.6	Criando a <i>Fita de Mobius</i>	46
5.7	Criando a <i>Garrafa de Klein</i>	49
5.8	Configurações finais	52
6	Navegação no <i>Solid Planning</i>	54
6.1	<i>Geometria Plana</i> no Aplicativo	54
6.2	Fórmulas <i>Geometria Espacial</i> no Aplicativo	54
6.3	Planificação no Aplicativo	56
6.4	Mais sólidos no Aplicativo	57
7	Resultados obtidos	58
7.1	Síntese do resultado	59
8	Conclusão	60
9	Referências Bibliográficas	61
10	Apêndice	63
10.1	A - Questionário aplicado aos alunos	63
10.2	B - Links para <i>Download</i>	64

1 Introdução

Nas últimas décadas a educação juntamente com o mundo vem sendo modificada com novas tecnologias a ritmos inimagináveis, e isso conseqüentemente impõe ao professor uma nova maneira e novo ritmo para ensinar. É inegável que nos últimos anos a introdução do computador na educação provocou diversas indagações sobre sua utilização, principalmente por professores à mais tempo no cargo com uma sólida estrutura didática que sempre funcionou. No entanto, é importante ressaltar que o computador tornou-se um objeto integrante no dia a dia dos alunos e professores, assim o ideal seria unir esse objeto ao aprendizado.

Esse trabalho consiste na apresentação de uma possibilidade para o ensino da geometria a partir da criação de um aplicativo de realidade aumentada. É apresentado um programa de criação de jogos e aplicativos, o qual é usado em 3 dimensões e que sendo assim pode ser apresentado aos alunos para que tenham a oportunidade de interações com a *Geometria* de forma didática, tendo que pensar o que fazer em cada situação e perceber as regularidades, estabelecendo assim, estratégias de investigação. As atividades apresentadas foram feitas norteadas na experiência do autor em trabalhar com as séries do ensino fundamental II e ensino médio. É fato que é relevante, contemporaneamente, o desenvolvimento de uma cultura informática no interior da escola e como educadores devemos estar cientes que em razão da crescente informatização da população a presença de tecnologias na instituição de ensino deve ser examinada com uma realidade que não cabe mais discutir, mas cujo real significado precisa ser construído. Na busca da obtenção por um aprendizado mais eficaz, é preciso estar preparado para gerir e fazer uso eficaz dessas novas tecnologias, e como elementos fundamentais do processo ensino e aprendizagem, sempre buscar atingir um ponto de equilíbrio.

Em conformidade com as orientações para elaboração de trabalhos de dissertação do *PROFMAT*, este enquadra-se na modalidade desenvolvimento de aplicativos. O objetivo geral deste trabalho é apresentar o caminho necessário para que o professor em sala juntamente com o aluno tenha condições de criar o aplicativo que desejar independente da disciplina.

Os objetivos específicos são:

- Mostrar como é feita a criação de um aplicativo de realidade aumentada;
- Auxiliar a compreensão de figuras geométricas por meio do aplicativo aqui criado;
- Utilizar recursos computacionais no ensino de matemática;
- Propiciar a criação de conjecturas sobre propriedades de figuras geométricas;
- Desenvolver as capacidades de raciocínio e de expressão em linguagem matemática oral, escrita e gráfica;
- Estimular a criatividade e a criticidade;
- Desenvolver a autonomia e a cooperação.

Um destaque importante a ser feito, em relação a inserção de computadores na educação, diz respeito à formação de professores para atuar em ambientes informatizados.

Para Valente (1999, p. 19):

questão da formação do professor mostra-se de fundamental importância no processo de introdução da informática na educação, exigindo soluções inovadoras e novas abordagens que fundamentam os cursos de formação.

O autor afirma que apesar da quantidade de cursos para formação de professores, em sua grande maioria esses cursos são ineficazes para educação em sala, pois os referidos cursos auxiliam apenas no aprendizado de *software*.

Cabe mencionar que, a formação do professor não tem acompanhado o avanço tecnológico, e para muitos, as mudanças pedagógicas são difíceis de serem assimiladas e implantadas na escola. O que se encontra, na maioria das escolas, são profissionais muitas vezes despreparados para usarem as tecnologias. Almeida (2000, p. 109) afirma que:

mesmo o professor preparado para utilizar o computador para a construção do conhecimento é obrigado a questionar-se constantemente, pois com frequência se vê diante de um equipamento cujos recursos não conseguem dominar em sua totalidade.

Note que a inserção das tecnologias na educação coloca em xeque alguns métodos ultrapassados na educação, segundo Mario Sergio Cortella:

A escola precisa dar uma distinção entre aquilo que, vindo do passado, já não serve mais, o que a gente chama de arcaico, daquilo vindo do passado que precisa ser preservado, guardado, que é a tradição. Uma das bandeiras arcaicas é lidar com o conhecimento como se ele fosse inamovível, absoluto e indubitável. Portanto, tudo isso tem que ser ultrapassado para que a gente consiga construir uma capacitação das pessoas.

Portanto, para que ocorra uma ruptura significativa com esse modelo ultrapassado é necessário que os professores assumam uma postura reflexiva em relação à sua formação. É preciso estudar maneiras de propiciar um aprendizado mais eficaz. Nesse sentido, é preciso estar atento às novas tecnologias, e como elementos fundamentais do processo ensino-aprendizagem, sempre buscar atingir um ponto de equilíbrio.

Em relação ao uso do computador em sala de aula, segundo as Diretrizes Curriculares Nacionais de Educação para o Ensino Médio (2011 - PPP/Cap. VIII):

Concretamente, o projeto político-pedagógico das unidades escolares que ofertam o Ensino Médio deve considerar: VIII – utilização de diferentes mídias como processo de dinamização dos ambientes de aprendizagem e construção de novos saberes

Essa consideração apontada pelas Diretrizes Curriculares enfatiza a necessidade de análise das tecnologias em sala de aula, não apenas as que os colégios disponibilizam mas também as que os alunos utilizam durante as aulas como os celulares, e trabalhá-las na construção de novos saberes. Os professores devem ficar atentos à utilização sistemática de *Softwares* produzidos fora do âmbito da escola. Em princípio, todo professor pode ser usuário e autor dos programas para a rede. Os educadores precisam trabalhar a modernidade em favor de suas disciplinas, pois hoje há um número elevado de alunos que possuem esses aparelhos e muitos deles têm acesso à internet, outro instrumento que pode ser utilizado em sala de aula para pesquisas. CARVALHO (2000, p. 15) afirma que:

A educação em suas relações com a Tecnologia pressupõe uma rediscussão de seus fundamentos em termos de desenvolvimento curricular e formação de professores, assim como a exploração de novas formas de incrementar o processo ensino-aprendizagem.

Os autores acentuam a necessidade da formação dos professores quanto às tecnologias que se apresentam em sala de aula, mesmo quando se pensa nas TIC (Tecnologia de Informação e Comunicação) como os computadores, até quando se pensa nos aparelhos celulares, mais acessíveis em sala de aula, ou seja, os professores precisam preparar-se frente à realidade tecnológica da escola e dos próprios alunos.

Com relação ao uso de aplicativos e programas para educação qualquer professor pode ser simultaneamente usuário e produtor de materiais e, no entanto, a maioria vive frequentemente sob jugo de um material preparado por outro. Este trabalho tira o professor da zona de conforto, mas não apenas isso, pois definitivamente o objetivo não é criar alunos “apertadores de botões”, e é por esse motivo que este trabalho consiste não apenas na utilização do aplicativo mas também na criação do mesmo, para que o aluno durante a criação possa de acordo com seu senso crítico criar ferramentas melhores para o ensino/aprendizado. Segundo Gravina (1997, p.1):

No contexto da Matemática, a aprendizagem nesta perspectiva depende de ações que caracterizam o “fazer matemática”: experimentar, interpretar, visualizar, induzir, conjecturar, abstrair, generalizar e enfim demonstrar. É o aluno agindo, diferentemente de seu papel passivo frente a uma demonstração formal do conhecimento, baseada essencialmente na transmissão ordenada de “fatos”, geralmente na forma de definições e propriedades. Numa tal apresentação formal e discursiva, os alunos não se engajam em ações que desafiem suas capacidades cognitivas, sendo-lhes exigido no máximo memorização e repetição, e conseqüentemente não são autores das construções que dão sentido ao conhecimento matemático.

Este trabalho é justamente o “fazer matemática” citado pela autora, pois aqui, o aluno terá a raiz da criação para que experimente, interprete, abstraia, demostre e crie. O ponto é que não basta apenas colocar à disposição do estudante um programa de geometria dinâmica, são os desafios propostos pelo professor que vão orientar o trabalho, e assim os desafios construirão o genuíno interesse dos alunos. Logo, o uso das novas tecnologias na sala de aula é um grande aliado no ensino da matemática, desde que se faça um planejamento adequado com objetivos a serem alcançados.

2 A importância do estudo da *Geometria* no ensino básico

2.1 Histórico

Quando se pensa em *Geometria* algumas imagens e conceitos vêm à mente. Sabe-se que a *Geometria*, segundo Ferreira (1999, p.983) é “um ramo da matemática que estuda as formas, plana e espacial, com as suas propriedades, ou ainda, ramo da matemática que estuda a extensão e as propriedades das figuras (*Geometria Plana*) e dos sólidos (*Geometria Espacial*). Ainda pode-se acrescentar que de acordo com Boyer (1996, p. 5), “o desenvolvimento da geometria pode ter sido estimulado por necessidades práticas de construção e demarcação de terras, ou por sentimentos estéticos em relação a configurações e ordem”. *Geometria* é uma palavra de origem grega que significa: “geo”, terra, e “metria”, que vem da palavra “métron” e significa medir. Sendo assim, a *Geometria* é uma ciência que se dedica a estudar as medidas das formas de figuras planas ou espaciais, bem como sobre a posição relativa das figuras no espaço e suas propriedades. Na história da *Geometria* destacam-se alguns matemáticos por estudarem esse conteúdo, matemáticos esses que são chamados de geometras. Destacaram-se geométricos como Arquimedes, Descartes, Tales de Mileto, Euclides (considerado o pai da Geometria), entre outros. Seguem algumas descobertas dos grandes geometras.

A aproximação do valor numérico do número pi (*Arquimedes*)

O método usado por *Arquimedes* é chamado de exaustão e consiste em calcular uma aproximação do valor de pi por meio de polígonos com perímetros muito parecidos com o de uma circunferência. Primeiramente, constroi-se um quadrado inscrito de perímetro P e outro circunscrito de perímetro p , ambos em uma circunferência de raio r . Sabendo que, ao dividir o perímetro C da circunferência por seu diâmetro $2r$, obtém-se o valor de pi, ele pôde construir a seguinte desigualdade:

$$\frac{p}{2r} < \frac{C}{2r} < \frac{P}{2r}$$

$$\frac{p}{2r} < \pi < \frac{P}{2r}$$

Dessa forma, pi é um número encontrado entre os perímetros do quadrado menor e do quadrado maior, ambos divididos pelo diâmetro. Veja a *Figura 1*.

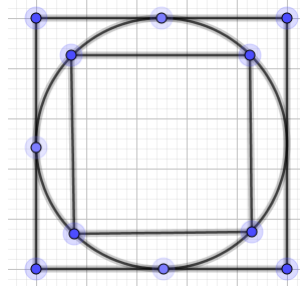


Figura 1: Círculo inscrito e circunscrito

E esse processo se repete de forma que o número de lados do polígono vá aumentando e ele seja sempre regular, assim seu formato se aproximará de uma circunferência. Arquimedes fez esses cálculos e aproximações para um polígono com 96 lados onde esse polígono se assemelhou muito à um círculo. Na utilização da desigualdade anterior para aproximação de pi, nesse caso, as medidas dos perímetros encontrados para os polígonos foram:

$$\frac{p}{2r} = \frac{22}{7}$$

e

$$\frac{p}{2r} = \frac{223}{71}$$

Ou seja,

$$\frac{22}{7} < \pi < \frac{223}{71}$$

$$3,14285714 < \pi < 3,14084507$$

Hoje em dia com a utilização de computadores é possível saber mais de 4 quatrilhões de casas decimais de pi.

Sistema de coordenadas (*Descartes*)

O plano cartesiano é um objeto matemático plano e composto por duas retas numéricas perpendiculares, ou seja, retas que possuem apenas um ponto em comum, formando um ângulo de 90°. Esse ponto comum é conhecido como origem e é nele que é marcado o número zero de ambas as retas. O plano cartesiano recebeu esse nome por ter sido idealizado por René Descartes e é usado fundamentalmente para sistematizar técnicas de localização no plano.

As duas retas que dão origem ao plano cartesiano precisam ser retas numéricas, pois essa é a condição que tornará possível encontrar localizações de pontos quaisquer no plano. Essa localização é a base fundamental de muitos conhecimentos comuns no cotidiano, como distância entre pontos.

Uma reta numérica é uma reta comum em que foi estabelecida uma correspondência com os números reais. Desse modo, cada ponto da reta está ligado a um único número real e é esse fato que permite qualquer localização. Um número real qualquer terá apenas uma localização em toda a extensão infinita da reta. Veja a *Figura 2*



Figura 2: Reta numérica

O plano cartesiano é formado por duas dessas retas: Uma responsável pela coordenada horizontal e outra responsável pela coordenada vertical. É comum usar as letras x para a primeira e y para a segunda e os termos *coordenada x* e *coordenada y*.

No plano cartesiano, a reta vertical responsável pelas coordenadas y é chamada de *ordenada*, e a reta horizontal, responsável pelas coordenadas x , é chamada de *abscissa*. Veja a *Figura 3*



Figura 3: Plano cartesiano

Ângulos opostos pelo vértice são iguais (*Tales de Mileto*)

Um ângulo é a medida da abertura formada por duas semirretas de mesma origem. Sendo assim, o encontro entre duas retas forma quatro ângulos. Observando-os dois a dois, é possível concluir que: ou esses ângulos estão lado a lado e, por isso, são adjacentes; ou se opõem um ao outro e, por isso, são chamados de opostos pelo vértice. Veja na *fig 4* ângulos adjacentes.

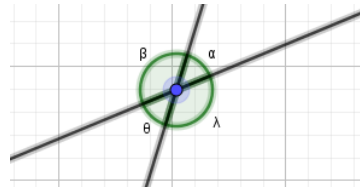


Figura 4: Ângulos adjacentes

Na imagem acima, os ângulos adjacentes são α e β , β e θ , θ e λ , α e λ ; os ângulos opostos pelo vértice são: α e θ e β e λ .

Propriedades

- Ângulos adjacentes são suplementares, ou seja, soma das medidas desses ângulos é igual a 180° . Veja *Figura 5*

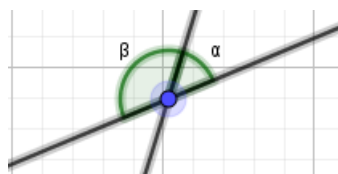


Figura 5: Ângulos suplementares

- Ângulos opostos pelo vértice são congruentes, ou seja, duas figuras são congruentes quando suas medidas são iguais. Observe a *Figura 6*.

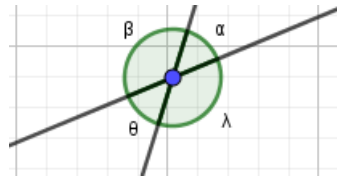


Figura 6: Ângulos opostos pelo vértice

Observe que os ângulos α e β são adjacentes, assim como θ e β . Além disso, observe que α e θ são opostos pelo vértice. Sendo assim, pode-se escrever as seguintes somas:

$$\begin{aligned}\alpha + \beta &= 180^\circ \\ \theta + \beta &= 180^\circ \\ 180^\circ &= 180^\circ \\ \alpha + \beta &= \theta + \beta \\ \alpha &= \theta + \beta - \beta \\ \alpha &= \theta\end{aligned}$$

Como α e θ são iguais, pode-se dizer que ângulos opostos pelo vértice possuem a mesma medida.

Essas descobertas apesar de sua importância são apenas uma gota no oceano que é a *Geometria*. Pela *Geometria* ser uma área muito extensa, ela foi subdividida em três subáreas da seguinte forma:

- *Geometria Analítica*: Relaciona a *Álgebra* e a *Análise* matemática com a *Geometria*;
- *Geometria Plana*: Também chamada de *Geometria Euclidiana*, estuda o plano baseando-se nos de Euclides;
- *Geometria Espacial*: Realiza o estudo de figuras tridimensionais. Nessa área de estudo, é possível calcular o volume de um sólido geométrico.

A *Geometria* está presente em nosso cotidiano nas mais diversas formas e seu ensino, portanto, é fundamental. É sabido que por fatores diversos que a *Geometria* não é um assunto amplamente estudado na formação do profissional das séries iniciais, sendo que a especialização do profissional tende a vir por cursos de pós-graduação ou até mesmo experiência do dia a dia. Segundo (DUMONT, 2008, resumo44)

a aprendizagem da geometria nas séries iniciais acontece com maior dinâmica tendo como suporte a observação e a manipulação de objetos. Outro ponto a ser destacado é que o conhecimento profissional dos professores em Geometria é oriundo das práticas pedagógicas desenvolvidas no dia-a-dia e, que mesmo com pouca experiência nessa área da matemática, os docentes, tendo o livro didático como suporte didático principal, ousaram a implementar práticas mais inovadoras em geometria.

Sendo assim, apesar de o aplicativo *Solid Planning* ter foco no ensino *Fundamental II* e *Ensino Médio*, nada impede a utilização moderada à alunos das séries iniciais.

2.2 O auxílio de um *Software* de acordo com os Parâmetros Curriculares Nacionais e as novas tecnologias

Os conteúdos relativos à *Geometria Clássica* continuam presentes, mas há uma ênfase na *Geometria das Transformações*, desde as séries iniciais até as finais do *Ensino Fundamental*. Alguns conteúdos passam a serem tratados já nas séries iniciais (plano cartesiano, simetria e semelhança, por exemplo, entram a partir do quinto ano). Além disso, a Base Nacional sugere o desenvolvimento de habilidades como “identificar movimentações de pessoas e objetos no espaço e suas representações no plano”, algo que não aparecia nos PCNs.

O uso de um aplicativo por exemplo para construção de gráficos fará a ligação entre concreto e teórico para o aluno, de forma dinâmica e autônoma. Bona (2009) destaca que um *Software* poderá permitir ao aluno desenvolver a capacidade de construir, de forma autônoma, o conhecimento sobre um determinado assunto quando seu embasamento estiver fundamentado em uma teoria de aprendizagem cientificamente comprovada. Neste contexto é relevante a colocação de Bonilla (1995, p. 68):

... para que um *Software* promova realmente a aprendizagem deve estar integrado ao currículo e às atividades de sala de aula, estar relacionado àquilo que o aluno já sabe e ser bem explorado pelo professor. O computador não atua diretamente sobre os processos de aprendizagem, mas apenas fornece ao aluno um ambiente simbólico onde este pode raciocinar ou elaborar conceitos e estruturas mentais, derivando novas descobertas daquilo que já sabia. Isto evidencia que é possível trabalhar numa linha construtivista de aprendizagem utilizando recursos da tecnologia de informação.

Esse ambiente informatizado torna-se ideal para nossos estudos, pois os objetos matemáticos passam a ter representações mutáveis, distintos dos tradicionais ambientes "lápis e papel" ou "giz e quadro-negro". De acordo com Damasco Neto (2010, p.69):

Tal dinamismo é permitido através da manipulação direta sobre os objetos presentes na tela do computador. Por exemplo: em geometria os elementos de um desenho são manipuláveis (o centro e o raio de uma circunferência, a reta e os pontos pelos quais ela for definida); no estudo de funções de primeiro grau as suas respectivas representações gráficas são objetos manipuláveis permitindo descrever a relação de crescimento/decrescimento entre os coeficientes e suas respectivas representações algébricas.

Assim, segundo visto acima percebe-se que como uma grande aliada em sala de aula tem-se a tecnologia. A tecnologia vem ganhando cada vez mais força, mas é importante que com ela venha um planejamento prévio do professor visando a plena interpretação, experimentação e, por fim, demonstração ao aluno.

3 Planificação

A planificação de um sólido geométrico é a apresentação de todas as formas que constituem sua superfície em um plano, ou seja, é uma forma de representar um objeto tridimensional em apenas duas dimensões. Essas planificações são usadas de várias maneiras, como por exemplo para calcular a área da superfície de um sólido. Para tanto, basta construir cada superfície externa do sólido do modo que essa figura seria no plano, respeitando suas medidas. Mas atenção, quando essa superfície é plana e poligonal, ela é chamada de face; quando ela é curva, é preciso imaginar como seria se ela fosse sobreposta com toda sua superfície tocando no plano. A superfície curva do cilindro, por exemplo, pode ser compreendida como um paralelogramo que foi enrolado. Confira as planificações dos sólidos geométricos mais conhecidos.

3.1 Planificação de um prisma

Prisma é um sólido geométrico definido no espaço tridimensional. Para sua definição, são necessários um plano, um polígono paralelo a esse plano e uma reta r concorrente a ele. O conjunto de segmentos de reta paralelos a r que tem como extremidades o polígono e o plano forma o sólido que conhecemos como prisma. Veja na *Figura7*:

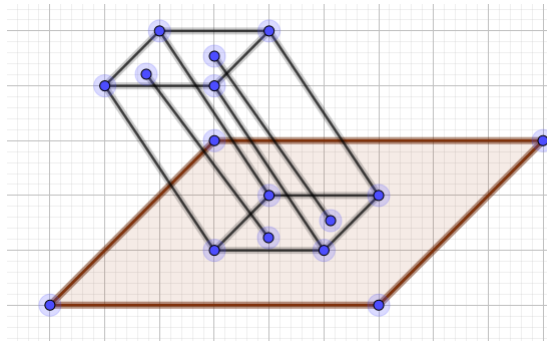


Figura 7: Prisma

Observe a *Figura8* a seguir, na qual são destacados os elementos de um prisma. Observe que os polígonos que formam as bases são IJKL e EFGH.

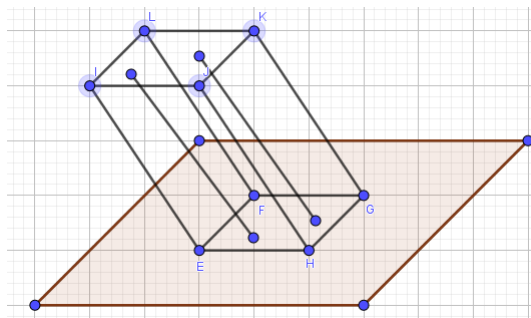


Figura 8: Elementos do prisma

O prisma possui uma de suas bases pertencentes ao plano, sendo a outra base a face paralela à ela. Com relação às faces laterais, são os polígonos que não são bases. Um exemplo na imagem acima é o polígono IJHE. Note que as faces laterais de um prisma sempre são quadriláteros. Note também que, em razão de os segmentos de reta que partem de IJKL até o plano serem paralelos e pelo fato de o próprio polígono ser paralelo ao plano, as faces laterais do prisma são paralelogramos.

Um prisma pode ser classificado quanto ao número de lados de suas bases. Assim, se a base de um prisma for um triângulo, ele será chamado de *prisma triangular*. Se a sua base for um quadrilátero, ele será chamado de *prisma quadrangular* e assim por diante.

O número de paralelogramos presentes na planificação do prisma é igual ao número de lados de uma de suas bases. Além disso, na planificação, aparecerão dois polígonos congruentes, que são as bases. A *Figura 9* a seguir mostra a planificação de um prisma de base retangular:

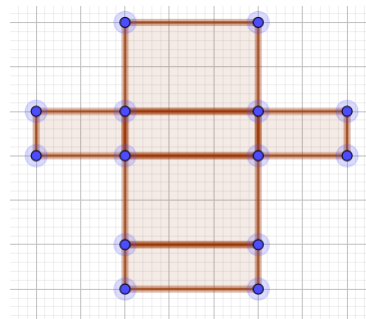


Figura 9: Planificação de um prisma

Como o número de paralelogramos é igual ao número de lados da base do prisma, um prisma de base pentagonal possui cinco paralelogramos em sua planificação.

Quanto à área total de um prisma, ela é dada pela soma da área das bases com a soma da área lateral. A fórmula que pode ser usada para determinar a área total do prisma é:

$$A_T = A_b + A_l$$

Onde A_T = Área total; A_b = Área da base; A_l = Área lateral

O princípio de Cavalieri

O *princípio de Cavalieri* hoje em dia é tido como postulado e é usado para determinar fórmulas para o cálculo de volume de sólidos geométricos. Por meio dele, é possível chegar ao volume de qualquer prisma utilizando o volume de um prisma conhecido, desde que o segundo possua a mesma altura que o primeiro e que ambos possuam áreas da base congruentes.

Veja o que acontece com dois prismas distintos que possuem o mesmo volume quando um deles é deformado. Primeiramente, coloca-se os dois prismas de mesmo volume sobre um mesmo plano α . Veja *Figura 10*.

Os dois prismas acima foram colocados sobre o plano e possuem área da base e altura congruentes. Pode-se dizer que os prismas são congruentes porque possuem

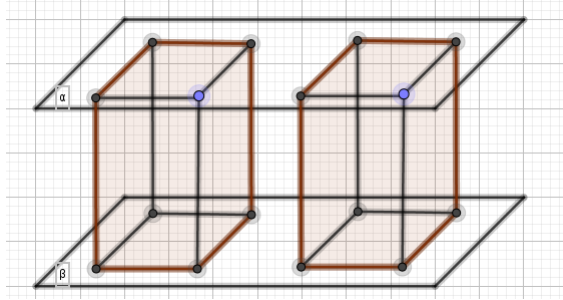


Figura 10: Volumes *Cavalieri*

medidas iguais e também que são equivalentes porque possuem volumes iguais. *Cavalieri* observou que, deformando um dos dois prismas sem modificar o formato de suas bases ou sua altura, eles continuam com volumes iguais. Veja *Figura11*

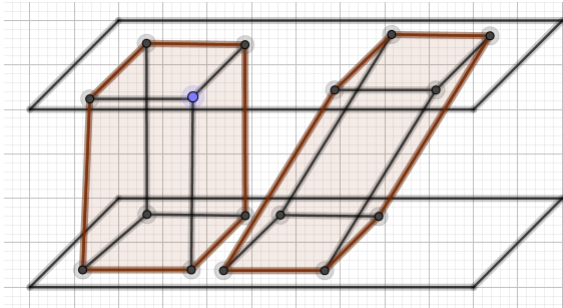


Figura 11: Princípio de *Cavalieri*

Na *Figura11*, note que o segundo prisma foi deformado, como se sua base estivesse fixa ao plano e seu topo tivesse sido empurrado para a direita. Isso não modificou o formato de sua base, que permanece quadrada e congruente à do outro prisma, nem sua altura.

Dessa maneira, *Cavalieri* propôs que, independentemente da altura em que esse corte é feito, o formato da figura obtida no segundo prisma é igual ao da primeira e elas são congruentes. Dessa maneira, como os dois prismas possuem a mesma altura, continuam equivalentes (com volumes iguais).

Daí segue que o volume de um prisma (reto ou oblíquo) é o produto da área da base pela altura. Em outras palavras

$$V = A_b \cdot h$$

V = Volume do prisma; A_b = área de sua base; e h = altura.

Formalmente dados dois sólidos geométricos A e B de mesma altura e áreas das bases, que, por sua vez, estão contidas no mesmo plano α , os sólidos A e B têm o mesmo volume se qualquer plano β , paralelo a α , determinar duas secções transversais com áreas iguais.

Dessa maneira, o *princípio de Cavalieri* pode ser usado também para sólidos completamente diferentes, mas que possuem mesma altura, bases com áreas iguais e que qualquer corte realizado nos dois por um mesmo plano resulte em figuras com áreas

iguais.

3.2 Planificação de um cilindro

O cilindro é considerado um sólido geométrico por ser uma figura geométrica espacial. Por terem uma superfície curva são chamados corpos redondos. A definição de um cone pode ser dada como dados dois planos paralelos α e β , um círculo C no plano α e uma reta r secante a esses planos, um cilindro é o conjunto de segmentos paralelos a r que possuem como extremidade o círculo C no plano α e algum ponto do plano β .

O raio do cilindro é definido como raio do círculo C , e a altura do cilindro é definida como a distância entre os planos α e β . A imagem a seguir mostra alguns dos segmentos que fazem parte da definição do cilindro.

Um cilindro é uma figura geométrica espacial, ou seja, só pode ser definida no espaço tridimensional. Sua definição formal é a seguinte: dados dois planos paralelos α e β , um círculo C no plano α e uma reta r secante a esses planos, um cilindro é o conjunto de segmentos paralelos a r que possuem como extremidade o círculo C no plano α e algum ponto do plano β .

O raio do cilindro é definido como raio do círculo C , e a altura do cilindro é definida como a distância entre os planos α e β . A *Figura12* a seguir mostra um exemplo de um cilindro.

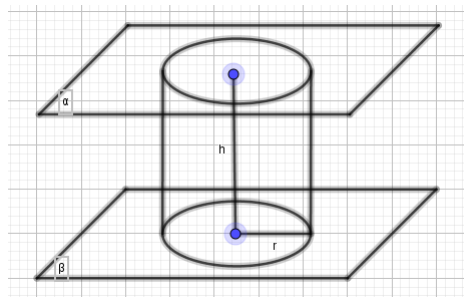


Figura 12: Cilindro

Como todas as secções transversais de um cilindro são círculos congruentes à sua base e como toda as secções transversais de um prisma são polígonos congruentes às suas bases, pelo princípio de Cavalieri, um cilindro tem volume igual a um prisma, desde que ambos tenham bases em um mesmo plano, alturas iguais e a área da base do cilindro seja igual à área da base do prisma. Assim, o volume do cilindro pode ser dado pelo produto da área de sua base por sua altura.

$$V = A_b \cdot h$$

Observando que a base de um cilindro é um círculo, podemos reescrever essa fórmula da seguinte maneira:

$$V = \pi \cdot r^2 \cdot h$$

A *Figura13* a seguir mostra a planificação de um cilindro.

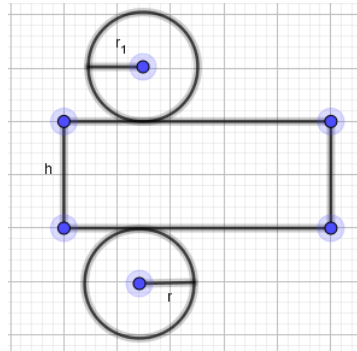


Figura 13: Planificação de um cilindro

Quanto à área total de um cilindro, ela é dada pela soma da área das bases com a soma da área lateral. Veja que a área lateral é formada por um retângulo de altura h e base $2\pi.r$. A fórmula que pode ser usada para determinar a área total do cilindro é:

$$A_T = A_b + A_l$$

Onde $A_T = \text{Área total}$; $A_b = \text{Área da base}$; $A_l = \text{Área lateral}$

3.3 Planificação de uma pirâmide

Pirâmides são poliedros construídos a partir de uma base poligonal e um ponto fora do plano onde se encontra essa base. São tridimensionais e, por isso, elas só podem ser definidas em um espaço que possui três ou mais dimensões. A definição formal das pirâmides é um conjunto de segmentos de reta cujas extremidades são um polígono e um ponto fora do plano que contém esse polígono. Veja na *Figura14*:

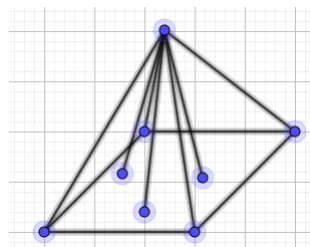


Figura 14: Pirâmide

As pirâmides podem ser classificadas de acordo com o seu número de faces. Note que esse número sempre é igual ao número de lados da base somado a uma unidade. Note também que, exceto pela base da pirâmide, todas as faces são triangulares. Pirâmide triangular: possui um triângulo como base; Pirâmide quadrangular: possui um quadrilátero como base; Pirâmide pentagonal: possui um pentágono como

base. E assim segue a classificação, que depende do número de arestas da base da pirâmide.

Observe que o número de triângulos sempre será igual ao número de lados do polígono da base. A planificação de uma pirâmide pentagonal, por exemplo, é composta por cinco triângulos e por um pentágono, como mostra a *Figura15* a seguir:

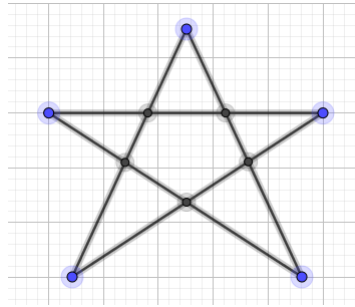


Figura 15: Pirâmide planificada

Dito isso, a planificação de uma pirâmide de base triangular é composta por quatro triângulos: uma da base e três das faces laterais; A planificação de uma pirâmide cuja base é um quadrilátero é composta por um quadrilátero e quatro triângulos, que também não são necessariamente congruentes.

Quanto à área total da pirâmide, ela é dada pela soma da área da base com a área lateral dela. Para encontrar a área lateral dessa figura, é preciso planificar sua superfície lateral. Essa planificação será um triângulo. Após a soma, a fórmula que pode ser usada para determinar a área total da pirâmide é:

$$A_T = A_b + A_l$$

Onde A_T = Área total; A_b = Área da base; A_l = Área lateral. Lembre-se de que A_b dependerá do polígono da base.

O volume da pirâmide é igual a um terço do volume do prisma que a contém. Portanto, a fórmula que pode ser usada para determinar o volume da pirâmide é:

$$V = \frac{A_b \cdot h}{3}$$

3.4 Planificação de um cone

O cone é considerado um sólido geométrico por ser uma figura geométrica espacial. Por terem uma superfície curva são chamados corpos redondos. A definição de um cone pode ser dada como sendo um círculo de centro C e raio r , pertencente a um plano α , e um ponto V fora do plano α , o cone é o conjunto dos segmentos de reta cujas extremidades são o círculo e o ponto V . A figura formada por essa construção é a *Figura16*:

É possível classificar um cone como considerando um segmento VC , que liga o centro C da base do cone ao seu vértice V , um cone pode ser classificado de duas maneiras:

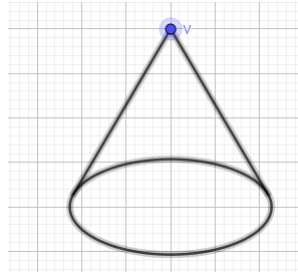


Figura 16: Cone

Cone reto, se o segmento VC é perpendicular à base do cone; e Cone oblíquo, se o segmento VC não é perpendicular à base do cone.

Um cone reto pode ser obtido por meio do giro de um triângulo retângulo, tendo como eixo de rotação um de seus catetos. Por esse motivo, esse cone também é denominado cone de revolução. A *Figura17* mostra um esquema com esse cone e o triângulo retângulo que o gerou.

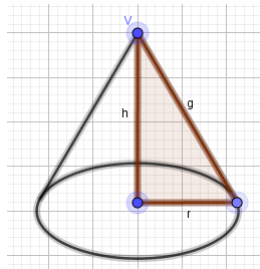


Figura 17: Revolução

Sabendo que as geratrizes do cone de revolução são congruentes, é possível encontrar a medida de uma delas por meio do *teorema de Pitágoras*:

$$h^2 + r^2 = g^2$$

Quanto à área total do cone é dada pela soma da área da base com a área lateral dele. Para encontrar a área lateral dessa figura, é preciso planificar sua superfície lateral. Essa planificação será um setor circular. Após a soma, a fórmula que pode ser usada para determinar a área total do cone é:

$$A = \pi.r.(g + r)$$

O volume do cone é igual a um terço do volume do cilindro que o contém. Portanto, a fórmula que pode ser usada para determinar o volume do cone é:

$$V = \frac{\pi.r^2.h}{3}$$

A planificação do cone apresenta um setor circular e um círculo, como mostra a *Figura18* a seguir:

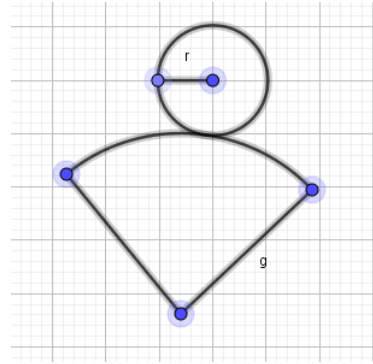


Figura 18: Planificação Cone

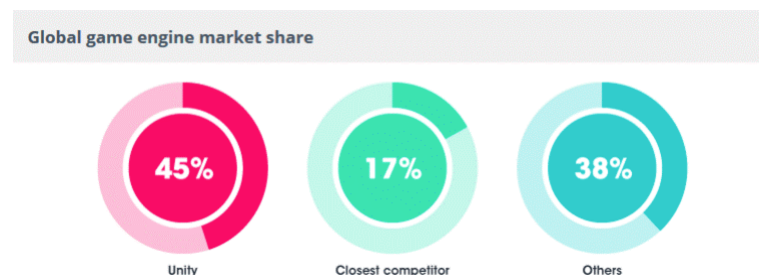
4 O Software Unity

O *Unity*, também conhecido como *Unity3D* foi criado pela *Unity Technologies*. O *Unity* é considerado um *motor de jogos* ou muito conhecido pelo termo em inglês *game engine*. Um *motor de jogo* é um programa de computador e/ou conjunto de bibliotecas para *videogames* e/ou computadores rodando sistemas operacionais. O termo *game engine* originou-se no meio da década de 1990, especialmente em conexão com jogos 3D.

4.1 A Utilização do Unity

Segundo o site de notícias de tecnologia *The Next Web* o *Unity* domina 45% do mercado global de *videogames*. Veja o gráfico da utilização do *Unity* na *Figura19*. Veja a matéria completa pelo link:

<https://thenextweb.com/gaming/2016/03/24/engine-dominating-gaming-industry-right-now/>

Figura 19: Gráfico de utilização do *Unity*

Segundo a própria empresa desenvolvedora da ferramenta, 34% dos 1.000 maiores jogos para celulares, conhecidos como *mobile*, são feitos com ela. Um dos motivos para a *engine* ser uma das primeiras opções lembradas por desenvolvedores de jogos é a sua versão gratuita, que permite criar um game do zero sem custo, e é justamente por esse motivo, pela facilidade de acesso até mesmo para os alunos, que aqui usou-se o *Unity*. Para instalar o *Unity*, bem como ter acesso ao tutorial contendo instruções de uso, características principais e exemplos pode-se inicialmente acessar o link:

<https://www.unity3d.com>

Caso necessite de ajuda, um tutorial pode ser acessado pelo link:

<https://www.youtube.com/watch?v=FMVwz3WJNpw>

Veja a tela de *download* na *Figura20*.

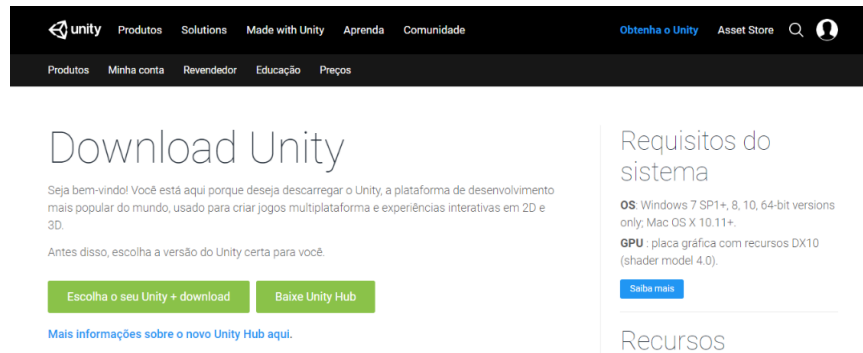


Figura 20: Tela de *download* do *Unity*

Serão abaixo descritos alguns recursos específicos necessários para a aplicação das atividades desenvolvidas.

4.2 *Microsoft Visual Studio*

O *Microsoft Visual Studio* é um ambiente de desenvolvimento integrado da *Microsoft* para desenvolvimento de *Software* especialmente dedicado às linguagens *Visual Basic (VB)*, *C*, *C++*, *C# (CSharp)* e *F# (FSharp)*. O *Unity* apresenta basicamente duas opções de linguagens para programar os movimentos : as linguagens *CSharp* e *JavaScript*. Elas possibilitam um mesmo fim apesar de cada linguagem ter seu diferencial no processo. Este trabalho foi inteiramente construído em *CSharp* e desenvolvido no *Microsoft Visual Studio*, mas é importante salientar que um trabalho como este pode ser construído de inúmeras outras formas com inúmeros outros *Softwares*, e é por isso que deixa-se claro que a utilização do desenvolvimento no ambiente *Microsoft Visual Studio* com a linguagem *C#* foi uma opção do autor desta dissertação. O *Microsoft Visual Studio* é uma plataforma gratuita e para baixa-la basta acessar

<https://visualstudio.microsoft.com/pt-br/vs/?rr=https%3A%2F%2Fwww.google.com%2F>

Caso necessite de ajuda, um tutorial pode ser acessado pelo link:

<https://www.youtube.com/watch?v=HLdTxMCOtIQ>

4.3 *Layout do Unity*

No canto superior direito da tela do *Unity* existe o ícone *Layout*. Usou-se aqui o *Layout 2 by 3*. Com esse *Layout* ativo foram ativadas as abas *Scene* e *Game* do lado esquerdo, e *Hierarchy*, *Project* e *Inspector* do lado direito da tela. Note que na área *Scene* é possível visualizar todos os objetos criados e suas respectivas modificações, enquanto na área *Game* é possível ver como o projeto ficará após apertar o *Play*. Por outro Lado, toda parte escrita e cronológica é feita no lado direito da tela onde a aba *Hierarchy* apresenta todos os itens criados que serão apresentados na aba *Scene*, enquanto a aba *Project* apresenta na forma de pastas e subpastas todos os itens que serão usados no projeto, inclusive itens não mostrados na tela *Scene*. E por fim a aba *Inspector* é fundamental para visualização e modificação referentes à localização e demais itens inseridos através de *Scripts* ou Seleção. Veja o *Layout* na *Figura21*.

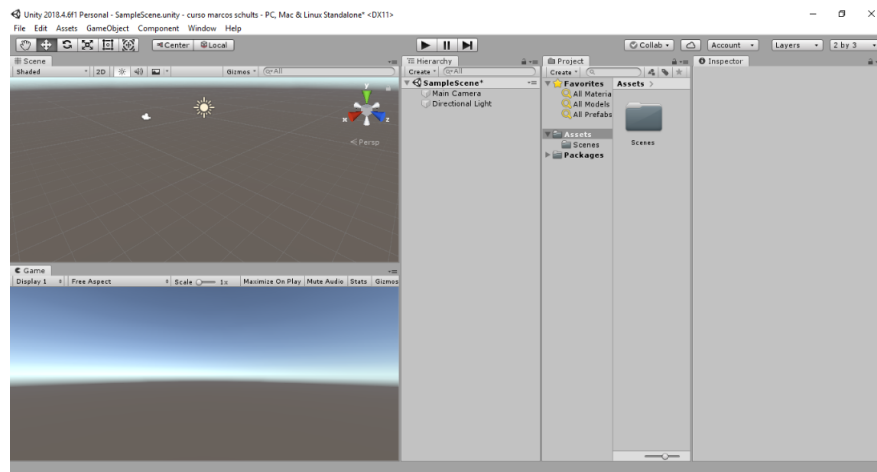


Figura 21: *Layout Unity* vazio

A *Barra de Menus* apresenta comandos que possibilitam a criação e/ou alteração de toda interface do *Unity*, são elas *File*, *Edit*, *Assets*, *GameObject*, *Component*, *Window* e *Help*. Enquanto a *Barra de ferramentas* é segmentada em alguns ícones que possibilitam a alteração dos objetos visualizados na aba *Scene*, onde as principais são *Transladar*, *Rotacionar* e *Ampliar*. Ao clicar em qualquer objeto é possível configurar/criar seus movimentos por meio de *scripts* que serão feitos por conveniência no *Visual Studio*. Abaixo, um *Cubo* foi criado para apresentação do *Software*. O caminho utilizado foi:

GameObject > *Object3D* > *Cube*

Veja o cubo na *figura22*.

Perceba que na aba *Inspector* é criado automaticamente alguns ícones, tais como *Transform* que mostra a localização especial, *Cube* que mostra seu formato, *Mesh Renderer* que possibilita a visualização do objeto na aba *Scene* e *Box Colider* que faz com que o objeto colida com outros. Na aba *Inspector* é possível adicionar incontáveis outros ícones que serão apresentados durante a criação do aplicativo proposto.

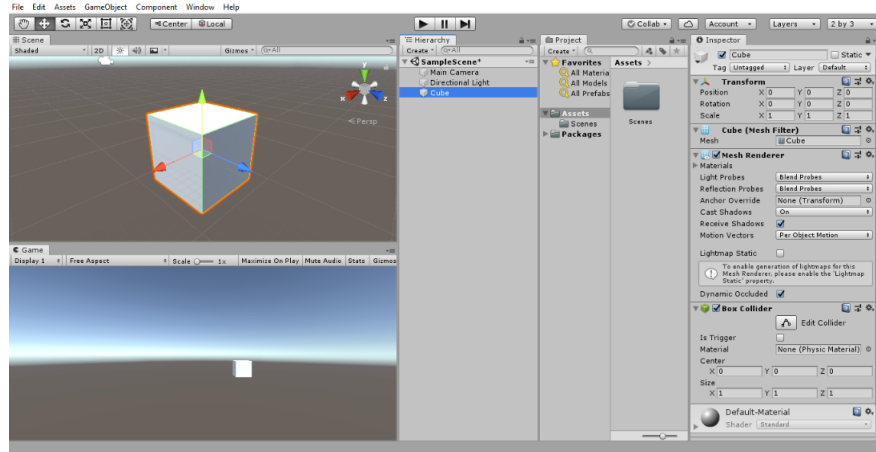


Figura 22: *Layout Unity* com objeto

5 Criação do Aplicativo *Solid Planning*

Neste capítulo, são apresentadas sugestões de atividades que aproximam os alunos da *engine Unity* que por meio de conceitos matemáticos será criado o aplicativo. As atividades têm o propósito de fazer com que os alunos sejam capazes de efetuar construções de figuras geométricas planas e espaciais e, com isso, possam apropriar-se de conhecimentos que os auxiliem no cálculo de área total e volume de sólidos, bem como melhorar a percepção espacial. Para tal criação é necessário que os alunos tenham conhecimento prévio sobre plano cartesiano e conjuntos numéricos, e conforme o trabalho for se desenvolvendo o aluno aprenderá indiretamente o básico e necessário da linguagem *C#* na criação. Caso algum desses conceitos não seja de conhecimento dos alunos, o professor deve fazer as adaptações necessárias ou até mesmo uma introdução de tais conceitos, sempre respeitando o nível de aprendizagem dos alunos. Algo importante a ser citado aqui é que será feita detalhadamente a criação da planificação de um cubo que por sua vez será análoga à planificação de vários outros sólidos.

5.1 A criação das faces do cubo

Orientações

- Número de aulas: 1
- Público alvo: Alunos de ensino fundamental II e alunos de ensino médio.
- Material Utilizado: Um computador com a *Unity* instalada.

Objetivos

- Identificar um plano;
- Compreender os comandos básicos do *Unity*;
- Criar uma face quadrada;
- Multiplicar a face já criada;

Orientações ao professor

Nesta atividade o aluno reconhece como funciona um plano em um eixo de três di-

mensões, e como é o deslocamento e modificação de um objeto neste eixo. Abaixo são enumerados alguns comandos básicos que podem ser verificados na *Figura23*:

- 1 - Movimenta a cena;
- 2 - Movimenta o objeto selecionado;
- 3 - Rotaciona o objeto selecionado;
- 4 - Amplia/Reduz o objeto selecionado;
- 5 - Movimenta/Amplia/Reduz o objeto selecionado;
- 6 - Rotaciona/Movimenta/Amplia/Reduz o objeto selecionado;
- 7 - São apresentadas todas as transformações citadas nos itens anteriores;

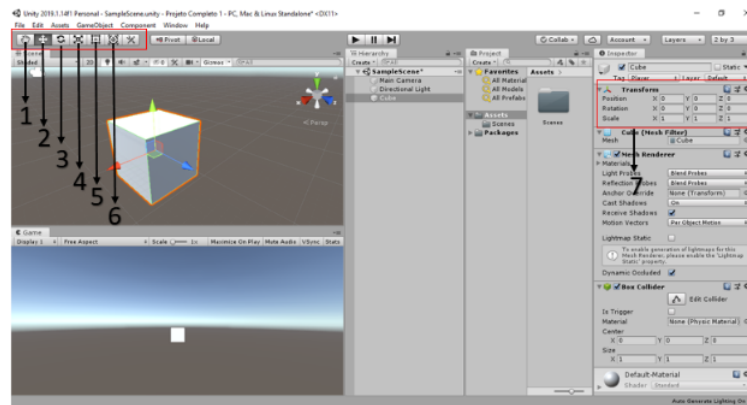


Figura 23: Ferramentas de alteração de objetos

Roteiro

Primeiramente crie um cubo seguindo o caminho:

GameObject > 3D Object > Cube

Veja o cubo na *Figura24*.

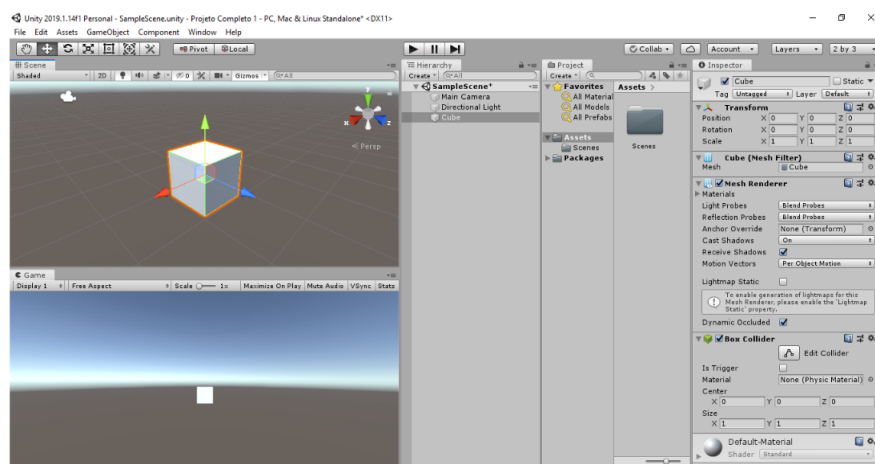


Figura 24: Cubo sólido

Agora reduza a *Scale y* a 0,001. Veja que o *Cubo* se tornou praticamente um plano, pois seu comprimento no *eixo y* é tão pequeno que apenas os comprimentos

nos outros dois eixos ficaram visíveis.

Repita o processo de forma que se tenha pares de faces paralelas. Renomeie cada face como *Face1*, *Face2* e assim sucessivamente até a *Face6*, veja que elas estarão duas a duas sobrepostas. Para que elas saiam da sobreposição e fique cada uma em um local adequado de forma que sejam faces do cubo é preciso alterar a posição de cada uma, para isso, nas duas faces que possuem a escala em X igual a 0,001 altere a posição em X, uma para 0,5 e a outra para -0,5, da mesma forma nas duas faces que possuem escala em Y igual a 0,001 altere a posição em Y uma para 0,5 e a outra para -0,5. Repita o processo em Z. Veja o resultado na *Figura25*.

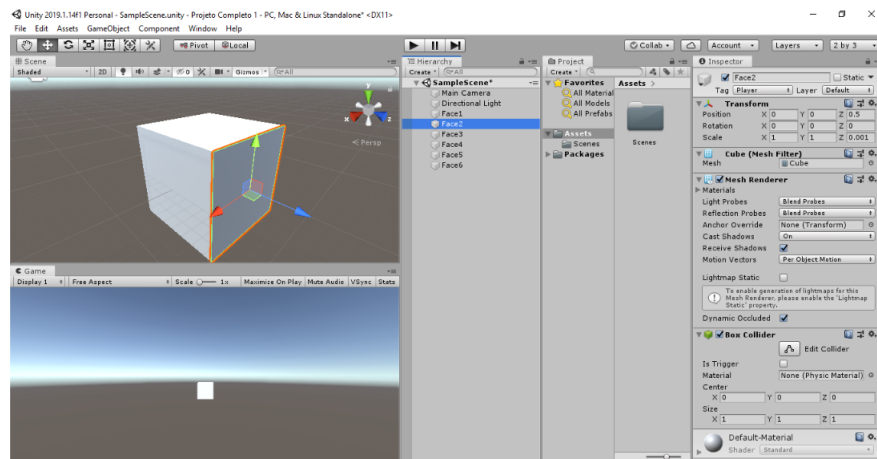


Figura 25: Cubo formado por 6 objetos

Veja que será formado um cubo perfeito. Note que cada *Face* representa um objeto distinto no *Unity*

5.2 Programando a abertura da *Face1*

Orientações

- Número de aulas: 4 aulas / 1 atividade.
- Público alvo: Alunos de ensino fundamental II e alunos de ensino médio.
- Material Utilizado: Computador com os Programas *Unity* e *Visual Studio* instalados.

Objetivos

- Relacionar a criação de pastas e subpastas a conjuntos;
- Aprender comandos básicos da linguagem C#;
- Compreender a relação de um *Script* e suas respectivas consequências na *Unity*;
- Criar a abertura da *Face1*;

Orientações ao professor

Na criação das pastas e subpastas reforce com o aluno os conceitos de conjuntos e subconjuntos. Segue abaixo a definição de conjuntos e subconjuntos.

Conjuntos e Subconjuntos

Para o ensino é fundamental o conhecimento da linguagem de conjuntos, uma vez que esta forma a base comum a todos os campos da Matemática atual. Este conhecimento é importante, mesmo para que se saiba adequar o grau de formalismo da linguagem de conjuntos a cada série da educação básica. Pode-se considerar que um conjunto é uma coleção de elementos e quando todos os elementos de um conjunto A qualquer pertencem a um outro conjunto B, diz-se, então, que A é um subconjunto de B, ou seja $A \subset B$. Uma definição de subconjunto seria: Sejam A e B conjuntos. Se todo elemento de A for também elemento de B, diz-se que A é um subconjunto de B, que A está contido em B, ou que A é parte de B. Para indicar este fato, usa-se a notação $A \subset B$. A relação de $A \subset B$ chama-se relação de inclusão. Quando A não é um subconjunto de B, escreve-se $A \not\subset B$. Isto significa que nem todo elemento de A pertence a B, ou seja, que existe pelo menos um objeto x tal que $x \in A$ e $x \notin B$. Dois conjuntos muito importantes são:

Conjunto Vazio: não possui elementos e pode ser representado por $\{ \}$ ou \emptyset .

Conjunto Universo: possui todos os elementos de acordo com o que se está trabalhando, pode ser representado pela letra maiúscula U.

Programando em C# no Visual Studio

Entrando no campo da programação; ao abrir o *Software Visual Studio* pelo *Unity*, já será aberta no *Visual Studio* uma estrutura pré-programada, então para uma melhor compreensão do leitor, na *Figura28* foram descritas algumas orientações em cor verde. Outro ponto é que como todo trabalho será baseado na programação C#, segue uma tabela para auxílio com variáveis em C# na *Figura26*.

Variáveis na linguagem C#

Tipo	Tamanho	Valores Possíveis
bool	1 byte	true e false
byte	1 byte	0 a 255
sbyte	1 byte	-128 a 127
short	2 bytes	-32768 a 32767
ushort	2 bytes	0 a 65535
int	4 bytes	-2147483648 a 2147483647
uint	4 bytes	0 to 4294967295
long	8 bytes	-9223372036854775808L to 9223372036854775807L
ulong	8 bytes	0 a 18446744073709551615
float	4 bytes	Números até 10 elevado a 38. Exemplo: 10.0f, 12.5f
double	8 bytes	Números até 10 elevado a 308. Exemplo: 10.0, 12.33
decimal	16 bytes	números com até 28 casas decimais. Exemplo 10.991m, 33.333m
char	2 bytes	Caracteres delimitados por aspas simples. Exemplo: 'a', 'ç', 'o'

Figura 26: Variáveis na linguagem C#

Para uma introdução à linguagem de programação C#, veja abaixo um exemplo de

como é usada a programação em um sistema bancário simples.

Suponha que em uma conta bancária, os interesses queiram armazenar o saldo de uma conta e o nome do correntista, esse é um caso onde o C# vai armazenar um dado em uma determinada região da memória, essa região na programação é chamada variável. E são vários os tipos de variáveis. Pode-se armazenar um número inteiro, um número real, um nome, e é por isso que deve-se primeiramente declará-la no texto do programa. Quando se declara uma variável, além de especificar seu tipo (texto ou inteiro, por exemplo) é preciso também nomeá-la, e esse nome vem logo após sua especificação, veja o exemplo:

```
int Limite;
```

Essa é uma variável de número inteiro que se chama *Limite*. Sempre é preciso terminar uma linha de códigos com “;” na linguagem C#.

Também existem situações como:

```
double saldo = 100.0;
double valorDoSaque = 10.0;
saldo = saldo - valorDoSaque;
```

Saiba que para saldo pode-se usar também:

```
saldo -= valorDoSaque;
```

Quando o compilador do C# encontra o saldo -= valorDoSaque, essa linha é traduzida para a forma vista anteriormente: saldo = saldo - valorDoSaque. Além do -=, tem-se também os operadores += (para somas, que alias, será usado para criar um cronometro neste trabalho), *= (para multiplicações) e /=(para divisões).

Esses mesmos conceitos usados acima serão usado durante este projeto.

Roteiro

Em *Project*, crie uma pasta com o nome *Scripts*. Crie também um *Script* com o nome *Face1*, em seguida arraste o *Script Face1* para dentro da pasta *Scripts*. Veja na *Figura27*.

Dê dois cliques no *script* que automaticamente abrirá o modo de edição no *Visual Studio*. Veja na *Figura28*.

Neste ponto o aluno deve compreender a relação de abrir e fechar chaves como um conjunto, ou seja, assimilar que uma chave que abre e fecha será um subconjunto de outras chaves caso hajam.

Iniciando o código tem-se:

```
Public class Face1: MonoBehaviour
```

Note que aqui tem-se *Public Class*, o que significa que *Face1* é uma classe pública e ela traz como herança *MonoBehaviour*, que faz com que esse *Script* reconheça

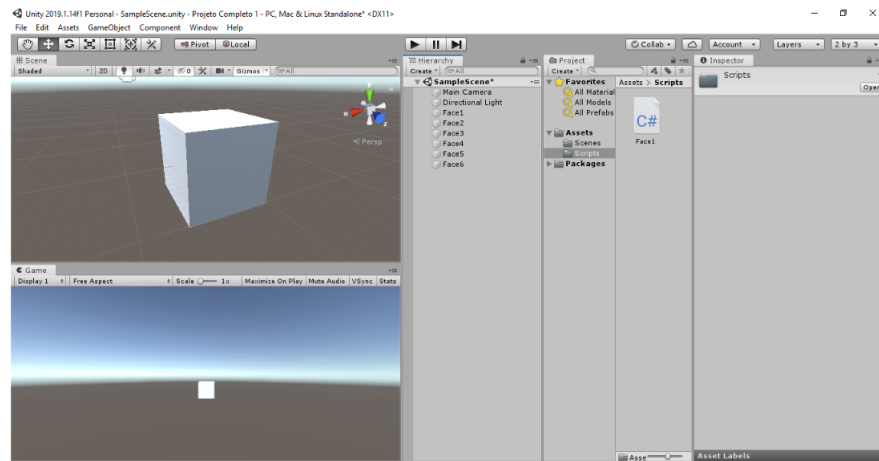


Figura 27: Pasta *Scripts*

```

1  //Essas serão as bibliotecas que serão básicas para edição.
2  // Ao abrir o Visual Studio pelo Unity elas já vêm automaticamente.
3  using System.Collections;
4  using System.Collections.Generic;
5  using UnityEngine;
6
7  //Esse será o projeto, veja que já abre com o nome do script
8  // referências
9  public class Facel : MonoBehaviour
10 {
11     // Todos os códigos digitados aqui serão verificados apenas no início
12     // referências
13     void Start()
14     {
15     }
16
17     // Todos os códigos digitados aqui serão verificados a cada frame
18     // referências
19     void Update()
20     {
21     }
22 }
23

```

Figura 28: Interface do *Visual Studio*

as estruturas deste código como, por exemplo, *void Update* que é um espaço da programação verificado a cada *frame* e *void Start* que é um espaço da programação verificado ao se apertar *Play*.

Agora, os alunos devem criar duas variáveis:

Ligado: que funcionará com Liga e Desliga

Ângulo: que dará a abertura em graus

Veja:

```
public class Face1 : MonoBehaviour
{
public bool Ligado;
private float Ângulo;
```

Como dito, pelo motivo da variável *Ligado* ser pública, ela aparecerá na aba *Inspector*.

Note que ao apertar *Play* na parte superior central da tela, apesar de não dar nenhum erro, aparentemente não irá acontecer nada, e esse motivo se justifica, pois, as variáveis foram criadas, mas ainda não foram descritas no código, ou seja, elas por enquanto apenas aparecem sem função nenhuma.

Agora, em *void Update*, serão especificados os movimentos da *Face1*.

O comando *transform.Rotate(x, y, z)* irá movimentar o objeto rotacionando. É preciso especificar a velocidade no eixo(x, y e/ou z) que se deseja, mas para que essa velocidade seja a mesma em qualquer computador é preciso usar também o comando *10*Time.deltaTime*.

De acordo com o local de cada *Face* ela irá rotacionar sobre um eixo específico, aqui a *Face1* irá rotacionar sobre o eixo x, assim o comando será:

```
void Update()
{
transform.Rotate(10*Time.deltaTime, 0, 0);
}
```

Feito isso esse comando irá rotacionar a *Face1* caso esse *Script* seja adicionado à seu *Inspector*, mas se isso fosse feito neste momento veja qual seria o resultado da rotação na *Figura29*.

Veja que é necessário que esse *Script* funcione como uma dobradiça de porta, portanto ele deve funcionar na parte inferior desta face, sendo assim cria-se um objeto vazio para ser colocado na base desta face, para tal:

GameObject>Create Empty

Clique com o botão direito e renomeie-o para *AberturaFace1*. Como a posição da *Face1* é (0, 0, 0.5) e a abertura precisa ser feita em sua base, a posição da *AberturaFace1* deverá estar na posição (0, -0.5, 0,5). Feito isso, Na aba *Inspector* do Objeto *AberturaFace1* clique em *Add componente* e digite o nome do *Script*: *Face1*. Pronto, o *Script* de abertura já está preparado para abrir a face, mas a face ainda não está conectada com a abertura, então neste momento, observe que

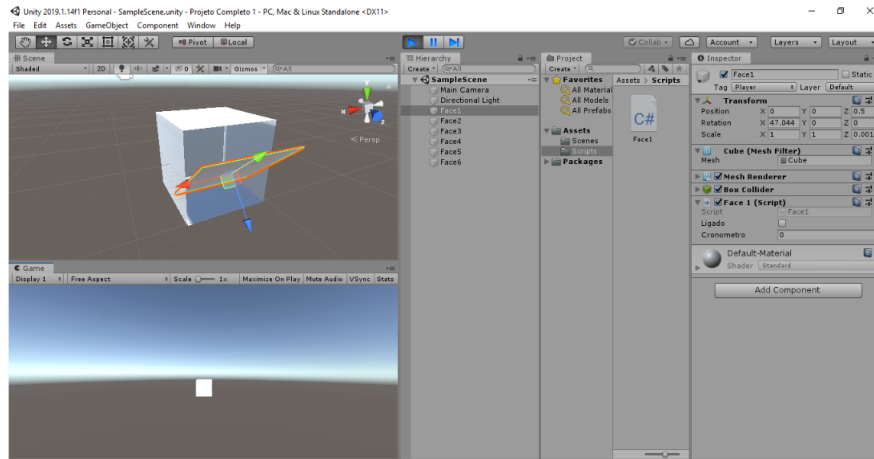


Figura 29: Rotação no centro da face

Face1 será um subconjunto de *AberturaFace1*, para isso, arraste o objeto *Face1* para *AberturaFace1*. Note na *Figura30* que a abertura da face agora estará acontecendo de forma correta.

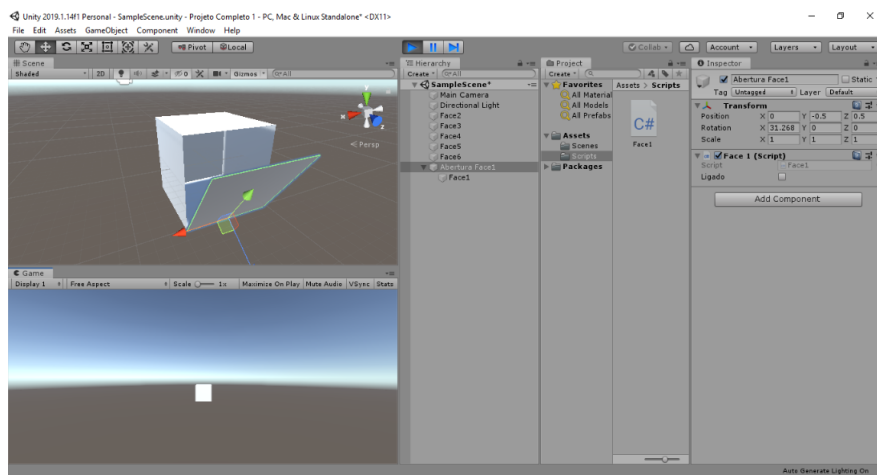


Figura 30: Abertura ilimitada de uma face

Neste momento, perceba que a face está abrindo e rotacionando ilimitadamente, e é justamente por isso que foram criadas as variáveis *Ligado* e *Ângulo*, as quais serão utilizadas na próxima seção

5.3 Limitando a abertura da *Face1* em 90°

Orientações

- Número de aulas: 4 aulas / 1 atividade.
- Público alvo: Alunos de ensino fundamental II e alunos de ensino médio.
- Material Utilizado: Computador com os Programas *Unity* e *Visual Studio* instalados.

Objetivos

- Relacionar a criação de pastas e subpastas a conjuntos;
- Aprender comandos básicos da linguagem C#;
- Compreender a relação de um *Script* e suas respectivas consequências no *Unity*;
- Criar um limite de abertura de 90°;

Orientações ao professor

Nesta seção, um comando muito utilizado na programação será apresentado, o comando *if/else*. Na criação das pastas reforce com o aluno os conceitos de conjuntos e subconjuntos.

if/else – Referência em C#

Uma instrução *if* identifica qual instrução executar com base no valor de uma expressão. No exemplo a seguir, a variável *bool condition* deve ser definida como *true* e, em seguida, verificada na instrução *if*.

```
If(Ligado == true)
{
Cronometro += Time.deltaTime;
}
else
{
Cronometro = 0
}
```

Em uma instrução *if-else*, se a condição for avaliada como verdadeira, a ação será executada. Se a condição for falsa, a ação não será executada. Após o comando *if* ser executado, o controle é transferido para a próxima instrução.

Para uma única instrução, as chaves são opcionais, mas recomendadas.

Roteiro

Com a estrutura estipulada neste trabalho o aluno deve aprender de forma gradual, sendo assim é fundamental que nenhum aluno comece a atividade desta seção antes de ter concluído as atividades anteriores. O objetivo aqui é estipular um limite para a abertura da face que será de 90°.

Neste momento a base será o comando *if-else* citado nas orientações ao professor. Primeiramente iguala-se a variável *Ângulo* à abertura de cada face, ou seja:

```
Void Update
{
Ângulo += 10*Time.deltaTime
}
```

Então é preciso mostrar que esse é o limite para a abertura da face, portanto será necessário que ela desligue ao chegar neste limite, assim será primordial que a face só se movimente se a variável *Ligado* estiver ativada, então:


```

Void Update
{
if( Ligado == true)
{
Ângulo += 10*Time.deltaTime
transform.Rotate(10*Time.deltaTime,0,0);
}
if(Ângulo >= 90)
{
Ligado = false;
}
}

```

E para finalizar esta abertura estipula-se um pequeno tempo para que a variável *Ligado* ative, para isso cria-se uma variável privada de nome *Ativar* e determina-se que quando esta variável alcançar 5 segundos após a ativação do botão *Play* ela ative a variável *Ligado*.

Veja:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Face1 : MonoBehaviour
{
public bool Ligado;
private float Ângulo;
private float Ativar;
void Start()
{
}
void Update()
{
Ativar += Time.deltaTime;
if( Ativar >= 5)
{
if (Ângulo < 90)
{
Ligado = true;
}
else
{
Ligado = false;
}
}
if (Ligado == true)
{
Ângulo += 10 * Time.deltaTime;
transform.Rotate(10 * Time.deltaTime, 0, 0);
}
}
}

```

}
 }
 }

Agora aperte o botão *Play* e veja o resultado. Observe a *Figura31*.

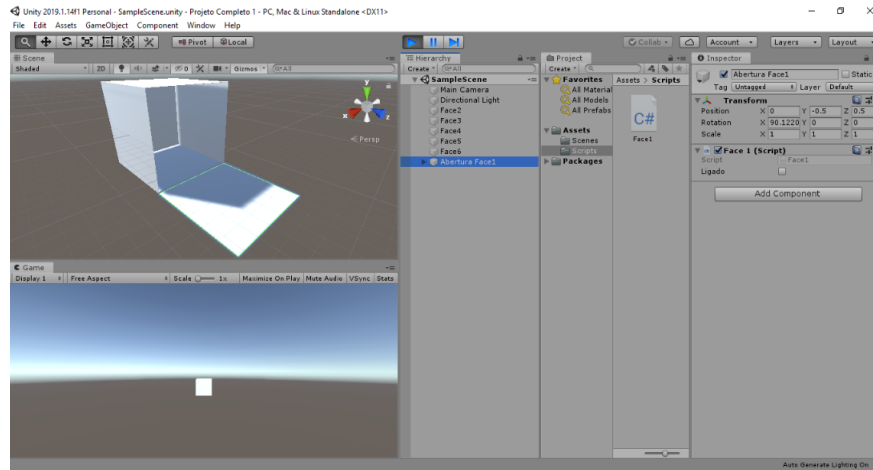


Figura 31: Abertura em 90° de uma face

O *Script* está completo, agora basta que seja adaptado à cada face.

5.4 Planificação finalizada

Orientações

- Número de aulas: 4 aulas / 1 atividade.
- Público alvo: Alunos de ensino fundamental II e alunos de ensino médio.
- Material Utilizado: Computador com os Programas *Unity* e *Visual Studio* instalados.

Objetivos

- Relacionar a criação de pastas e subpastas a Conjuntos;
- Aprender comandos básicos da linguagem C#;
- Compreender a relação de um *Script* e suas respectivas consequências no *Unity*;
- Criar um limite de abertura de 90° para todas as faces ;

Orientações ao professor

Na criação das pastas é importante reforçar com o aluno os conceitos de conjuntos e subconjuntos. Perceba que como em uma expressão matemática, caso alguma chave seja aberta mas não seja fechada ocorrerá um erro.

Roteiro

A atividade desta seção será uma continuação da seção anterior, sendo assim, é

fundamental que nenhum aluno comece esta atividade antes de ter concluído a atividade da seção anterior.

O objetivo aqui é transferir o *Script* criado na aula anterior para todas as outras faces, portanto, lembrando que esse *Script* funcionará como uma dobradiça, é necessário criar um objeto nulo e colocar cada “dobradiça”, ou seja, colocar cada *Script*. Foram nomeados, portanto, *AberturaFace1*, *AberturaFace2* e assim sucessivamente. Em seguida coloca-se cada face como respectivo subconjunto de cada abertura. Para que não seja um processo repetitivo e desnecessário, é possível copiar e colar cada *Script* para enfim, arrastar para sua respectiva abertura. Mas é fundamental que renomeie o nome dentro do *Script*, veja:

```
public class Face1 : MonoBehaviour para Face1
public class Face2 : MonoBehaviour para Face2
```

Duas observações importantes, a primeira é que a *Face6* será a base, assim, não será necessário nenhum *Script*. A segunda é que o topo, chamado aqui de *Face*, terá que ser uma subpasta de uma das faces, neste trabalho foi escolhida a *Face1*. Feito isso a planificação estará finalizada, aperte *Play* e veja o resultado. Aqui é possível ver na *Figura32*.

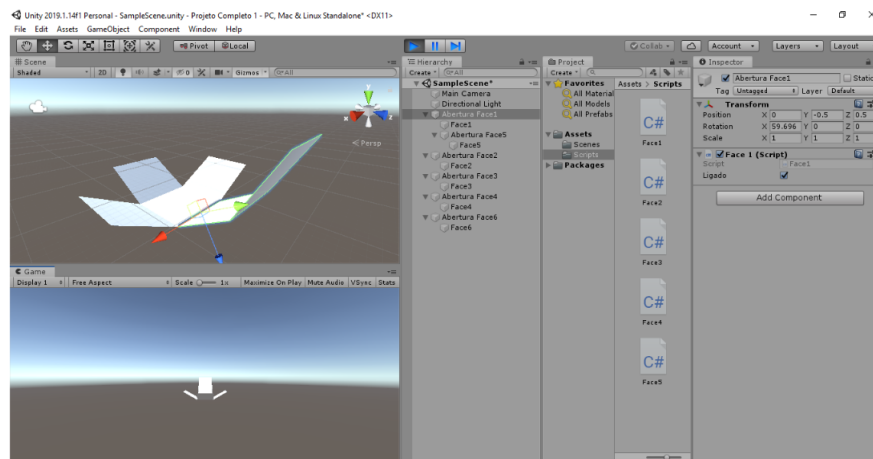


Figura 32: Planificação finalizada no *Unity*

5.5 Criando a realidade aumentada no computador com o auxílio do *Vuforia*

Orientações

- Número de aulas: 2 aulas / 1 atividade.
- Público alvo: Alunos de ensino fundamental II e alunos de ensino médio.
- Material Utilizado: Computador com os Programas *Unity* e *Visual Studio* instalados.

Objetivos

- Aprender a configurar o *Unity* para a utilização do *Vuforia*;
- Aprender a configurar o *Vuforia*;

- Criar o aplicativo *Solid Planning*;

Orientações ao professor

Primeiramente é necessário que o *Unity* esteja configurado para trabalhar com o *Vuforia*, para tal:

File>Build Settings>PC, Mac Linux>Player Settings

Então ative a caixa *Vuforia Augmented Reality Supported*

Em seguida:

File>Build Settings>Android>Player Settings

E ative a caixa *Vuforia Augmented Reality Supported*

Pronto, o *Unity* está configurado.

Caso necessite de ajuda, um tutorial pode ser acessado pelo link:

<https://www.youtube.com/watch?v=euXgmj04SMQ>

Roteiro

Na seção anterior foi concluído o trabalho de planificar um cubo na *Engine Unity*. Neste momento o foco será fazer a planificação da seção anterior ser ativada assim que a câmera do computador/*Smartphone* identificar a imagem de um cubo. Sendo assim é necessário se ter ou criar uma imagem de um cubo para que seja usada com este propósito. Neste trabalho usou-se por opção a imagem da *Figura33*.

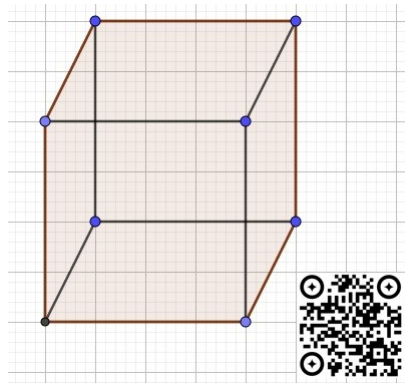


Figura 33: Imagem de ativação

Esta figura precisa ser registrada no site do *Vuforia* para que seja gerado um código que futuramente será transferido para o *Unity*, para isso acesse:

<https://developer.vuforia.com/vui/develop/licenses>

Após se registrar siga:

Develop>Get Development Key

Dê o nome de *Cubo* para a licença, aceite os termos e confirme.

Feito isso siga:

Target Manager > Add Database > Database Name(Cubo) > Device > Creat

Aparecerá o nome *Cubo*, clique nele e neste momento será adicionada a imagem de ativação para a realidade, ou seja, a *Figura33* já salva anteriormente no computador. É de extrema importância que tudo seja salvo onde se consiga encontrar com facilidade.

Para adicionar a imagem:

Add Target > Single Image > Browse... > Width(50) > Add

A imagem aparecerá na lista *Target Name*

Clique em:

Download Database > Unity Editor > Download

Feito isto, será gerado um código que é preciso ser levado para o *Unity*, para copia-lo siga:

License Manager > Cubo > License Key > Copiar o código

Na interface do programa *Unity* é preciso adicionar uma *AR Câmera*, isso significa acrescentar configurações que ativem a camera de realidade aumentada, para isso:

Game Object > Vuforia Engine > AR Câmera > Import

Na aba *Creat* clique uma vez sobre *AR Câmera*, em seguida na aba *Inspector* clique em *Open Vuforia Engine Configuration*. Em *App license Key* cole o código copiado do *Vuforia*.

Em seguida para que ela seja importada para a *Unity*, siga:

Game Object > Vuforia Engine > Image

Veja que a imagem aparecerá na aba *Scene* e também na aba *Create*. Enfim, crie um objeto nulo e renomeie-o para *Cubo* e arraste todas as aberturas criadas na seção anterior para dentro de *Cubo*, em seguida, a ordem de pastas e subpastas deve ser:

AR Câmera > ImageTarget > Cubo

Para finalizar é necessário que as imagens de gatilho salvas no site do *Vuforia* sejam impressas para que a câmera a reconheça e ative a planificação em realidade aumentada.

Para verificar o resultado basta utilizar um computador que tenha *webcam* e apertar *Play* com a imagem posicionada na frente da câmera. Veja o resultado do trabalho na *Figura34*.

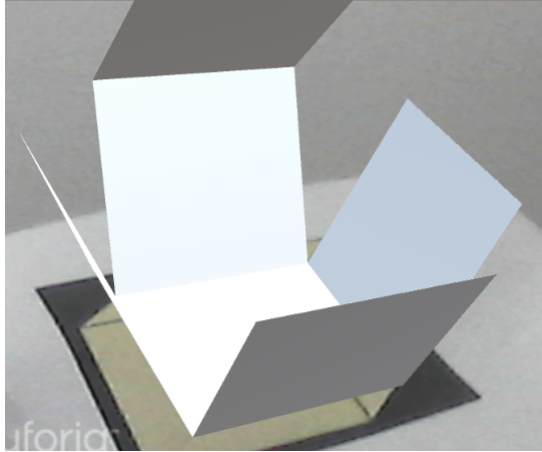


Figura 34: Realidade Aumentada - Planificação Cubo

5.6 Criando a *Fita de Mobius*

Orientações

- Número de aulas: 2 aulas / 1 atividade.
- Público alvo: Alunos de ensino fundamental II e alunos de ensino médio.
- Material Utilizado: Computador com o *Software Blender* instalado.

Objetivos

- Compreender a relação entre a *Geometria Espacial* e *Geometria Plana*;
- Compreender comandos básicos do *Software Blender*;
- Compreender a *Fita de Mobius*;

Orientações ao professor

Antes de mais nada, neste momento é preciso instalar o *Software Blender*. Para isso acesse o site oficial

www.blender.org

Caso necessite de ajuda, um tutorial pode ser acessado pelo link:

https://www.youtube.com/watch?v=NH1GZ_w5tD0list=PLxpJpQMFznCWhqBj2CfEfc2UFK_z-ao93index=2

O *Software Blender*

O *Blender* é um *Software* de criação em 3D. Gratuito, de código aberto e disponível para *Windows*, *Mac* e *Linux*, a ferramenta oferece funcionalidades completas para modelagem, animação, criação e visualização de conteúdo 3D interativo e pós-produção. Com o programa, o usuário cria desde projetos mais simples como modelos de maquetes e produtos até alguns mais complexos como jogos e animações. O programa implementa ferramentas semelhantes às de outros *Softwares*, que incluem:

- Dinâmica de corpo rígido (*rigid body dynamics*), dinâmica de fluido (*fluid dynamics*) e dinâmica de corpo macio (*soft body dynamics*);
- Avançadas ferramentas de modelagem;
- Ferramentas de animação de personagens;
- Composição de imagens e vídeos.

O *Blender* suporta vários idiomas e sua interface é totalmente personalizável, pois possibilita a distribuição do *Layout* reorganizando as ferramentas e componentes para que fiquem de acordo com a forma de trabalho do usuário.

Neste trabalho para construção de alguns sólidos será usado o *Blender*.

Roteiro

Neste momento é importante que o aluno faça a construção física de uma *Fita de Mobius* usando colagem de uma fita de papel para que a compreensão fique melhor, pois teoricamente o assunto ainda é muito avançado para as idades que contemplam o ensino médio e fundamental. Citaremos resumidamente que a fita de mobius é um espaço topológico obtido pela colagem de duas extremidades de uma fita após efetuar meia volta numa delas. O matemático e astrônomo alemão August Ferdinand Mobius estudou esse objeto em 1858, cuja importância prendia-se com a noção de orientabilidade, que não era ainda bem compreendida na época. Caso tenha interesse em se aprofundar no assunto existe um belo artigo que pode ser acessado pelo link:

<http://maraeducare.blogspot.com/2014/02/aplicaciones-de-la-increible-cinta-de.html>

Veja a *Fita de Mobius* na *Figura35*.



Figura 35: *Fita de Mobius*

Para a construção abra o *Software Blender*. Inicialmente será criado um retângulo e rotacionado em 90° , que ficará semelhante à *Figura36*.

Em seguida utilizando o comando *Extrude* será feita, olhando da vista superior, uma curva de aproximadamente meia elipse. Veja a *Figura37*.

Quando os alunos construíram a fita física tiveram que fazer uma rotação de 180° antes de colar as pontas. Aqui no *Blender* para fazer esta rotação é necessário usar os comandos *Rotation* e *Extrude* unir as arestas do início e final da fita. Veja na *Figura38*.

Então, para um resultado final mais atraente, na barra de ferramentas aplique dois *modifiers*: *Edgesplit* e *Subsurf*. Em seguida em *materials* utilize uma cor. Veja o resultado na *Figura39*.

Agora, basta importá-la para o *Unity* para dar continuidade no aplicativo.

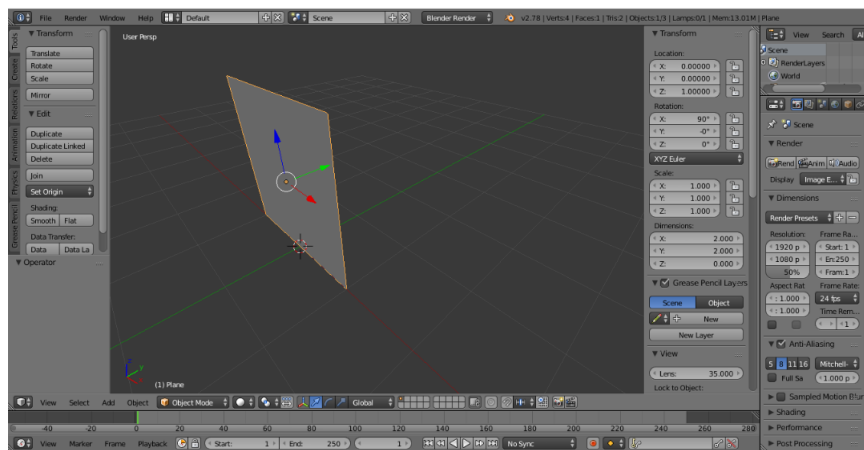


Figura 36: Plano no *Software Blender*

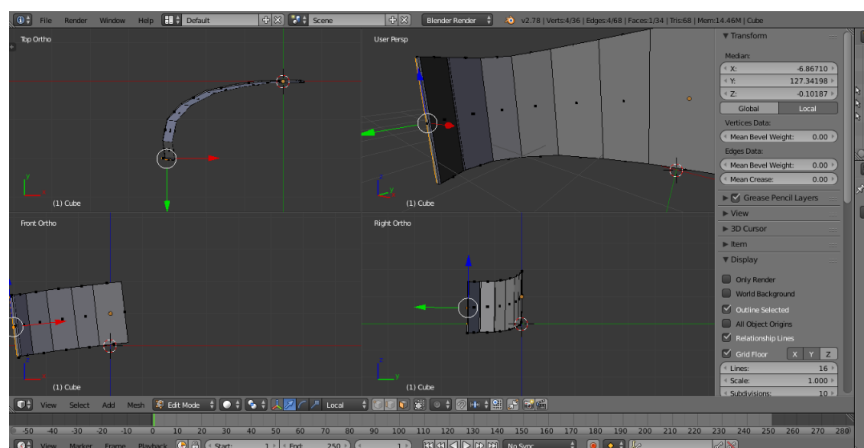


Figura 37: Construção *Fita de Möbius*

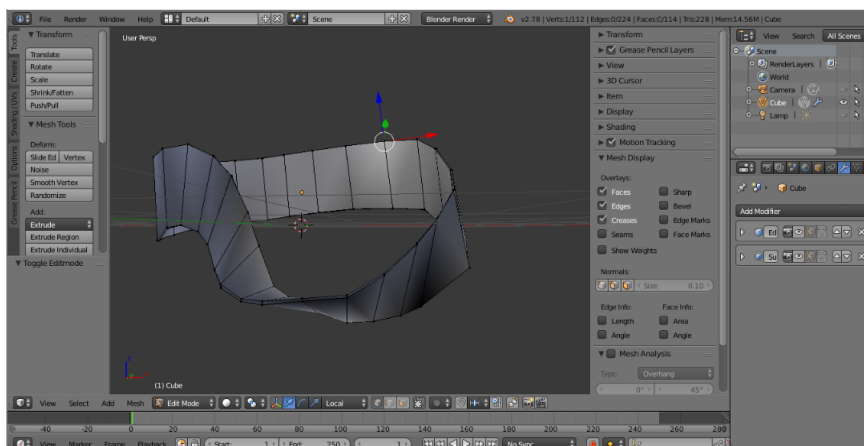


Figura 38: *Fita de Möbius* pré-pronta

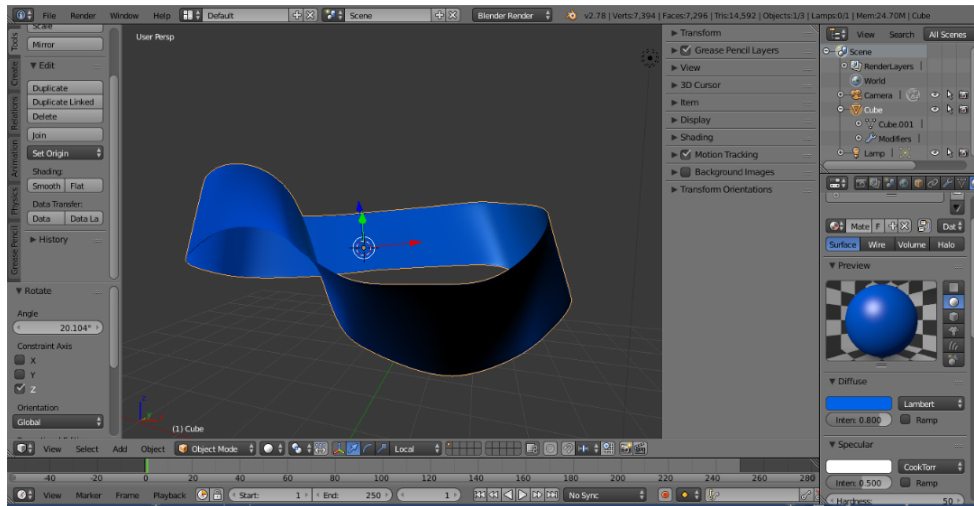


Figura 39: *Fita de Möbius* finalizada

5.7 Criando a *Garrafa de Klein*

Orientações

- Número de aulas: 2 aulas / 1 atividade.
- Público alvo: Alunos de ensino fundamental II e alunos de ensino médio.
- Material Utilizado: Computador com o *Software Blender* instalado.

Objetivos

- Compreender a relação entre a *Geometria Espacial* e *Geometria Plana*;
- Compreender Comandos básicos do *Blender*;
- Compreender a *Garrafa de Klein*;

Orientações ao professor

Para esta construção é de fundamental importância que o aluno tenha criado a *Fita de Möbius* para se familiarizar com os comandos, sendo assim apenas faça esta atividade caso já tenha concluído a atividade anterior.

Roteiro

Neste ponto é preciso ser feita uma introdução sobre o que é e como é formada a *Garrafa de Klein*, para tal, segue uma breve apresentação:

Descrita pela primeira vez em 1882 na Alemanha, pelo matemático Felix Klein (1849-1925), a chamada *Garrafa de Klein* é uma variedade unilátera fechada, não-orientável. Com característica de Euler igual a zero, ela não separa interior e exterior. Seu esquema topológico é parecido com o de uma *Fita de Möbius*. Enquanto uma *Fita de Möbius* é uma superfície com borda, uma *Garrafa de Klein* não possui borda (a título de comparação, uma esfera é uma superfície orientável sem borda). Uma *Garrafa de Klein* é um espaço topológico obtido pela colagem de duas *Fita de Möbius*. O nome se refere ao matemático Felix Klein. Caso tenha interesse em se aprofundar no assunto existe um belo artigo que pode ser acessado pelo link:

http://www.appoa.com.br/correio/edicao/243/o_discurso_histerico_e_a_garrafa_de_klein/183

Para a construção abra o *Software Blender*. Inicialmente será criado uma espécie de tubo oco. Para isso utilize dois cilindros com raios distintos e faça a diferença do maior e menor. Veja na *Figura40*.

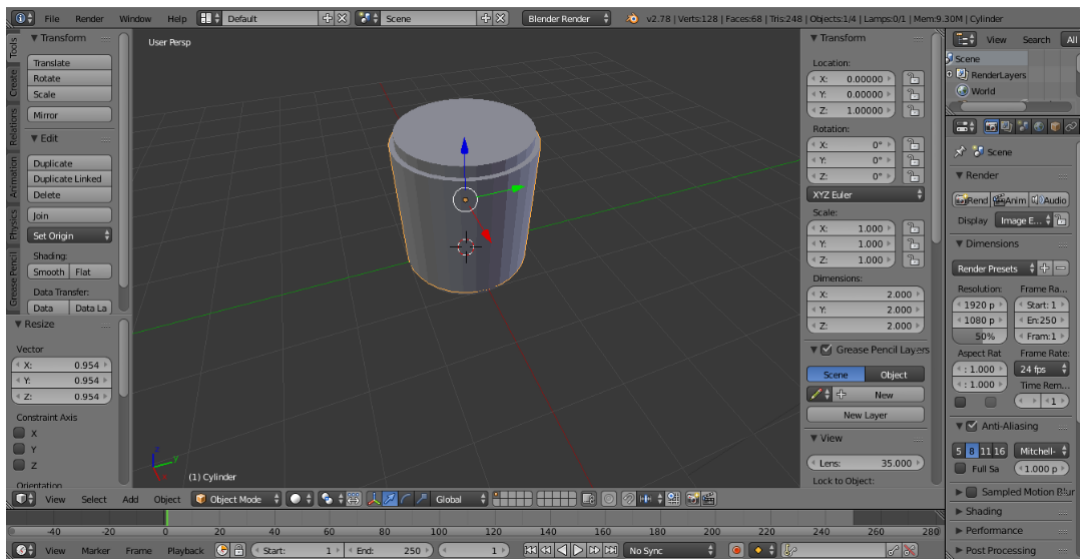


Figura 40: Cilindros no *Blender*

Em seguida utilizando o comando *Extrude* será feita uma especie de anel para poder ser modelado e se tornar uma *Garrafa de Klein*. Veja a *Figura41*.

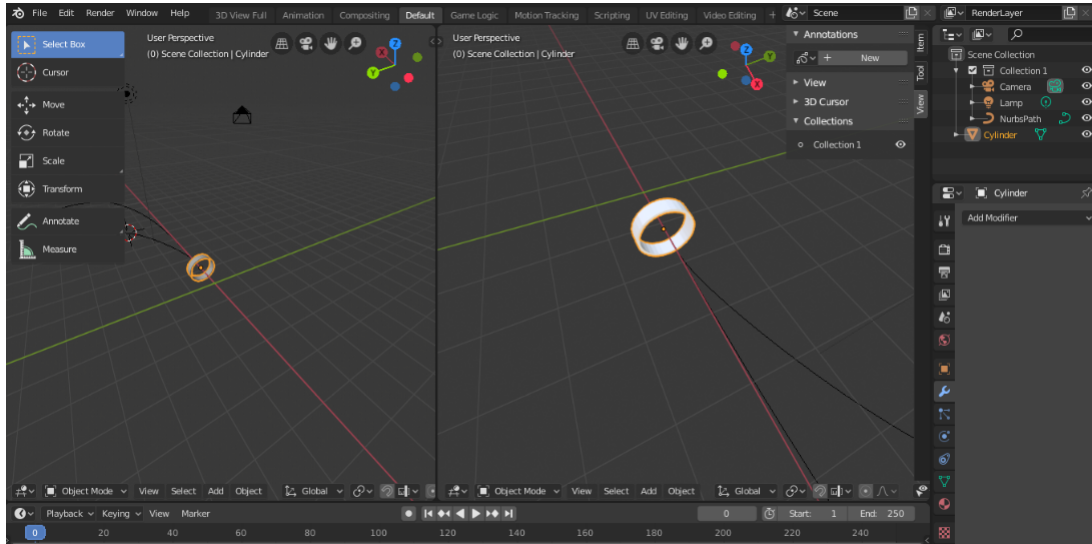


Figura 41: Anel no *Blender*

Então usando os comandos *Rotation* , *Extrude* e *Scale* será feita uma rotação de 360° seguindo o mesmo caminho da *Fita de Mobius* da atividade anterior, e por fim, basta unir a aresta à qual se está construindo a *Garrafa* à aresta do outro lado do anel que ficou imóvel desde o início da construção. Veja a *Figura42*.

Para um resultado final mais atraente, na barra de ferramentas aplique dois *modifiers*: *Edgesplit* e *subsurf*. Em seguida em *materials* utilize uma cor. Veja o resultado na *Figura43*.

Agora, basta importá-la para o *Unity* para dar continuidade ao aplicativo.

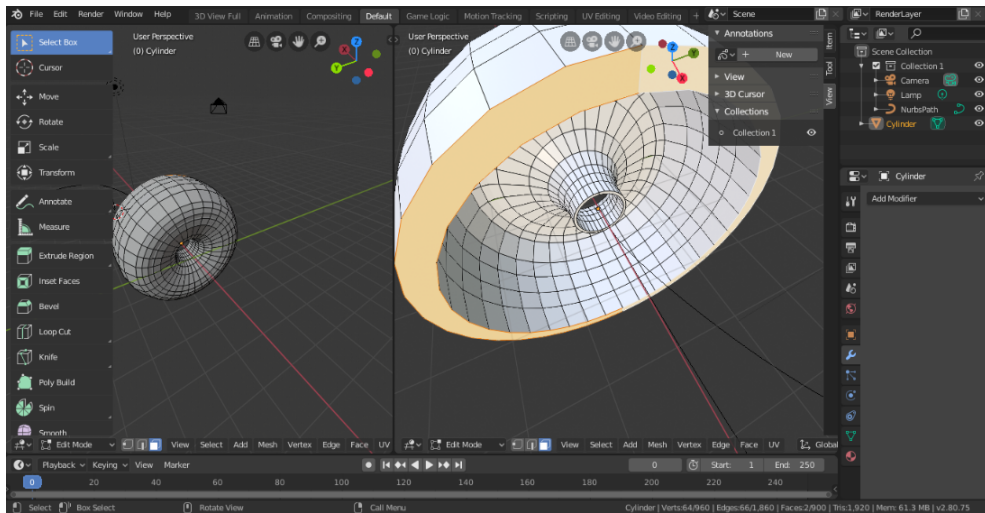


Figura 42: Garrafa de Klein em construção

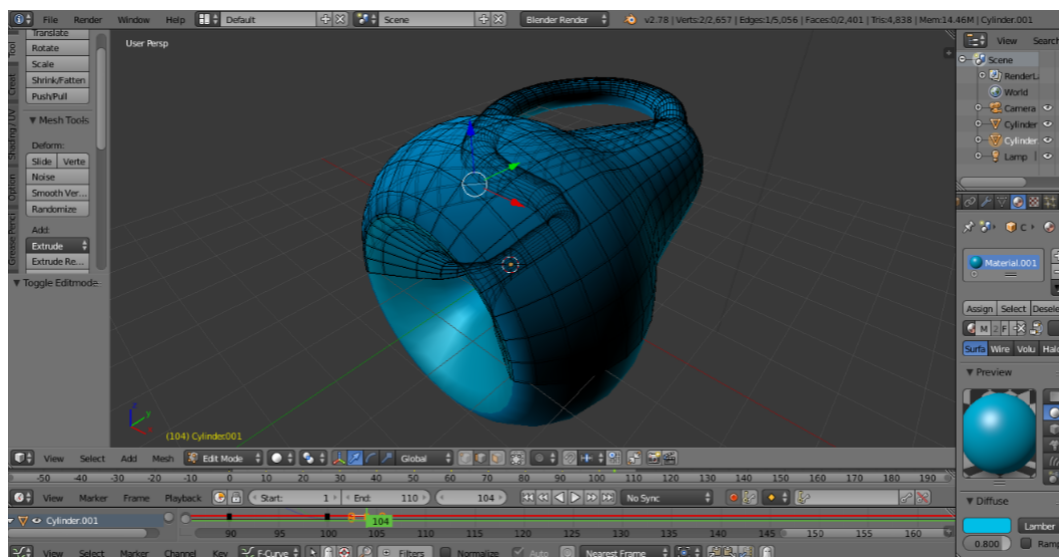


Figura 43: Garrafa de Klein finalizada

5.8 Configurações finais

Neste momento haverá uma quantidade muito grande de figuras, cenas e materiais na interface do *Unity*, portanto aqui cabe uma organização para que até mesmo o aluno entenda melhor o que representa cada um destes atributos.

Outro ponto importante é que até o presente momento o aplicativo possui apenas uma cena, o que tornaria inviável sua utilização uma vez que o usuário não teria um direcionamento para sua utilização. Logo existem alguns padrões que aplicativos devem seguir para que o usuário tenha uma boa experiência.

O sistema deve apresentar ao usuário uma interface simples e de fácil navegação. Ressaltamos aqui que apesar do aplicativo ter sido criado em sala, a utilização é livre para qualquer usuário, mesmo os que não tenham nenhuma experiência com programação, ou seja, o sistema deve ser intuitivo e o usuário não deve ter complicações em enxergar suas funcionalidades. Antes das criações das cenas por onde o usuário irá navegar, é importante que seja criado um diagrama para que a navegação fique mais simples. Este diagrama deve conter cenas que vão desde informações sobre o aplicativo, ou ajuda para utilizá-lo, até cenas com informações mais avançadas, por exemplo, de como utilizar a *Câmera AR*. O diagrama de navegação tem o intuito de demonstrar, de forma gráfica, a maneira como o usuário irá navegar de uma tela para outra no sistema. No aplicativo em questão a navegação entre as diversas telas do sistema se dá conforme o diagrama de navegação observado na *Figura 44*.

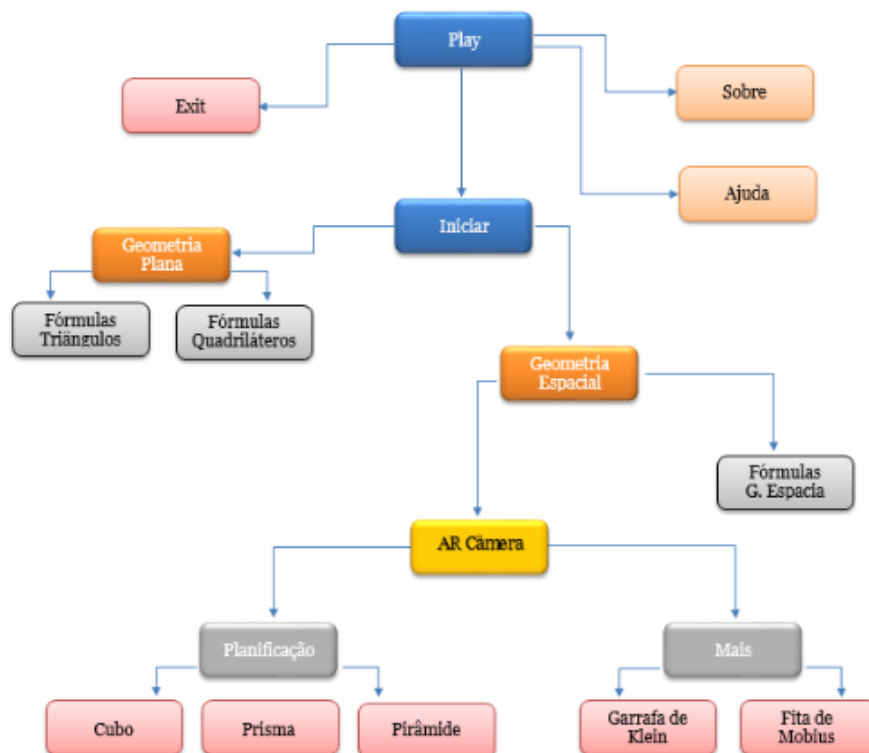


Figura 44: Diagrama de navegação

Após a construção deste diagrama, foi possível construir cenas e concluir a configuração no *Unity*. Para construir cada cena, basta seguir o caminho:

Creat>Scene

Outro ponto importante é que para transitar de uma cena para outra é preciso criar um botão e adicionar um pequeno *Script*, veja:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class LevelManager : MonoBehaviour
{
    public void LoadScene(string name)
    {
        SceneManager.LoadScene(name);
    }
    public void QuitGame()
    {
        Application.Quit();
    }
}
```

Feito isso e adicionando imagens de fundo segundo o gosto de cada aluno, clicando em *build* o aplicativo estará pronto para uso.

Caso necessite de ajuda, um tutorial pode ser acessado pelo link:

https://www.youtube.com/watch?v=5d-Mcj_YbUolist=LLV14CYKnQ3g3AawsdWdODIAgindex=34t=919s

6 Navegação no *Solid Planning*

A tela principal do aplicativo traz três caminhos, o primeiro botão é *Sobre* que é um botão de apresentação que traz informações sobre a criação. Outro botão é o *Ajuda* que é fundamental pois possui informações de como utilizar o aplicativo juntamente com o diagrama feito anteriormente, mas a importância deste botão se dá principalmente pelas informações de como utilizar a *AR Câmera* e o *link* de onde baixar as imagens. Por fim como destaque inclusive na cor, o botão *Iniciar*, que terá como uma das funções levar o usuário ao caminho da planificação. Veja esta tela inicial na *figura45*.



Figura 45: Tela inicial do aplicativo

6.1 *Geometria Plana* no Aplicativo

Como o objetivo principal é a planificação de sólidos, e como o usuário não conseguiria trabalhar de forma absoluta com a *Geometria Especial* sem uma boa base da *Geometria Plana*, o aplicativo traz em *Geometria Plana* as principais fórmulas sobre áreas e perímetros envolvendo triângulos e quadriláteros. Veja que as imagens e fórmulas apresentadas aqui foram criadas no *Software GeoGebra*, uma potente ferramenta que pode ser apresentada aos alunos na construção do aplicativo.

Veja os Quadriláteros na *Figura46*.

Veja os Triângulos na *Figura47*.

6.2 Fórmulas *Geometria Espacial* no Aplicativo

Quando o usuário clicar no botão *Geometria Espacial* ele será direcionado ao foco deste projeto, mas assim como foi feito na *Geometria Plana*, aqui existe uma base para a *Geometria Espacial*, a qual trará áreas e volumes para os principais sólidos geométricos. Veja que todas as figuras foram criadas no *GeoGebra* como anteriormente. Veja *Figura48*.

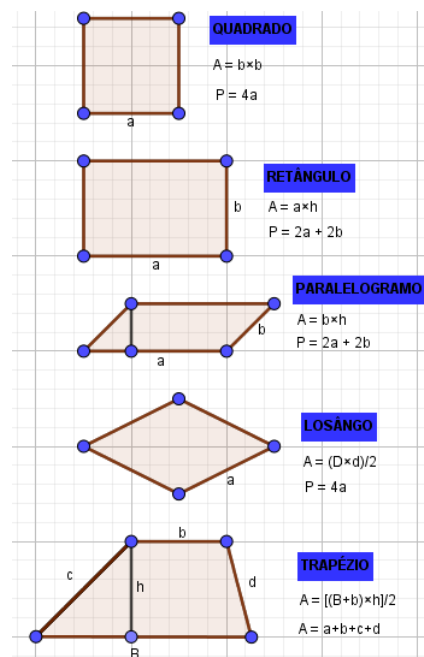


Figura 46: Fórmulas de quadriláteros

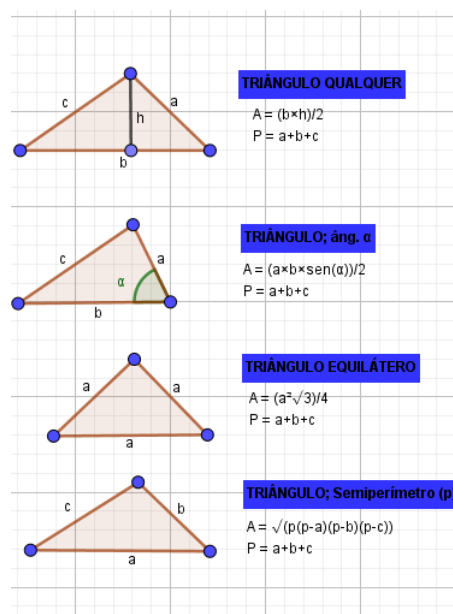


Figura 47: Fórmulas de triângulos

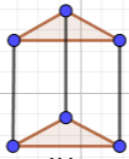
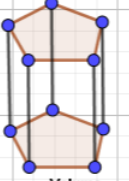
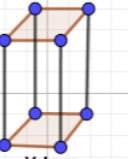

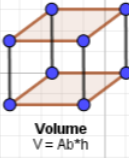
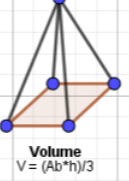
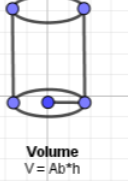
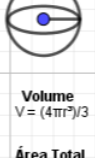
ÁREA TOTAL E VOLUME			
			
Volume $V = Ab \cdot h$	Volume $V = Ab \cdot h$	Volume $V = Ab \cdot h$	Volume $V = (Ab \cdot h) / 3$
Área Total $At = 2Ab + Al$	Área Total $At = 2Ab + Al$	Área Total $At = 2Ab + Al$	Área Total $At = \pi r(g + r)$
			
Volume $V = Ab \cdot h$	Volume $V = (Ab \cdot h) / 3$	Volume $V = Ab \cdot h$	Volume $V = (4\pi r^3) / 3$
Área Total $At = 2Ab + Al$	Área Total $A = Ab + Al$	Área Total $At = 2Ab + Al$	Área Total $A = 4\pi r^2$

Figura 48: Fórmulas *Geometria Espacial*

6.3 Planificação no Aplicativo

Neste momento o usuário estará pronto para a plena utilização do app, portanto será direcionado para o foco principal clicando em *AR Câmera* (Câmera de Realidade Aumentada). Como este projeto se propõe a ser um gatilho e amostra de como outros professores possam criar e planificar tudo quanto for necessário com os alunos, aqui não houve a necessidade de que muitas figuras fosse planificadas, no entanto a possibilidade de uma continuação e ampliação não é descartada. Sendo assim, por hora, três planificações serão apresentadas, o cubo, a pirâmide e o prisma. Em cada imagem será demonstrada a câmera sem a utilização do aplicativo e logo a direita com a utilização do aplicativo. veja a seguir a planificação do cubo na *Figura49*.

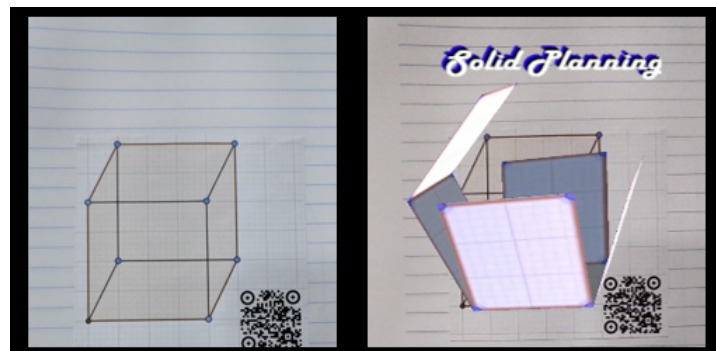


Figura 49: Planificação cubo

Como espera-se que este aplicativo seja utilizado em sala de aula, note que as imagens foram coladas em um caderno para a situação ser mais fidedigna ao contexto escolar. Veja agora a planificação do prisma na *Figura50*.

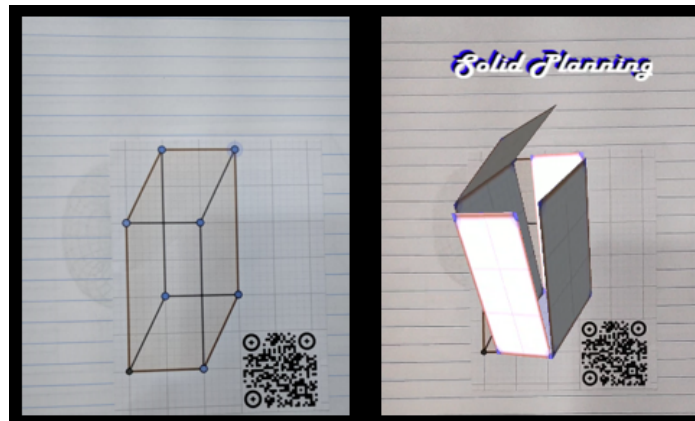


Figura 50: Planificação prisma

Por opção foi criado um prisma de base retangular, mas pode-se ser criado qualquer prisma. Veja a *Figura51*.

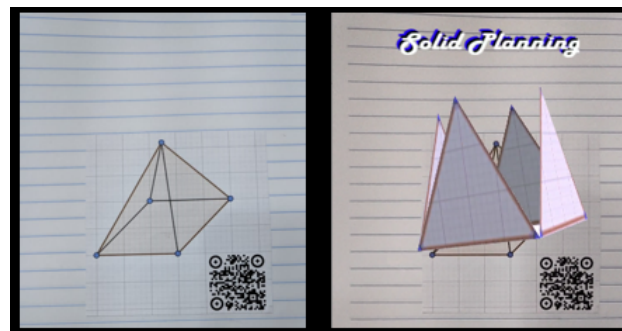


Figura 51: Planificação pirâmide

Da mesma forma do prisma, apesar desta pirâmide ser de base quadrada, pode-se ser criada qualquer pirâmide.

6.4 Mais sólidos no Aplicativo

Por fim, ao usuário clicar em *Mais* ele será direcionado à *Garrafa de Klein* e *Fita de Mobius*. Neste ponto pode-se dizer que o aplicativo traz um extra com a intenção de instigar a curiosidade do usuário. Esta é uma matemática mais avançada, e não será possível trabalhar com a planificação. Mesmo apesar de não serem planificadas elas estarão em constante movimento, e inclusive a *Garrafa de Klein* será aberta para que seja possível olhar por dentro. Veja a *Figura52*.

Nesta foto ela está fechada. É importante que o leitor utilize o aplicativo para ver os movimentos que a *Garrafa de Klein* reproduz. E por fim veja na *Figura53* a *Fita de Mobius*.

Como citado para *Garrafa de Klein*, é importante que o leitor utilize o aplicativo para ver os movimentos da *Fita de Mobius*.

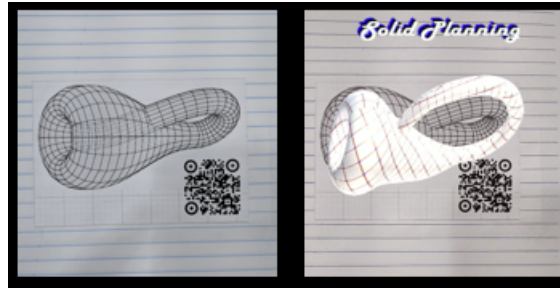


Figura 52: *Garrafa de Klein*

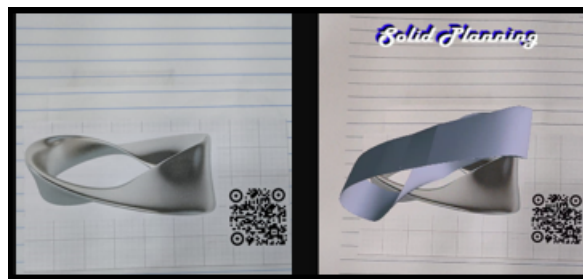


Figura 53: *Fita de Möbius*

7 Resultados obtidos

O presente estudo teve o intuito de demonstrar como um professor em sala de aula pode juntamente com seus alunos criar um aplicativo de realidade aumentada para facilitar a planificação de sólidos em sala de aula, sendo apresentados todos os *Softwares* juntamente com todas as programações envolvidas.

O aplicativo criado chamado *Solid Planning* foi bastante explorado, sendo que algumas alterações tiveram que ser realizadas durante o desenvolvimento do protótipo para se adequar ao que definiu-se como aplicativo final. Algumas ideias de funcionalidade surgiram no processo, mas foram limitadas para não fugir do escopo inicial, sendo que algumas foram listadas como trabalhos futuros na seção 6.1.

Como o protótipo foca o uso de smartphones, foi escolhido para teste o aparelho *Pocophone f1*, que possui uma tela de tamanho de 6.18 polegadas.

Todas as funcionalidades da aplicação foram validadas diversas vezes, se portando conforme o esperado. Porém, em alguns casos, constatou-se um movimeto desordenado da imagem na planificação, o que levou-se a conclusão de que deve ser feita uma boa impressão para que a câmera consiga ler.

Testes do aplicativo com usuários

A forma de avaliação do protótipo ocorreu por meio da aplicação de um questionário (ver Apêndice A). Os alunos selecionados para executarem os testes do aplicativo são todos pertencentes ao ensino fundamental e médio, em um número total de 6 alunos, os quais também são responsáveis pela elaboração do relatório do usuário.

Avaliação do produto

Por meio das questões propostas para o teste buscou-se avaliar os seguintes aspectos sobre a utilização deste aplicativo:

- a usabilidade;
- se o uso do aplicativo é considerado uma melhoria;
- quais melhorias podem ser realizadas no protótipo;

Para realizar esta avaliação o entrevistado deveria utilizar o protótipo e, ao final, responder algumas questões, as quais admitiam respostas no padrão *sim* ou *não*.

7.1 Síntese do resultado

A partir das respostas dos questionários, uma tabulação dos resultados foi realizada, a fim de subsidiar uma análise quantitativa dos resultados, conforme a *Figura 54*.

Perguntas	Sim (%)	Não (%)
Você precisou de algum treinamento ou experiência prévia para utilizar o protótipo?	85	15
O protótipo foi fácil utilização?	85	15
Você considera que a organização do aplicativo propicia uma navegação fácil e dinâmica?	100	-
A planificação pelo aplicativo é semelhante à real?	85	15
O diagrama apresentado no botão “Ajuda” auxiliou na navegação pelo aplicativo?	85	15
Você considera que a interface do aplicativo ajudou na utilização?	100	-
Você considera planificar utilizando o aplicativo em vez de usar folha cola e tesoura e construir um sólido de forma tradicional?	100	-
Você avalia que o protótipo atendo o objetivo proposto?	100	-

Figura 54: Resumo das respostas do questionário

Assim, pode-se constatar que os resultados foram satisfatórios, demonstrando que o aplicativo é de fácil utilização, as planificações muito se assemelham às planificações em papel, pode ser utilizado por qualquer pessoa, sem a necessidade de treinamento, além de ser útil e alcançar os objetivos deste trabalho.

Por outro lado, percebeu-se que não há uma manipulação do objeto no momento da planificação sendo que os movimentos dos sólidos já são previamente programados, o que causou um desconforto no usuário. Dada a oportunidade de que os entrevistados descrevessem sua opinião ou qualquer comentário que considerassem pertinente quanto ao presente trabalho, alguns pontos foram citados, dentre os quais, alguns se tornaram parte da atualização desta dissertação.

Ressalta-se que além das respostas ao questionário, cada usuário fez comentários verbais sobre a utilização do protótipo, que indicaram as vantagens de sua utilização em relação à planificação em papel.

Por fim, as falhas detectadas foram anotadas e posteriormente corrigidas, dentre elas, as cores da interface do aplicativo estarem muito opacas. Essas observações resultaram na imediata atualização do aplicativo.

8 Conclusão

O presente trabalho teve por meta auxiliar o professor na criação de um aplicativo para melhorar o trabalho com relação à planificação de sólidos em sala de aula. Deste modo, desenvolveu-se um aplicativo com foco principal na utilização da engine *Unity* juntamente com o *Vuforia* baseado em dispositivos móveis, cujo foco deu-se em aparelhos *smartphone*, podendo também ser utilizado em *tablets*.

Durante o desenvolvimento do protótipo foi utilizada a linguagem de programação *C Sharp*, que além de satisfazer as necessidades do projeto se mostrou uma escolha acertada pela facilidade para leigos aprenderem. Todas as ferramentas utilizadas no protótipo são gratuitas e apresentam vasta documentação para interessados em se aprofundar no assunto, possuindo listas de discussões em comunidades ativas de desenvolvedores, isto é importante para possibilitar a sua utilização sem onerar o Estado gerando com isso benefícios econômicos. Espera-se que este trabalho possa servir de reflexão e inspiração para a criação de aplicativos voltados à educação, utilizando-se do conceito de mobilidade e tentando abranger a maioria dos dispositivos móveis presentes no mercado brasileiro.

Dificuldades encontradas

Durante o processo de testes observou-se um problema que decorreu ao salvar o projeto; ao clicar no botão *Build* apareciam muitos erros com relação à versão do *Java* não estar compatível com a versão do *Unity*, sendo necessário portanto a utilização do *Unity 2017*. A respeito destes problemas foi realizada uma profunda busca a foruns e tutoriais de forma que ao final a maioria dos criadores apresentavam o mesmo problema e portanto após muitos estudos o consenso foi na utilização da versão do *Unity 2017* pela comunidade até novas atualizações.

Trabalhos futuros

Considerando que a aplicação foi apresentada em forma de um protótipo, poderão ser inseridas outras funcionalidades no futuro, que venham a contribuir para a evolução do trabalho realizado. Algumas possibilidades são apresentadas a seguir:

- Não precisar de uma imagem de gatilho para ativar a planificação possibilitando a planificação em qualquer superfície;
- Uma interface menos estática para uma melhor interação do usuário com o aplicativo;
- Um questionário no próprio aplicativo com um banco de questões próprio, de forma que o professor não precise de material complementar para atividades;
- Gráfico de desempenho do usuário com relação aos questionários oferecidos;
- Possibilidade do usuário dar um feedback visando a melhoria do app;

9 Referências Bibliográficas

- [1] ALMEIDA, Maria Elizabeth. ProInfo: Informática e formação de professores. Brasília: Ministério da Educação, Seed, 2000. 2v.
- [2] BONA, Berenice de Oliveira. Análise de *Softwares* educativos para o ensino de Matemática nos anos iniciais do Ensino Fundamental. Universidade Luterana do Brasil. Carazinho, RS – Brasil.
- [3] CAMARGO, Janira Siqueira. A Interação Professor aluno: A Escola como Espaço Interativo. In: MARTINS, João Batista (org.). Na perspectiva de Vygotsky. São Paulo: Quebra Nozes/ Londrina CEFIL, 1999. p.67-79.
- [4] CARVALHO, Marília G.; Bastos, João A. de S. L., Kruger, Eduardo L. de A./ Apropriação do conhecimento tecnológico. CEEFET-PR, 2000. Cap. Primeiro
- [5] DAMASCO NETO, J. R. Registros De Representação Semiótica E O GeoGebra: Um Ensaio Para O Ensino De Funções Trigonométricas. 2010. 130 f. Dissertação (Mestrado) - Curso de Pós-graduação Em Educação Científica E Tecnológica, Centro De Ciências Físicas E Matemáticas, UFSC, Florianópolis, 2010. Disponível em: . Acesso em: 21 dez. 2012.
- [6] DUMONT, A. H. Um estudo de caso sobre aspectos do conhecimento profissional de professoras que ensinam geometria em turmas de quarta série. Dissertação (Mestrado em Ciências) Universidade Federal Rural do Rio de Janeiro, Rio de Janeiro, 2008.
- [7] GEOGEBRA 5.0.414.0-d. *Software* de Matemática. Disponível em <https://www.geogebra.org/download>. Acesso em: 08 set. 2017.
- [8] GRAVINA, M. A.; SANTAROSA, L. M. A aprendizagem da matemática em ambientes informatizados. IV Congresso RIBIE, Brasília. 1998.
- [9] LIMA, Elon Lages.; CARVALHO, Paulo Cezar Pinto.; WAGNER, Eduardo.; MOR- GADO, Augusto Cesar. A Matemática do Ensino Médio - volumes 2 e 3 - 6.ed. - Rio de Janeiro: SBM 2006.
- [10] MORAN, José Manuel, MASETTO, Marcos, BEHRENS, Marilda. Novas tecnologias e mediação pedagógica. 6. ed. São Paulo: Papirus, 2003.
- [11] Orientações Curriculares para o Ensino Médio. Volume 3: Ciências Humanas e suas tecnologias Brasília-2006. Ciências humanas e suas tecnologias / Secretaria de Educação Básica. – Brasília: Ministério da Educação, Secretaria de Educação Básica, 2006.

[12] VALENTE, José Armando (org). O computador na sociedade do conhecimento. Campinas: UNICAMP/NIED, 1999

10 Apêndice

10.1 A - Questionário aplicado aos alunos

1. Você precisou de algum treinamento ou experiência prévia para utilizar o protótipo? <input type="radio"/> Sim <input type="radio"/> Não
2. O protótipo foi fácil utilização? <input type="radio"/> Sim <input type="radio"/> Não
3. Você considera que a organização do aplicativo propicia uma navegação fácil e dinâmica? <input type="radio"/> Sim <input type="radio"/> Não
4. A planificação pelo aplicativo é semelhante à real? <input type="radio"/> Sim <input type="radio"/> Não
5. O diagrama apresentado no botão “Ajuda” auxiliou na navegação pelo aplicativo? <input type="radio"/> Sim <input type="radio"/> Não
6. Você considera que a interface do aplicativo ajudou na utilização? <input type="radio"/> Sim <input type="radio"/> Não
7. Você considera planificar utilizando o aplicativo em vez de usar folha cola e tesoura e construir um sólido de forma tradicional? <input type="radio"/> Sim <input type="radio"/> Não
8. Você avalia que o protótipo atendo o objetivo proposto? <input type="radio"/> Sim <input type="radio"/> Não

Figura 55: Questionário

10.2 B - Links para *Download*

Download unity

<https://www.unity3d.com>

Download Visual Studio

<https://visualstudio.microsoft.com/pt-br/vs/?rr=https%3A%2F%2Fwww.google.com%2F>

Download Blender

<https://www.blender.org/download/>

Download GeoGebra

<https://www.geogebra.org/download?lang=pt>

Download figuras AR Câmera

<https://drive.google.com/open?id=1OnUMAENIjZTmwbHAOUWLg8c460DmcUYH>