LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-415585

Interactive Game for Teaching Laser Amplification Used at the National Ignition Facility

E. Lin

August 13, 2009

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Interactive Game for Teaching Laser Amplification Used at the National Ignition Facility

Eric S. Lin

Office of Science, Science Undergraduate Laboratory Internship Program

University of California, Berkeley

Lawrence Livermore National Laboratory

Livermore, California

August 6, 2009

Prepared in partial fulfillment of the requirement of the Office of Science, Department of Energy's Science Undergraduate Laboratory Internship under the direction of Jill Garland in the National Ignition Facility and Photon Science Directorate at Lawrence Livermore National Laboratory.

Participant: ___________________________

Advisor: ____________________________

Table of Contents

Abstract

Interactive Game for Teaching Laser Amplification Used at the National Ignition Facility. ERIC S. LIN (University of California, Berkeley, CA 94720) JILL D. GARLAND (Lawrence Livermore National Laboratory, Livermore, CA 94550)

The purpose of this project was to create an interactive game to expose high school students to concepts in laser amplification by demonstrating the National Ignition Facility's main amplifier at Lawrence Livermore National Laboratory. To succeed, the game had to be able to communicate effectively the basic concepts of laser amplification as accurately as possible and to be capable of exposing as many students as possible. Since concepts need to be communicated in a way that students understand, the Science Content Standards for California Public Schools were used to make assumptions about high school students' knowledge of light. Effectively communicating a new concept necessitates the omission on terminology and symbolism. Therefore, creating a powerful experience was ideal for communicating this material. Various methods of reinforcing this experience ranging from color choice to abstractions kept the student focused on the game to maximize concept retention. The program was created in Java to allow the creation of a Java Applet that can be embedded onto a webpage, which is a perfect medium for mass exposure. Because a game requires interaction, the game animations had to be easily manipulated to enable the program to respond to user input. Image sprites, as opposed to image folders, were used in these animations to minimize the number of Hypertext Transfer Protocol connections, and thus, significantly reduce the transfer time of necessary animation files. These image sprites were loaded and cropped into a list of animation frames. Since the caching of large transition animations caused the Java Virtual Machine to run out of memory, large animations were implemented as animated Graphics Interchange Format images since transitions require no interaction, and thus, no frame manipulation was needed.

This reduced the animation’s memory footprint. The first version of this game was completed during this project. Future work for the project could include the creation of focus groups to assess the effectiveness of communicating material through an interactive game. Numerical assessments programmed into the game could also be used to collect statistics that reflect difficulty or level of frustration that students experience.

1. Introduction

As the scientific community continues to accumulate more knowledge, it becomes necessary for students to learn an increasing amount of scientific material. This increasing amount of material in a student's curriculum creates the need to develop methods that can help make the learning process more efficient. Efficient learning is usually most apparent during the early stages of childhood where most of a child's knowledge is in the form of physical properties [1]. Taking advantage of this fact, the purpose of this project was to create an interactive computer program, or game, that exposes high school students in the United States to laser amplification concepts by allowing them to explore the method used in the National Ignition Facility's (NIF) main amplifier.

It was decided that a game was the best medium to communicate this new material for several reasons. First exposure to a concept necessarily implies that students are not familiar with the terminology or equations that go with laser amplification. Therefore, the material must be communicated through an experience as opposed to language and symbolism. A game creates an experience that lies more in the physical domain than other mediums such as videos or books, which utilize more visual and symbolic methods [1]. As such, a game communicates information in a way that is more easily understood. Although the physical domain of communication lacks the capability to explain the physical principles behind laser amplification, it is more important to present what goes on in laser amplification rather than why it works to students who have never seen a laser amplifier before, and therefore, a game remains an ideal medium.

2. Design

As with any communication medium, the key to creating a powerful and effective experience is to focus on one idea, which from here on will be referred to as the essential experience [2]. For this program, it was decided that the essential experience would be the amplification of a laser beam. After an essential experience had been decided upon, the manifestation of this experience, the theme, was devised such that it would reflect the true nature of laser amplification; therefore, it was decided that the theme would be creating lots of energetic photons, which will be named good photons. Having a theme meant that theme reinforcement was one of the top priorities in the design of this program since a theme with weak reinforcement would create a lackluster experience [2]. It will be pointed out throughout the design explanation how the theme was reinforced in various parts of the game. In addition, in order to make this experience most effective in conveying its message, the California educational standards were used to make assumptions about what students do and do not know about light and lasers.

After the more general guidelines of essential experience and theme were laid out, the balance between the game's challenge and students' skills had to be thought out. This balance is necessary to maintain motivation and interest in continuing the game by avoiding the student becoming either bored or frustrated, which would greatly distract the student [2]. Initially, the only reward that the student would receive was seeing new animations or earning a good score after completing a stage. While these are rewarding in their own way, it was decided that there was not enough justification to create challenges that put too much emphasis on the student's critical thinking skills. Therefore, critical thinking was kept to a mild level, and a more physically oriented approach was developed. In addition, the game gives three distinct challenge-reward cycles instead of one large cycle in order to keep the student from accumulating too much challenge at one time before experiencing the reward. Also, it was determined that the game was best portrayed in two dimensions, as adding the third dimension increases complexity but does not enhance the amount or depth of material that can be portrayed.

The first stage of the program demonstrates the concept of stimulated emission of photons. In stimulated emission, photons that encounter excited electrons with the same amount of extra energy induce the electron to return to its ground state and to emit an identical photon. To demonstrate this, the photon was made to run across the screen to represent a traveling photon (see Figure 1). The photon was colored blue to give a sense of swiftness that reinforces the idea of an energetic photon. The smile on the photon was put there to give it a more energetic feel as well since the idea of energy is more associated with a positive attitude as opposed to a negative one. As for the electron, it was given the same smile to help the student perceive that the photon and electron are working towards a common goal. This communicates that electrons and photons are likely to interact, which is an important mechanic of stimulated emission.

After the elements were in place, the way in which the student would interact with this environment was determined. First, a method was devised to excite the electron. It was decided that pressing the spacebar key was the most intuitive way to make an electron jump towards the photon because many Massive Multiplayer Online Role Playing Games use the spacebar to jump virtual characters. A little bit of chance was also incorporated into the program by having the photon appear at random heights. Chance is important to making a game interesting by introducing some uncertainty [2]. However, the introduction of chance also has the potential to increase the complexity of the game for the student. This drawback was avoided by not having the student determine how high the electron is to jump. This allows for some challenge in coordination but with no added complexity in the student's interaction. The speed of the running photon was also designed so that the student's physical coordination would be challenged. In addition to dealing with the speed, the student also had to make contact with the photon at the right angle in order for the interaction to occur. These aspects were necessary to create the challenge since, by itself, pressing a key on the keyboard presents little challenge.

There was a couple of physical phenomenon that were intentionally omitted to prevent any distractions from disrupting the concentration of the student. One major phenomenon that was left out was the occurrence of spontaneous emission when the electron returns to its ground state when not being induced by a photon. The most important reason for omitting spontaneous emission was that the student would be questioning, as the game progressed, what happened to those photons that were spontaneously emitted. The student would expect to see them further along in the game, but would never find them since spontaneously emitted photons are not desired in a laser beam; therefore, this would also conflict with the theme. Whenever an unanswered question lingers in a student's mind, concentration is broken, and that is an undesirable effect for a game. In the same spirit, discrete energy levels were omitted because the jumping of electrons by appearing in one spot in one frame and somewhere else entirely in the next frame is not intuitive. It would cause the student to feel unnecessarily awkward and thus further distract the student's attention.

The second stage involved the functionality of a spatial filter, a device used to do spatial frequency manipulations on a beam of light. The lens for the spatial filter in this game is meant for a circular beam, although the one used on NIF is meant for a rectangular beam. The reason for using a circular lens was that students in their classrooms and everyday life are more familiar with the circular lens, such as those used in classroom laboratory experiments or a typical magnifying glass. Because high school students are not yet introduced to the concept of the Fourier transform and its relationship to optics, an alternate depiction was devised to be more intuitive to students. Since students are taught the basics of lens optics and ray tracing and are familiar with their general usage in everyday experience [3], the depiction of the spatial filter was created to make the lens seem much like a train switchyard. The lens would cause all beams coming in from the same angle of incidence to the plane perpendicular to the optical axis to focus at the same point. In this way, the student would be able to pass all the good photons through a hole and block off the rest of the photons (see Figure 2). To achieve this, the student would be able to move the lens using the left and right arrow keys on the keyboard, and use the up and down arrow keys to move the hole.

For this stage, there were not many things considered not intuitive since the entire depiction was changed based on the argument of being more intuitive for the student. However, one aspect of the spatial filter was omitted, and that was the other half beyond the hole where the beam gets collimated back to its original shape. The reasoning behind this was that many students during their youth probably have done the old experiment of burning a newspaper with a magnifying glass using the sun. Because of their knowledge and experience with the phenomenon of a magnifying glass making light from the sun burn something, the same configuration was used to fortify the theme of creating energetic photons. Students will see this familiar configuration as something that fits naturally into a laser amplifier.

The third and final stage involves the multi-pass system that is implemented in NIF to extract the correct amount of power from the laser glass. NIF utilizes the fact that certain crystals can change the polarization, or direction of electromagnetic oscillation, of the laser light and that a polarizer will reflect light, for example, that flows perpendicular to a certain direction of oscillation and transmits light that is parallel to that direction. The crystal used to rotate polarization requires that a voltage be applied to change the polarization of the light, which allows its use as a polarization switch. In the game, this switch is depicted by a selection box of two polarizations that are perpendicular to one another. By switching between the two polarizations at the right times, the student can make the laser pulse pass through the amplifier at least four times to extract all the energy out of the amplifier (see Figure 3). The student is able to change the polarizations by using the up and down arrow keys to select the desired polarization. Each time the laser enters the amplifier, electrons converge onto the center of the amplifier giving the impression that the amplifier energy is being sucked into the pulse, increasing the power of the laser. Finally, the laser pulse is allowed to exit, and if enough energy was absorbed, the game is completed.

A couple of compromises were made in the realization of this stage. First, the beam pulse is represented by a ball with an arrow inside it that oscillates to depict the current polarization of the laser. The reason that a beam line was not used was that there was not an obvious way to represent the polarization of the laser in that shape. An arrow bouncing around in a ball is much clearer in terms of conveying that the laser has a certain oscillation direction. The oscillation was also made to speed up with every pass through the amplifier. While the laser does not change frequency when going through the amplifier, the alternative of changing the ball’s color to a darker color to portray a denser beam presents the qualitative representation of frequency change, which could mislead the student even more than a faster oscillating arrow. This is because an oscillating arrow is a symbolic message that is not as easily absorbed by a student compared to the visual message that would impact the student more. Therefore, the misleading message was directed towards a medium that is not as likely to stay with the student.

3. Implementation

This program was designed to be embedded on the educational section of the NIF website. As such, the programming language that was chosen was Java. Java allows the creation of Java Applets that allow a program to be run straight off a webpage. Since this program involved animations, Java's capability of creating multiple threads was used to implement a convenient approach. One thread would be used to handle the applet's lifecycle and user input from the keyboard and mouse, which will be called the control thread. The other thread would be in charge of continuously running the animations and game mechanics until the program was terminated, which will be called the datapath thread. Since Java only allocates a small amount of memory for any of its programs by default, the amount of memory the applet could use at one time was limited. This made memory optimization a priority over processor speed. The problems encountered with this limited memory size will be explained shortly.

The animation of the game was the first part to be implemented. The common method of double buffering was used to prevent flashing of the animation during runtime. Also, a utility class that was named Animation was created to take an image sprite file, the number of frames in the file, and the dimension of each frame to create a linked list of frames that could be drawn onto the screen. An extra parameter of the Animation class that was later found necessary is an off-screen junk image that is used to force the frames to be cached by drawing onto this image. Otherwise, the frames would be referenced but not cached, which would degrade the performance of the animation during its first execution. By using this utility class, the datapath could easily advance the animation and retrieve the current animation frame to be drawn onto the screen. Initially, all animations were instantiated at the initialization of the applet. For most of the animations, this only caused a slight load time of a couple of seconds before the game began. However, when the transition animations were introduced, the memory allocation of the animations failed due to lack of memory to store the frames.

To avoid this memory overflow, it was decided to take advantage of the fact that no interaction took place during the transition. However, the frames could not be stored into a list in the program to reduce the memory footprint. Instead, an animated Graphics Interchange Format (GIF) image was used to have the image itself manage the framing. An animated GIF does not allow for the frame manipulation that was required for the implementation of the stages, but since no interaction occurs during the transition animations, this drawback was not a hindrance. The use of animated GIFs meant that no preloading was necessary, and the product was a smooth animation with little memory usage.

Since the animation was done using frame drawing, a method of ensuring constant frames per second was necessary to create a smooth animation. To do this, a timestamp was taken at the beginning of the animation loop. All the mechanics and image drawing occurred afterwards. At the end of the loop, another timestamp was taken. The datapath thread was then instructed to wait for the difference in milliseconds if the difference came out positive. If the difference came out negative, the thread would continue its loop without the wait. Although this only ensures that the frame rate of an animation would be bounded above by the inverse of the delay time, any speed slower than that would be caused by the inability of the student's computer to execute the datapath instructions fast enough. This factor cannot be improved much more as the variety of computers currently being used is too large to be able to accommodate all of them. However, one small optimization was made by eliminating the conditional that checks for a negative difference in the timestamps and replacing it with a mathematical statement that takes the difference and compares it to 0, taking the maximum of the two values. Therefore, if the difference is negative, the thread will wait for 0 milliseconds. This optimization works because a failed branch prediction on a conditional causes the processor pipeline to be emptied, which results in wasted clock cycles.   Therefore, minimizing conditionals that occur frequently such as this sign check causes fewer clock cycles to be wasted. A frame rate of 30 frames per second was used in the animation of this program.

One optimization that does not lie in computer performance is the use of image sprites instead of folders containing images of the frames. Because the program is meant to be run from a browser, every file must eventually be downloaded by the user through a Hypertext Transfer Protocol (HTTP) connection. If a frame folder were used to implement this program, the student's computer would have to create a new HTTP connection for each and every frame. That causes inefficiency in the download time as it takes time to establish each of these connections. Instead, the frames are merged into one large image, which can then be divided up into frames by the program. This means that the student's computer only has to establish one connection for each animation compared to the possible hundred that a frame folder would require.

Modularity was also a priority in this program as the design of the game allows the insertion of more stages and animations in later revisions. To achieve the most modularity as possible, several things were done. The animations that required no interaction with the user were drawn as full-sized frames drawn on the (0,0) coordinate of the screen with all objects and background included. Therefore, if there were to be any change in these animations, the only change that would need to be made is the replacement of the animated GIF file. Also, each utility class was made to be self-contained and require no other classes beyond the ones provided in the Java library with only one exception. The one exception is called the MovableObject class. Since many of the objects in the animation change position and direction, this class was made to use the Direction and Location utility class that was implemented in this project to function as an object that keeps track of its own location and direction as well as its own animations. This made moving and animating objects convenient and made the source code much easier to comprehend.

Lastly, the scoring system was implemented using mathematical criteria that fit the type of challenge each stage provided. Each score system tried to always allow the student room to improve which encourages the student to play again. The first stage had photons running across the screen while the student attempted to jump an electron to make contact. Limiting of the number of photons was not an option, as it would needlessly frustrate students who take a longer time to calibrate their coordination. Therefore, the choice was to keep track of the number of photons that were successfully duplicated and the number of total photons that have been generated. Using these two numbers, the ratio of success to total is created to measure the efficiency of the student's performance. The second stage required the alignment of the lens and hole to allow all the good photons to pass through. Therefore, the scoring was based upon how fast the student could align the beams. The last stage is the most complicated of the three because the program attempts to approximate how much energy is extracted from each pass. The first four passes follow an exponential curve to cause more energy to be extracted at first and less at each successive pass until all of the energy has been extracted. After the first four passes, in order to encourage completion in exactly four passes, the efficiency is calculated by taking the inverse of the number of extra passes, causing the four-run solution to be the only solution receiving 100% efficiency.

4. Future Work

Not enough time was given to get feedback on this program. However, some ideas for getting feedback include the creation of focus groups. One possible method could be to get feedback from adults with no significant scientific background beyond high school, who are more readily accessible, and then moving on to high school students. Feedback from high school teachers could also prove very valuable in evaluating this program's performance. With these responses, the game could be changed and enhanced to more readily suit high school students with whom this game attempts to communicate.

Other more quantitative feedback methods could be integrated into the program as well. Things such as the distribution of completion times and the average number of keystrokes per stage could be measured and stored in a file. Such data could indicate how difficult a stage is or how frustrating certain stages are to the students.

5. Acknowledgements

This project was conducted at Lawrence Livermore National Laboratory under the National Ignition Facility and Photon Science Directorate. I would like to take this opportunity to thank my mentor, Jill Garland, for her guidance and support. I would also like to thank Tom Reason for creating the animations and sound effects as well as Clifford Widmayer for providing the technical details of the main amplifier. Additionally, I would like to thank the Department of Energy, the Office of Science, and the Science Undergraduate Laboratory Internship program for giving me the opportunity to participate in this rewarding internship.

6. References

[1] A. Kay, "Doing With Images Makes Symbols," *The Distinguished Lecture Series: Industry Leaders in Computer Science*, [Videorecording]. Apple Computer Inc., 1987.

[2] J. Schell, *The Art of Game Design: A Book of Lenses*. Burlington: Elsevier, 2008.

[3] California State Board of Education, "Science Content Standards for California Public Schools: Kindergarten Through Grade Twelve," *Content Standards*, 1998. [Online]. Available: http://www.cde.ca.gov/be/st/ss/documents/sciencestnd.pdf. [Accessed: June 10, 2009].

Figures

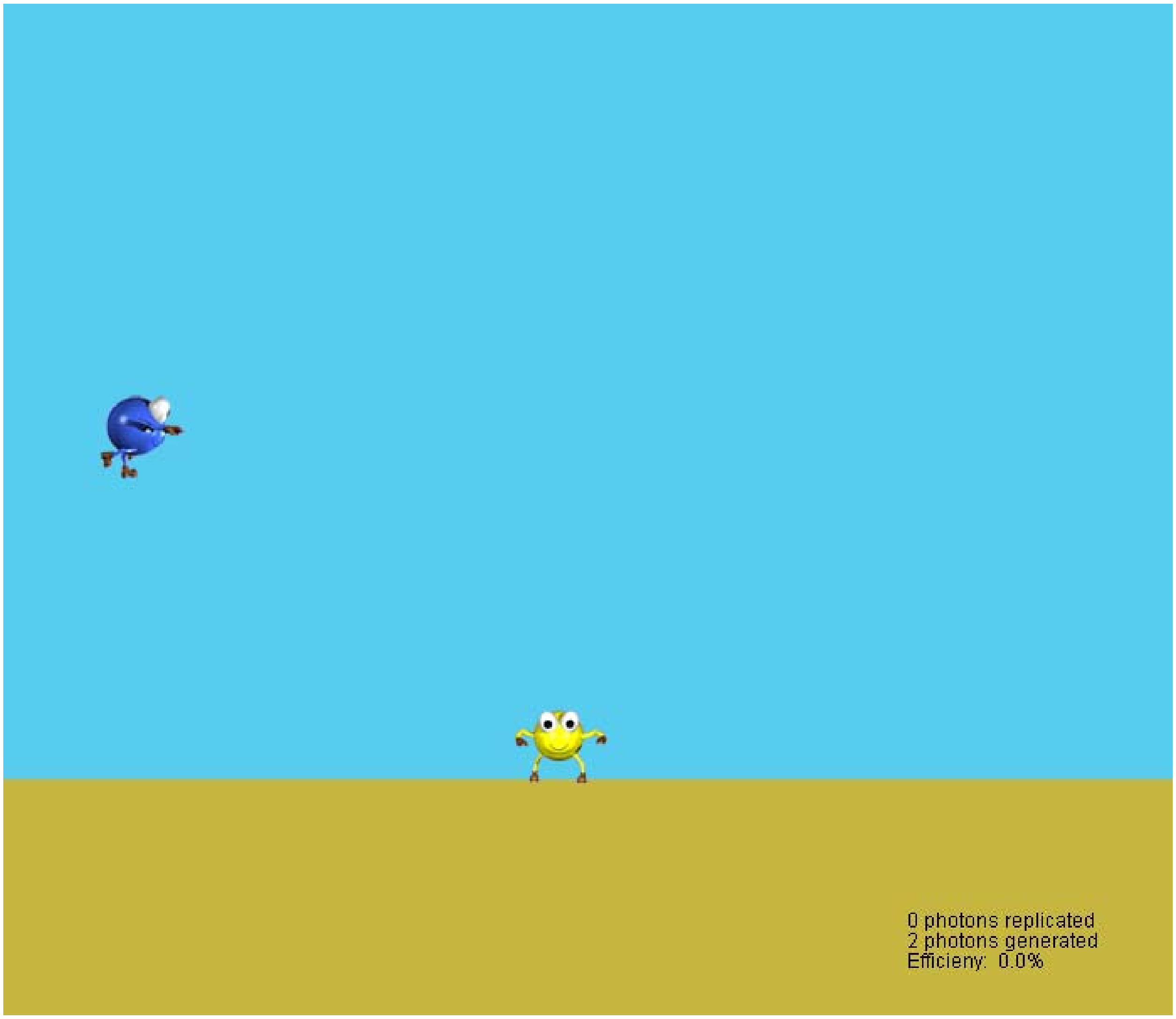

Figure 1: The first of the stage of the program during rapid prototyping. The photon runs across the screen, and the electron waits to jump to make contact with the photon.

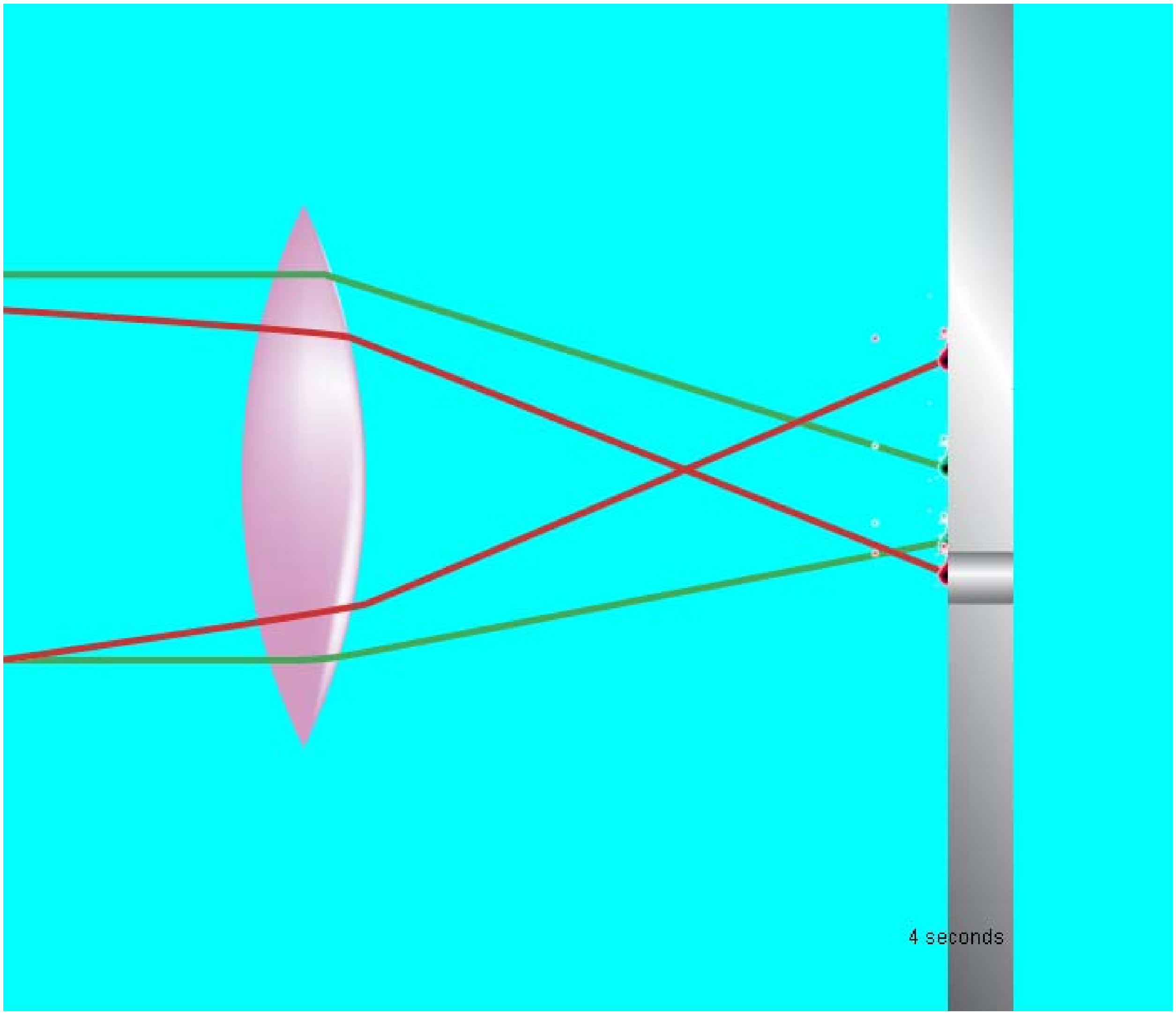

Figure 2: The second stage of the program during rapid prototyping. The lens is serving as a switchyard for the different beam angles.

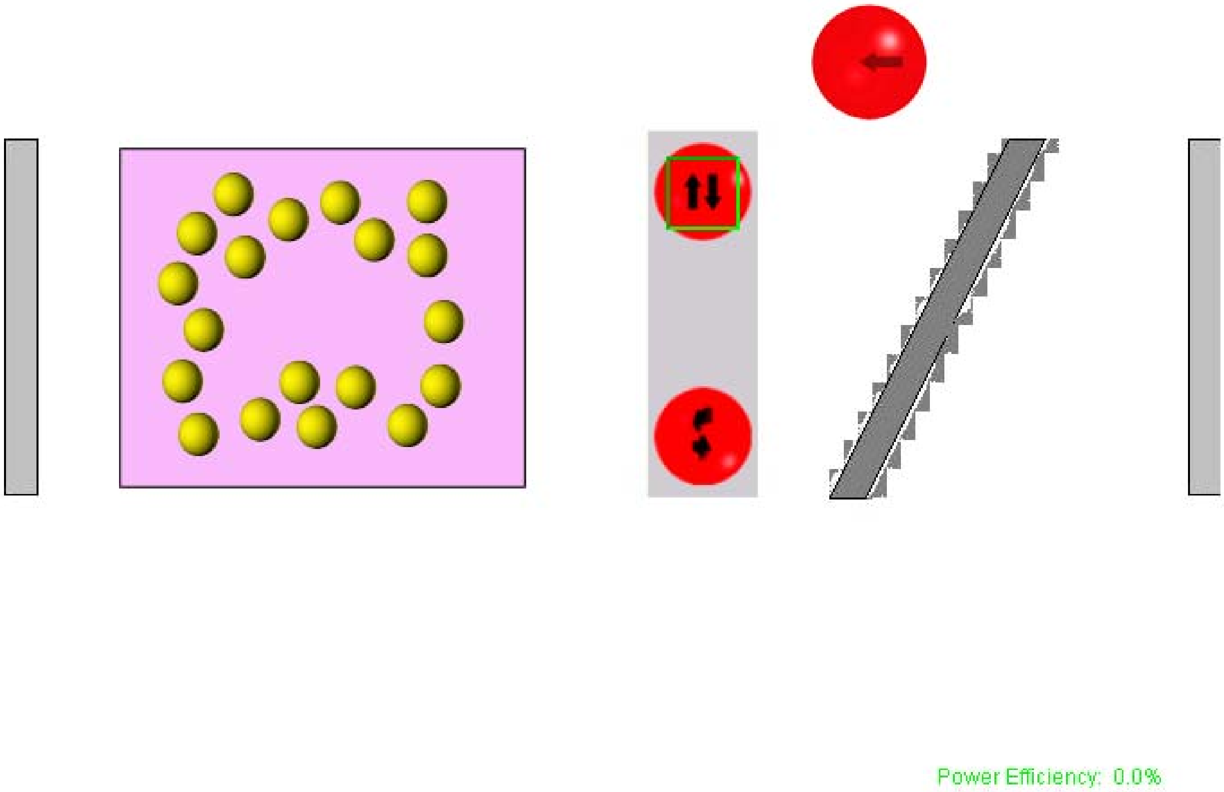

Figure 3: The last stage of the program during rapid prototyping. The ball representing a beam of light is polarized in one direction that can be changed using the gray selection box.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.