

Relatório Final

Simulação de experimentos de óptica
para ensino de física – Interface gráfica e
Fendas

Aluno: Carlos Eduardo Scussiatto

RA: 001447

Orientador: Pedro Miguel Raggio Santos

Curso: F809 – Instrumentação

1. Introdução

O projeto foi desenvolvido com a intenção tornar o ensino de óptica mais simples através de recursos visuais. Pensando nisso foi escolhido desenvolver um programa de computador em ambiente gráfico que possa simular tópicos de óptica.

Conforme descrito no projeto, o ambiente gráfico e o módulo de Fendas foram desenvolvidos pelo aluno Carlos E. Scussiatto e o módulo de refração em Lentes pelo aluno Marcelo de Freitas Rigon. Além dos alunos diretamente envolvidos neste projeto, o desenvolvimento do mesmo contou com o auxílio de Pedro Santos, do professor José J. Lunazzi e o apoio informal de diversos amigos.

Nesse relatório consta uma breve descrição da escolha de linguagem de programação Python usada no desenvolvimento do programa, bem como, da descrição do sistema e perspectivas futuras.

2. Por que Python ?

Cada programa tem sua própria cultura e estilo, sua própria forma de resolver problemas. O Python tem um estilo próprio distinto das demais linguagens. Guido van Rossum, foi o criador da linguagem. Ele tinha em mente uma linguagem que pudesse tornar as coisas mais fáceis na hora de programar. Dessa idéia surgiu o Python, uma linguagem que tenta deixar a tarefa de programar mais “ortogonal”, mas nem tanto, e que ajuda o programador tanto quanto possível.

Guido não parou por aí. Ele queria uma linguagem que todas as pessoas pudessem, com um mínimo de conhecimento em informática, aprender e desenvolver programas simples para auxiliar na vida diária.

A escolha do Python surgiu por causa desses motivos: uma linguagem simples e poderosa. O Python não é tão poderoso quanto um C ou um Fortran, mas sua facilidade permite resultados muito mais rápidos dos quais poderiam ser obtidos em qualquer outra linguagem.

3. Módulo de fendas

No módulo de fendas encontra-se o estudo da difração segundo a teoria de Fraunhofer. A difração de Fraunhofer corresponde ao caso em que a fonte de luz e o anteparo no qual o fenômeno da difração é observado estão em posições que correspondem efetivamente a distâncias infinitas do obstáculo ou da fenda em estudo.

Matematicamente, a difração de Fraunhofer corresponde ao caso que ambas, a onda incidente e a emergente, podem ser consideradas como sendo frentes de ondas planas.

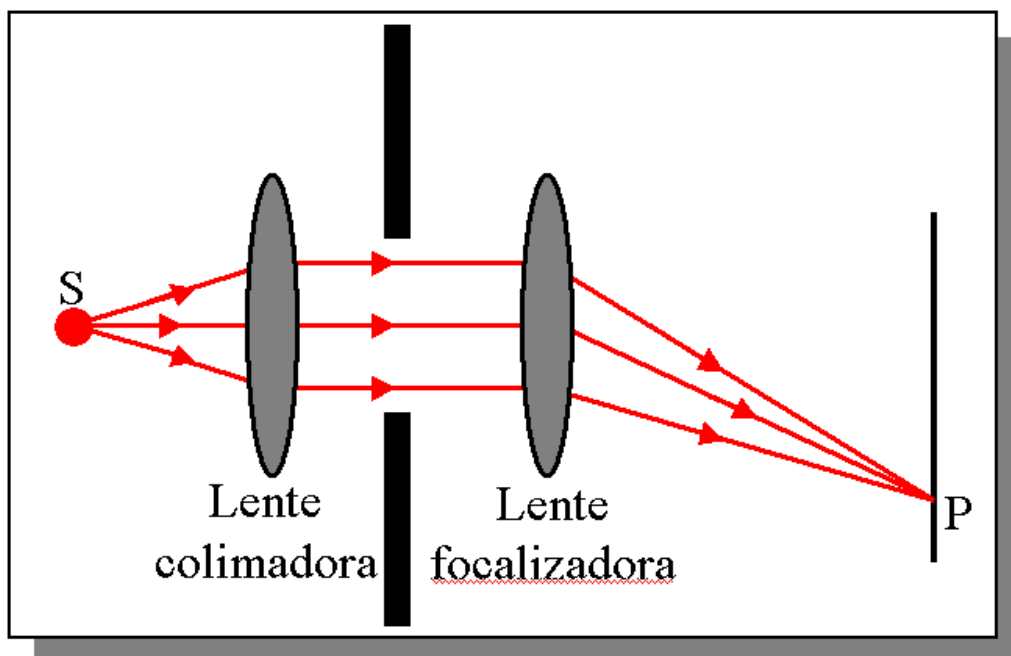


Figura 1. Arranjo experimental da difração de Fraunhofer com fenda única.

A rede de difração

Utilizando um número grande de fendas, N , podemos contruir uma rede de difração. A figura a seguir apresenta o esquema de fendas múltiplas que representam uma rede de difração:

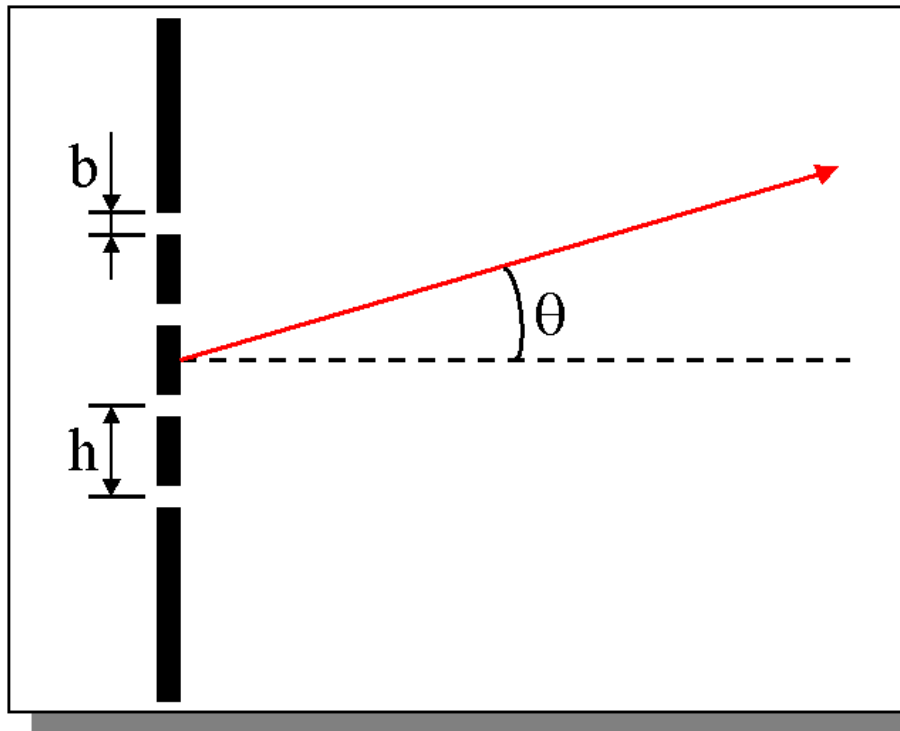


Figura 2. Fendas múltiplas

Através da teoria de Fraunhofer, que é baseada no princípio de Huygens, pode-se obter uma equação que nos fornece a relação entre as intensidades do feixe de luz incidente e da luz num ponto qualquer do anteparo (ref.B-3):

$$I = I_0 \left| \frac{\text{sen } \beta}{\beta} \right|^2 \left| \frac{\text{sen } N \gamma}{N \text{ sen } \gamma} \right|^2$$

, onde

- I: Intensidade do feixe num ponto do anteparo
- I_0 : Intensidade do feixe incidente
- N: Número de fendas

β e γ são variáveis que dependem de k, b e θ :

$$\beta = \frac{1}{2} k b \text{ sen } \theta \quad \gamma = \frac{1}{2} k h \text{ sen } \theta$$

- k: número de onda
- b: tamanho da fenda
- h: distância entre duas fendas

4. Arquitetura do sistema

O programa foi totalmente desenvolvido em Python 2.2 e a interface gráfica foi construída a partir do pacote gráfico Pygame 1.5.5, pelo aluno em questão. O Pygame permite o acesso rápido a modos gráficos, desenho de objetos, carregamento de imagens e funções de entrada e saída a partir do mouse e teclado. A existência de uma interface gráfica facilita em muito a utilização. O tempo de aprendizado fica muito menor do que se fosse utilizado uma interface em modo texto.

O objetivo principal do projeto era disponibilizar um programa de física cujo aprendizado ficaria facilitado através de simulações em multimídia. Por causa do tempo ganho no aprendizado do programa o usuário terá mais tempo para aprender os conceitos e experiências contidos nos módulos. O sistema consiste em uma interface gráfica dividida basicamente em três seções:

- Módulos de óptica
- Módulos de mecânica
- Créditos

Os arquivos do sistema estão em código fonte, ou seja, não estão compilados. Tendo em vista que o python é uma linguagem interpretada, para rodá-los não é necessária a compilação. Isso facilita no desenvolvimento porque é possível executá-lo em qualquer plataforma que tenha o Python instalado, além de permitir que o próprio usuário faça pequenas modificações ou mesmo, inclua novos módulos. Os pacotes necessários para utilização do sistema encontram-se no Anexo-III.

A figura a seguir mostra a janela inicial do programa.

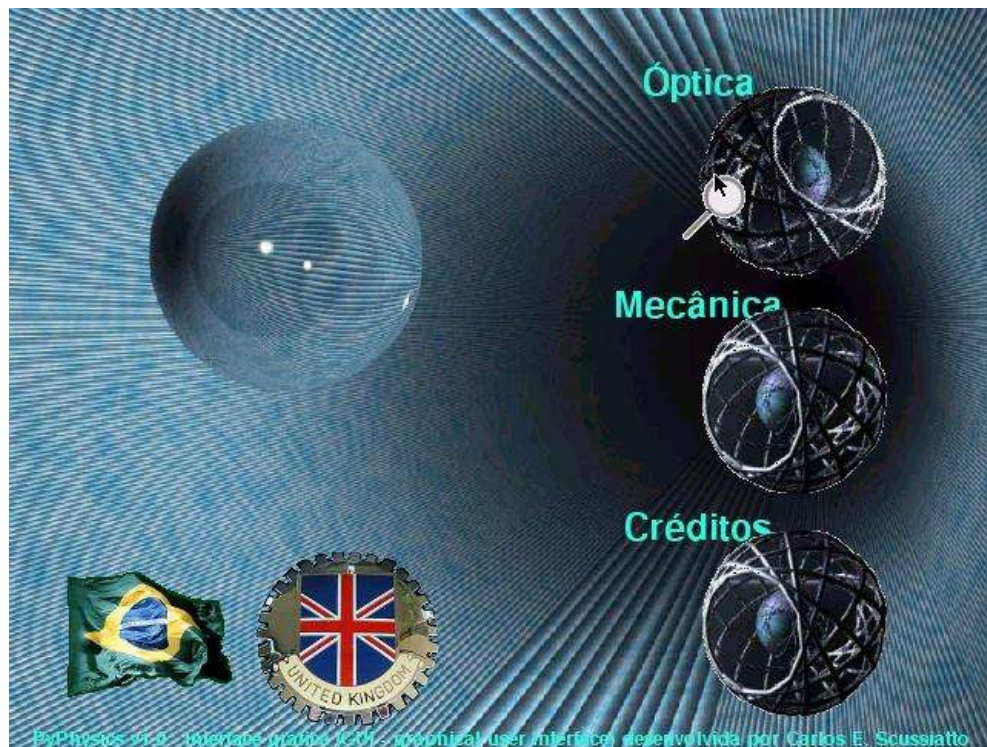


Figura 3. Tela inicial do programa

Para acessar cada um dos menus o usuário basta clicar no menu desejado. Além disso, a tela inicial permite que o usuário selecione o idioma. Por exemplo: se o usuário não souber português basta que ele clique no botão que contém a bandeira do Reino Unido e o programa automaticamente muda todos os menus para o idioma Inglês.

Se o usuário desejar conhecer as pessoas envolvidas diretamente no projeto basta que ele clique no menu “Créditos”. Dessa forma aparecerá a janela da figura a seguir, com informações sobre e-mail e icq dos projetistas do sistema. Essa é uma maneira dos usuários poderem pedir mais informações sobre programa ou mesmo sugerirem novos módulos e alterações dos mesmos.

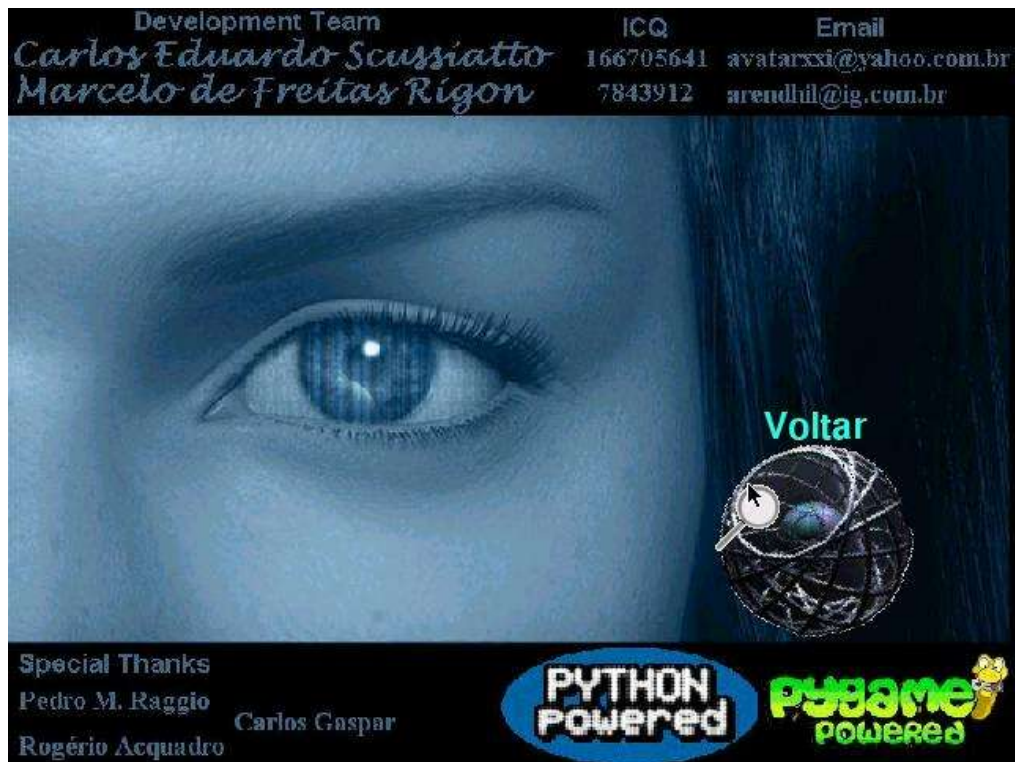


Figura 4. Tela do menu “Créditos”

Clicando no menu “Mecânica” o usuário terá acesso ao menu que permite entrar nos módulos de mecânica, desenvolvidos no semestre anterior por Rogério Acquadro.

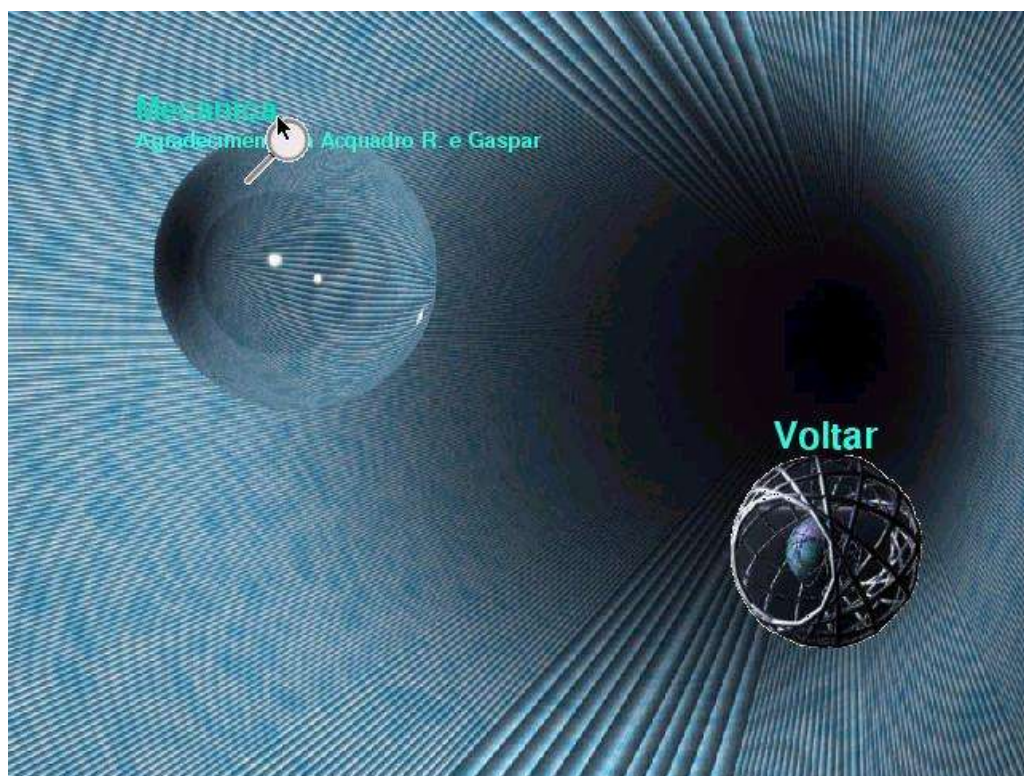


Figura 5. Tela do menu “Mecânica”

Da mesma forma o usuário pode acessar também o menu “Óptica”, clicando sobre ele. Esse menu permite o acesso aos módulos de fendas e lentes desenvolvidos pelos alunos Carlos E. Scussiatto e Marcelo F. Rigon, respectivamente.

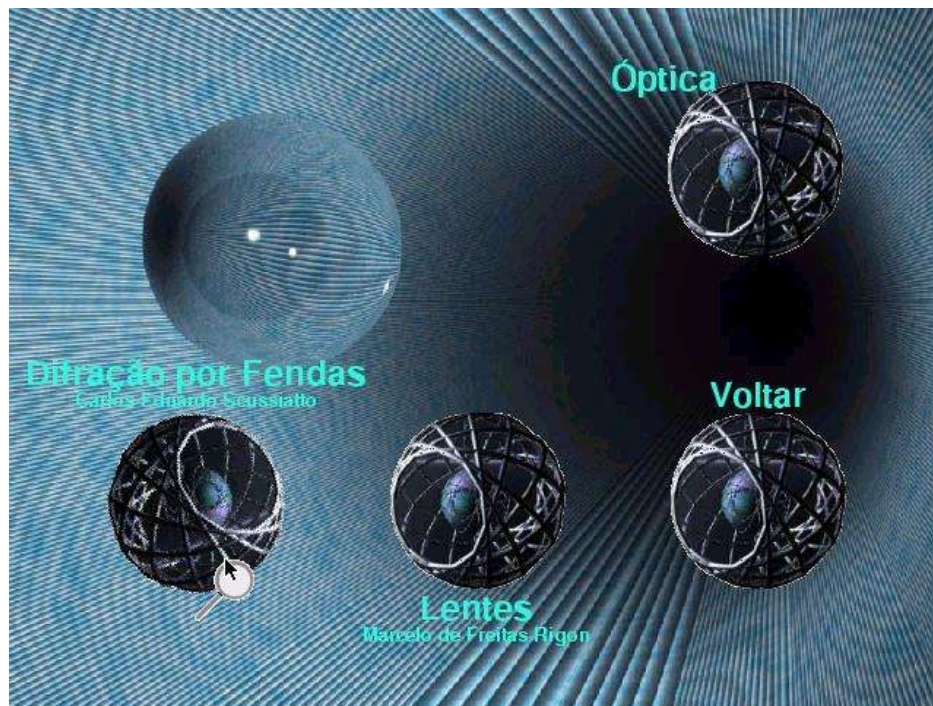


Figura 6. Tela do módulo “Óptica”

Utilização do módulo de fendas

Para utilizar o módulo de fendas basta que o usuário clique no menu “Difração por Fendas”.



Figura 7. Tela do módulo de fendas

A utilização dos módulos é muito simples. Basta que o usuário entre com os parâmetros N, b, h e comprimento de onda λ e clicar no botão “rodar”. Para colocar os parâmetros o usuário pode clicar no texto do parâmetro desejado e digitar seu valor. Além disso, se o botão “Enter” do teclado for pressionado o programa automaticamente vai para o próximo parâmetro.

Mesmo que o usuário não digite nenhum valor ele pode clicar no botão rodar. Contudo os valores utilizados serão valores padrão do sistema. Assim que o botão “rodar” for clicado, os dados utilizados são atualizados automaticamente.

O botão “rodar” faz com que o programa calcule e apresente o resultado de uma curva de I/I_0 em uma janela nova. Além disso, na mesma janela de dados é desenhada uma figura de difração em tons de cinza representando os picos de intensidade da luz. Ou seja, para valores grande de I/I_0 a cor será próxima do branco; para valores pequenos de I/I_0 ela se aproxima do preto.

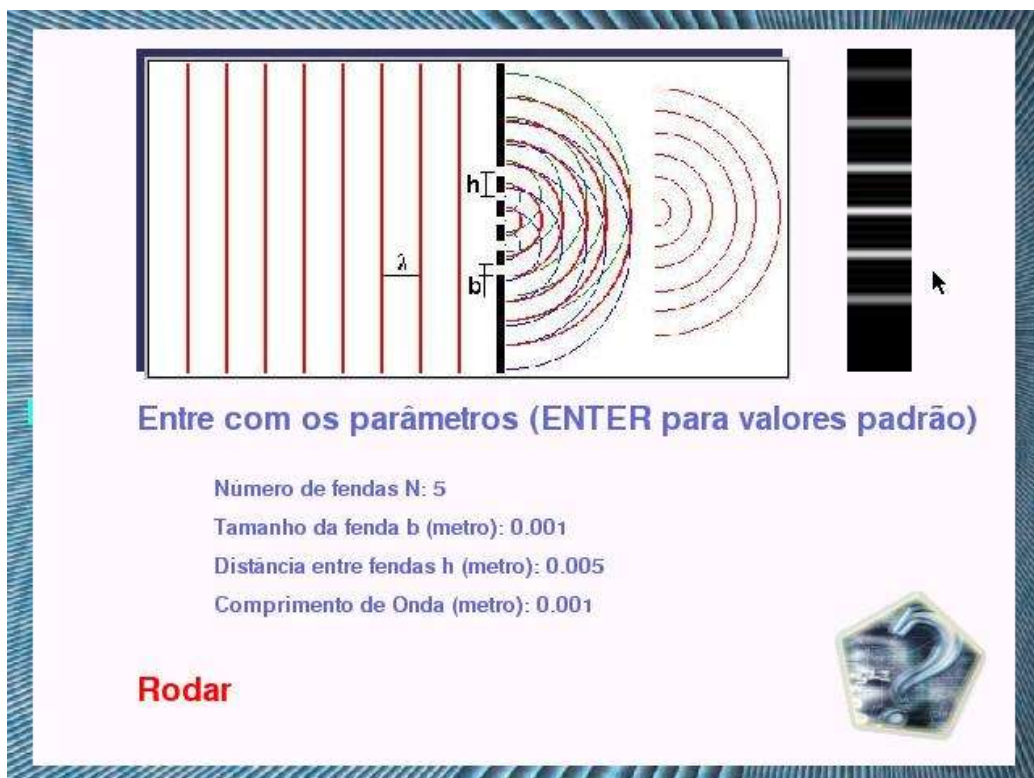


Figura 8. Tela mostrando a figura de difração e a entrada padrão de dados

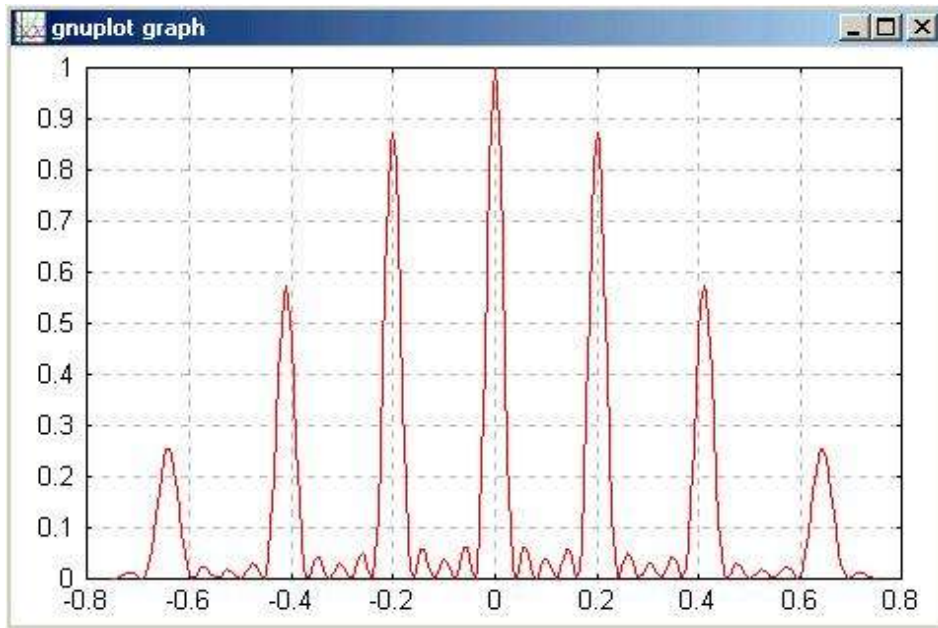


Figura 9. Tela da gráfico equivalente a entrada padrão de dados

Além do módulo de fendas, o usuário pode usar o módulo de lentes. Neste módulo o usuário poderá adicionar lentes a sua escolha e alterar parâmetros como raio de curvatura das faces e índice de refração das lentes.

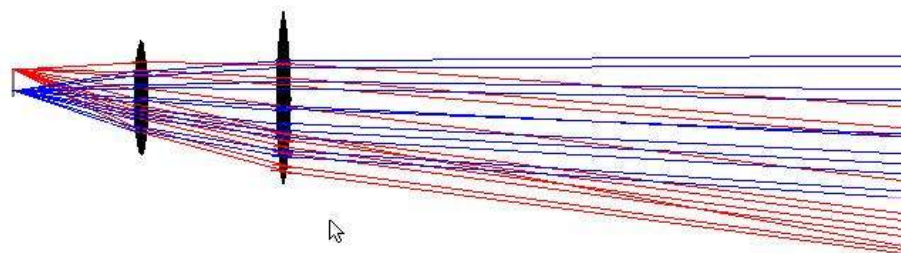
Posição do ponteiro:(232, 326)

Nova lente

Apaga lente

Apaga os raios

Apaga tudo



Lente número:

Posição no x:

Altura:

Espessura:

Curvatura 1:

Curvatura 2:

Índice de refração:



Figura 10. Tela do módulo lentes (negativo das cores)

De acordo com o que foi previsto no projeto elaboramos um sistema que permite o ensino de tópicos de óptica. Nesses módulos adicionamos também um botão de ajuda rápida. Ao clicar na interrogação o sistema abre um arquivo pdf com ajuda sobre o módulo. Agradecemos a Carlos Gaspar por ter escrito esses arquivos de ajuda em pdf. Além disso também incluímos o programa Mecânica desenvolvido no semestre anterior.

Apesar de terminamos o programa de acordo com o cronograma esse sistema permite que outras pessoas dêem continuidade ao projeto. Além disso também temos a intenção futura de adicionar outras simulações no tópico mecânica que usam o pacote “Kinetics Kit”, desenvolvido por Markus Gritsch utilizando a biblioteca Visual Python. Para incentivar o desenvolvimento do software livre aplicado ao ensino de ciências pretendemos também, a sim que possível, publicar um artigo numa revista do assunto, como por exemplo, na Revista Brasileira de Ensino de Física.

6. Agradecimentos

Gostaria de agradecer ao trabalho realizado no semestre passado e o apoio nos dado nesse semestre por Rogério Acquadro e Carlos Gaspar. Pois foram eles os responsáveis pela criação do projeto piloto de uma pequena rede de micros GNU/Linux no semestre passado além do programa GPL intitulado Mecânica desenvolvido pelo Rogério. Também agradeço ao apoio dado por nosso orientador Pedro Miguel Raggio Santos e pelo professor José Joaquim Lunazzi que possibilitou o refinamento de nosso sistema até o ponto de se tornar um programa robusto e ao mesmo tempo simples de uso.

7. Referências

Bibliografia:

- 1) ***Python Tutorial*** – release 2.2.2, de Guido van Rossum, editado por Fred L. Drake, PythonLabs
- 2) **Física** – Volume 3, de José Gondeberg – Companhia Editora Nacional, Editora da Universidade de São Paulo.
- 3) **Principles of Optics** , de Born & Wolf, sexta edição, Cambridge
- 4) **Pygame v1.5.5 Documentation**

Internet:

1. <http://www.gnu.org/>
2. <http://www.wxpython.org/>
3. <http://www.debian.org/>
4. <http://www.linux-france.org/prj/mek/>
5. <http://www.pythonemproject.com/>
6. <http://www.scipy.org/>
7. <http://www.opensource.org/>
8. <http://www.vex.net/parnassus/>
9. <http://www.pygame.org>

ANEXOS

Anexo I – Software Livre

Os programas mais divulgados na população, atualmente, são programas proprietários. Os programas proprietários são distribuídos de forma a permitir que o usuário utilize o programa sem poder modificar e adaptá-lo ou melhorá-lo. Isso é feito não transmitindo ao usuário o código fonte (arquivo utilizado para a criação do programa) e/ou através de licenças que o proibem de modificar o programa de qualquer forma.

Os programas livres, ao contrário, incentivam a modificação do seu código fonte pelos seus usuários, de forma a evoluir, corrigir problemas ou adaptar esse programa aos mais diversos tipos de necessidades de seus usuários.

Fazendo uma analogia com roupas:

Os programas proprietários seriam como uma roupa que você compra e não pode fazer a barra, apertar a cintura e coberta por um plástico opaco que não permita que você veja como é a costura dela.

Os programas livres são as roupas da maneira que são comercializadas atualmente. Você pode apertar a cintura se a calça estiver caindo, ou fazer a barra se estiver muito comprida para que ela se torne algo nas medidas certas para você.

Anexo II - Nasce uma nova linguagem: Python

No início dos anos 80, Guido van Rossum trabalhava em uma equipe que estava desenvolvendo uma linguagem chamada ABC no Instituto Nacional de Pesquisa em Matemática e Computação Científica (Centrum voor Wiskunde en Informatica, CWI) da Holanda. ABC era uma linguagem clara que tinha a intenção de ensinar programação a pessoas inteligentes que utilizavam computadores, mesmo não sendo programadores - seus estudos incluíam de físicos e cientistas sociais, a linguistas - mas essas pessoas ficaram surpresas com as limitações, restrições e regras arbitrárias que as linguagens estabeleciam. Entre muitas razões e a não existência de uma internet para fazer a distribuição dessa linguagem, o projeto não foi um sucesso.

Em 1986, ele se mudou para um projeto diferente no CWI, o projeto Amoeba (Amoeba era um sistema operacional da época). No final dos anos 80, eles descobriram que estavam precisando de uma linguagem de scripting. Como Guido van Rossum já tivera uma experiência com a linguagem ABC e liberdade o suficiente para criar um mini projeto dentro do escopo do seu atual trabalho, ele decidiu desenvolver uma linguagem de scripting simples que possuísse as melhores qualidades do ABC sem os seus problemas.

Pegando os ingredientes do ABC e mexendo um pouco, surgiu o Python. O Python é muito semelhante ao ABC, mas também tem algumas diferenças. Entre elas as variáveis do tipo lista e dicionário, comandos básicos, o uso de indentação para gerenciar início e fim de blocos de comandos.

A mais importante e inovativa diferença do Python para o ABC que garante grande parte do sucesso que essa linguagem vem tendo é a facilidade para estendê-la.

Anexo III – Instalação do programa

O programa requer que usuário tenha instalado em seu sistema o Python 2.2 e as bibliotecas Pygame 1.5.5, Numeric e Scipy. Esses programas são de fácil obtenção e instalação. Basta que o usuário acesse os sítios de cada pacote e baixe a versão correta e a plataforma correspondente. Os programas são gratuitos e estão disponíveis para várias plataformas.

- **Python 2.2**

<http://www.python.org/>

- **Pygame 1.5.5** for python 2.2

<http://www.pygame.org/>

- **Numeric** for python 2.2

<http://www.vex.net/parnassus/>

- **Scipy** for python 2.2

<http://www.scipy.org/>

Quando todos os pacotes estiverem instalados basta que o usuário rode o arquivo principal do nosso sistema, “Main.py”, a partir do Python 2.2