Visualising the life cycle of stars*

Levi Peeters and Nikki Rademaker

LIACS, Leiden University, Niels Bohrweg 1, Leiden, The Netherlands
j.c.c.peeters@umail.leidenuniv.nl, n.s.rademaker@umail.leidenuniv.nl

Supervised by Lisa Pothoven and Fons Verbeek
lisa.pothoven@xs4all.nl, f.j.verbeek@liacs.leidenuniv.nl

Abstract. Astronomy students are required to learn the evolutionary cycle of stars. This is a difficult task since the process knows many scenarios, and because study material for this subject tends to be limited. Therefore, an interactive program is created for the visualisation of the life cycle of stars. Visual study material can often be useful. We did research on whether or not the created program will help understanding such life cycles. In total, four types of stars are implemented in the program, where each life cycle is divided into a few phases. To measure the usability of the program, we divided them into categories (interface usability, suitability, educational value, and satisfaction). We held two user evaluations to measure the usability, this resulted in a high System Usability Scale score. Additionally, multiple astronomy students reported that this program could help them understanding the life cycle of stars if it were improved in a few ways.

1 Introduction

Introduction to Stellar Evolution

Astronomy is the study of everything that exists in space. One of the most common objects we encounter in space are stars. They come in many forms and sizes; ranging from small red dwarfs that burn for longer than the universe's current age to huge supergiants that go out in a glorious supernova. It will come of no surprise that the physics that underlie the evolution of stars are rather complicated, but to introduce the reader to this subject we will give a short explanation of the general evolution of a star.

All stars form when a cloud of gas, which is mostly Hydrogen, contracts due to gravity. When the gas gets more and more compact, the temperature and pressure rise until they are sufficiently high that in the core, Hydrogen can start to fuse into Helium. Gravity continues to try and pull the gas cloud closer together, but the fusion reaction balances this force with its outward pressure. The resulting balance is what we call a star. Stars spend the majority of their lives burning Hydrogen to Helium, which will start to build up in the core of

* Supported by Leiden University.

the star. This phase is called the Main Sequence, and it is where our sun is in its evolutionary cycle. Low-mass stars like red dwarfs will never go beyond this stage, and burn Hydrogen until they go out.
If a star is more massive, such as our sun, the life cycle becomes more interesting. As the star starts to run out of Hydrogen, the pressure of the fusion reactions drops and gravity will start to win the power struggle in the core, contracting it further. This will again increase the temperature and pressure, until it becomes so high that suddenly, Helium can also undergo its reactions into Carbon and Oxygen. This sudden burst of energy causes the reaction pressure to overpower gravity easily, and the star will expand into a red giant. This will be the eventual fate of the sun.
For stars that are even more massive, this process happens again when Helium starts running out, and Carbon and Oxygen will ignite to react to Neon. Then again to Magnesium, then to Silicon, and finally Iron. The more massive the star, the more reactions it will 'unlock' as it ages. Iron is the final phase, as it will never undergo fusion and it will continue to build up until the star dies.

Goals & Research Question

As an astronomy student, you will start to learn about this process in your second year. There is a lot of math and physics involved in the evolution of a star, and learning to calculate it all precisely is quite a daunting task. Visual study material, such as images or animations, can be a helpful tool to learn. However, study material for this subject often falls short as there are many different scenarios to visualise, and a lot of material focuses on a single scenario. In addition, many visualisations fail to show information such as mass, radius and accurate colour, leaving a student to guess what is accurate and what is not. We believe that a single interactive animation that covers different scenarios, and is clear about what data it uses and its approximations, will be very valuable to the aspiring astronomer. This led us to formulate the following research question, which has been slightly revised since the project plan:

To what measure will an interactive animation help users understand what reactions take place in a specific layer of an evolving star, and help them understand what the life cycle of the star looks like?

Section 2 will explore and define this problem with some more depth. In section 3, we will present our approach to solving this problem and our final product. Section 4 will present the results we have achieved with this solution, using the feedback from user evaluations. Section 5 will discuss how effective our solution is, and what should be improved. Moreover, a final conclusion will be presented. Finally, in section 6 we will discuss how this solution could be used and expanded in the future.

2 Struggles of an Astronomy Student

2.1 The Problem

As we introduced in section 1, stellar evolution is a complicated process that takes a lot of effort to learn. We believe that educative visualisations in this area are lacking in clearness, completeness and accuracy. An example image that students might use, is shown in figure 2. This figure shows the layers of reactions in the core correctly, but it has a lot of problems. For one, the figure shows no data of the star, such as mass or age. This leaves the student to guess or find out some other way what type of star will actually reach this state. In addition, there is no indication of how inaccurate this image is. A student will not know if the outer colour is accurate. In addition, the inner colours imply decreasing temperature towards the core, which is not accurate.

2.2 User Analysis

The main target users for this created program are astronomy students. The discussed inaccuracies are mostly clear to an experienced astronomer, but they are big hindrances to a student who is trying to learn. It would be very helpful for them to have images like this and be able to link them to stellar parameters like mass, age and temperature. Students will often be tasked to use complicated physics to calculate these stellar parameters. Therefore, they would benefit from an interface that accepts these parameters, that shows them what the star looks like, and that shows them how the star will evolve.

3 Interactive Animations support the Learning Process

Design philosophy

To tackle these issues, we decided to build an interactive animation that visualises this process for many different scenarios. The primary stellar parameters that we use as variables, are stellar mass and age. This is because these parameters have the largest impact on the structure of the star. For a star of a given mass, the evolution of their brightness and temperatures are fairly well understood. We used data from Iben (1967) [1] to find these parameters, as well as the core structure for a few stellar masses. Then, the radius and the colour of the star can be computed with relative ease.
For our user group, it is important that there are different scenario's combined into one interface. We have currently implemented four different star masses: A small dwarf star, a sun-like star, a massive star and a supermassive star. Adding more intermediate masses would be helpful, but it is not a main priority as most stars can be divided into one of these four categories fairly well. Instead, our priority was to show the life cycle of these four types as accurately as we can, while making it clear to the user where the approximations and uncertainties in the data are.

The life cycle is divided into a few phases. In a lot of representations, these phases are decided based on what the star looks like (e.g. 'red dwarf' or 'blue supergiant'), but these looks are only a consequence of what is really going on; the nuclear fusion reactions happening in the core. That is why we have divided our phases differently, where a new phase starts as soon as a new reaction is unlocked in the core. A red dwarf will only experience one phase, as it never unlocks Helium fusion, whereas a supergiant will go through all six phases. These phases are shown on the timeline, and it's made clear when the current star is unable to reach a phase.
To make sure users know what the star looks like at each point in its evolution, multiple reference objects are shown to compare the size of the star. We assume some astronomical knowledge in our users. For example, when they see a star the size of Jupiter, they'll know it is a dwarf. Colour is also an important part of the looks of a star. Therefore, it is calculated dynamically from the surface temperature of the star and updated each frame.
Another important part of our design philosophy is that we make it very clear where the approximations are in the data and computations. For students, this is important so that they will know how to interpret what they see. We made sure to be very clear where we were being inaccurate and where we were unsure if a situation reflected reality to prevent confusions.
To support the student's ability to remember what they see, we chose our colours to make sense with the data. The colour of the star reflects its actual colour in reality, but we were free to choose the colours of the layers. We decided to use a colour scale that reflects increasing temperature towards the core; blue for the outer layer, to deep red in the core. The buttons on the timeline share these colours as well; each button has the colour of the deepest layer that appears.

Usability specifications

Keeping all the above in mind, and the requirement that the interface itself should not be confusing to use, we defined our usability specifications:

- *Interface Usability*: Specifying how the user navigates the interface and searches for a desired result.
- *Suitability*: How well this type of visualisation suits the data.
- *Educational Value*: How well this visualisation helps a user learn the underlying concepts.
- *Satisfaction*: Whether the user is satisfied with the overall experience.

We tested interface usability by asking a user, not necessarily with an astronomy background, to complete a number of tasks. We measured how long it took them, and how many errors they made. Users were also asked for their opinion through the System Usability Scale [3]. Suitability, Educational Value and Satisfaction were tested by asking astronomy students to use the interface and recording their opinion through a questionnaire. Results will be discussed in the next section, for more detail consult our user evaluation documents.

4 Design process & Resulting Prototype

The interface we were able to build, provided most of the functions we envisioned. It shows four different mass stars which can be set to a specific phase, or go through an animation of the entire evolution. Its radius, colour, core structure and age are shown as they change, and we have made clear where the uncertainties and inaccuracies are to avoid confusion. In order to test whether users experienced these functions in the way we intended, we have held two user evaluations. In this section, the sessions will be described shortly and their results will be presented.

The first evaluation session was held in the beginning of the prototype building stage. The interface only supported one star and the animation repeated the same four frames. The goal of this user evaluation was to get feedback on our method of presenting the data, and to get suggestions for features we could add. The target users for this evaluation were astronomy students, of whom we reached about twenty.
From this session we learned that the users were mostly happy with the basic structure of the data, but as we expected they needed much more clarity as to what was happening on the screen. We discussed this and found that adding tooltips for the interface objects would be an effective way to accomplish this. Tooltips enable a user to hover over an object in the interface with a computer mouse, to gain more information on the functions. We had expected that we needed more clarification for the star itself, but were surprised to hear that users also struggled to find the interface items, such as the button to open/close the star. Users also noted that the program would really benefit from reference objects to show changing stellar radius and to change the colours of the core layers so that they better reflect increasing temperature towards the core.

After a few weeks of developing, we had implemented most of the feedback from the first session. The animations were smoother, reference objects showed the star scaling, colours were more representative, and tooltips clarified parts or the interface. We were aware that information was still a bit scarce, we had not gotten around to adding explanations of inaccuracies and approximations and there were still some bugs in the timeline functionality. Not ideal for a final testing session, but as time began to run short, we held a second evaluation at this stage, since all main interactions were already possible.

The second evaluation consisted of two parts. We were not able to reach astronomy students directly due to their exam period, but we did reach some online that were willing to evaluate our data presentation in a way similar to the first session. In addition, we tested functionality of the interface with a second group of evaluators, by giving them a list of tasks and observing how long it took them to complete them and how many errors they made along the way.
From the usability evaluation we learned where the users struggle when using our interface. When we asked the users to score our interface using a System

Usability Score [3], they were very positive overall, and gave the program an above average score (88.2 where 100 is the highest possible score). When we asked the users to perform some tasks, and measured the operation speed and the number of errors they made, we also found that the interface satisfies our initial requirements.
Additionally, when we asked this testing group for suggestions on how to improve the usability, we found that there were still improvements to be made. Although most functions were clear, users reported that they found the start/stop/pause buttons that we had set up to be confusing, and they would prefer a pause/play single button. Users also took some time to realise that using tooltips provided more information on the layers within the star, and found that the tooltips were too short and lacking in information.
From the second user group, the astronomy students we queried online, we also gained some valuable information. The main message from this group was that they saw potential in our interface as a learning tool, but that we needed to make some changes to realise this potential. These users were also confused about the tooltips, requesting more information on them, and suggested that a general information panel before starting the program might also benefit clarity. We could also present more information on the different types of stars, for example by explicitly stating that selecting low mass will show you the evolution of a red dwarf. As we expected, the astronomy users also pressed that we needed to implement a way to show how much time passes in each phase. This was already planned, but the evaluation gave some insight in how to do it, and made it clear that it was a high priority feature.
The second user evaluation allowed us to measure the target scores we set for the usability attributes of our usability specification. As shown in figure 3, all values satisfied our goals.

In the final version of the product, we were able to address most of these issues. Tooltips in the star itself are indicated by an "i" symbol to show a user that they can gain more information there (see figure 5). Tooltips and other text bits are now more descriptive, and an added information panel tells the user some basic information about the star they are viewing, including a description of some inaccuracies and the evolutionary parts of the star we do not show as is shown in figure 7. The buttons that control the animation were also changed, there is now a play button that turns into a pause button when the animation is running, and a separate stop button that will reset the animation completely. A way to display the age of the star is now implemented as well. The total age of the star is presented at the top of the screen, increasing as the animation plays out. In addition, each interval on the timeline shows how long the star will take to go through this phase. These intervals were added because some intervals are very short (some shorter than a single year), which means they do not show up on the total age counter which counts millions or even billions of years. To avoid this becoming confusing, the interval ages were shown separately.
As a final change, we added an introduction screen that explains what the in-

terface is for and what the user will be able to do, which is shown in figure 4.

5 Discussion and Conclusion

5.1 Difficulties & Insights

One of the main challenges for this project, is that the physics underlying stellar evolution are quite complicated. We knew we would be able to make a lot of simplifications and assumptions, but it proved difficult to decide just how much physics to do. We settled on using data from Iben, I [1], from which we calculated radius and colour ourselves. Radius seems to be accurate enough, but the colour of the star seems to be off for some situations. We made sure to mention this in the interface, but a better way might exist.
Designing our timeline in a comprehensive way also turned out to be challenging. Each star is different, and reaches a different amount of phases, but implementing a new timeline for each star was difficult to do neatly in Unity and turned out very chaotic. We opted for a single timeline instead, which gave us a different issue; how do we differentiate between a state that is not reachable and a state that is in the star's future? Of course, unreachable states would not be clickable, but a user may get confused when a button is inactive without reason or disappears completely. To clarify this, we added a tooltip explaining when a star will not reach this stage when the user hovers over its button, and made it so the time intervals towards unreachable stages said "N/A", widely accepted to mean "not available". We wanted to make the distinction even more clear by giving the buttons of unreachable states a different sprite, for example by crossing it out, but were too short on time to implement this. Still, we think it is sufficiently clear now.
An issue we were not as well prepared for, is that is was hard to guide a users attention to the parts of the interface we wanted. We discovered this issue in the second evaluation, where some users thought the reference object (which is the object that moves/scales) was the center of attention, instead of the actual star. Because the star gets so much bigger, we opted to keep the star itself at constant size and rescale the reference objects instead. While this is mechanically a convenient solution, it draws the user's attention towards the changing object instead of to the star that we are visualising. This effect was mitigated slightly by moving the camera so that the target star is in the center of focus, putting the reference objects to the side, but the fact that the reference objects are changing still draws much attention towards them. Another possible change could be to use a ruler for scale instead of full objects.
This issue of diverting attention also arose when attempting to explain the parts of the interface, where users would not intuitively hover over objects to view the tooltips. This became a lot better by adding an introductory window explaining this possibility. We also added hovering spots with an 'i' mark (for 'information'), but had no time to test the effectiveness of this solution.

5.2 Conclusion

As discussed in section 4, two user evaluations were held. From here, we concluded that the interface satisfies the usability specification goals that were set. Additionally, the overall feedback we received from the evaluators was positive. The suggestions that were made to improve the program were very helpful.
Coming back to the research question that was stated in section 1, we concluded that the created program could help astronomy students with understanding and memorising the life cycle of stars. However, for it to be an actual learning tool, more features and information should be added (section 6). Moreover, a lot of approximations were made in the data. This means that the program can not be a substitute for textbooks, but it could be an additional learning tool.

6 Future Work

In our design process, we have had a lot of good ideas, as well as suggestions from peers, to expand on this interface. In this section, we will discuss a few of ideas that we liked, but that we were unable to implement.

The stellar evolution cycle that our interface is built for, is most commonly represented in a so-called Hertzprung-Russell diagram, or HR-diagram for short. This is a graph that plots stellar luminosity against surface temperature, and astronomers have found that stars follow a very well-defined path along the HR-diagram as they age. See figure 1 for an example of an HR-diagram that contains the path the sun will take in its evolutionary cycle. Learning how to use an HR-diagram is another important part of studying stellar evolution, and it would be a great addition to our interface to be able to show this diagram as the star ages.

The different types of stars we have discussed will also end their life in wildly different ways. Small stars will, over extremely long time spans, run out of fuel and cool down until they are invisible. Sun-like stars go through a planetary nebula phase and turn into white dwarfs. Massive stars are the most interesting, going out in a massive explosions known as supernovae and leaving neutron stars or black holes behind. These end results are also a part of learning stellar evolution, and showing what happens at the end of a star in our interface would be a nice expansion.
While it is less interesting than the death of a star, the birth of one could also be added. The interface could show how a gas cloud of a certain mass contracts and how the reaction first ignites, although this has never been observed and would be very uncertain.

A very ambitious addition would be to make the underlying data more accurate. This would require complicated physical equations such as the Virial Theorem, but could be used to calculate the parameters, instead of interpolate them from a few data points. This would be a lot of work to program and it is questionable whether the increase in accuracy would be worth the effort.

References

1. Iben, I., "Stellar Evolution Within and off the Main Sequence", *Annual Review of Astronomy and Astrophysics*, vol. 5, p. 571, 1967. doi:10.1146/annurev.aa.05.090167.003035.
2. https://commons.wikimedia.org/wiki/File:Evolution_of_the_Sun_2_EN.svg. Reference date 16-12-2021.
3. Brooke, J.: SUS-a quick and dirty usability scale. Usability Eval. Ind. 189(194), 4–7, 1996

Appendix

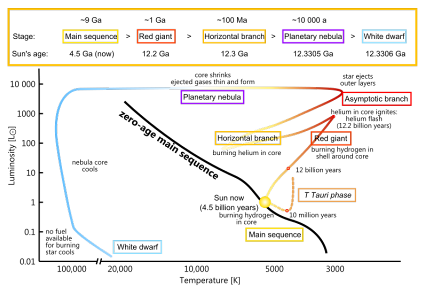

Fig. 1. The evolution of the Sun on the Hertzsprung-Russell diagram. Our interface starts on the zero-age main sequence, and runs until the sun reaches the asymptotic branch. CC BY-SA 4.0 [2]

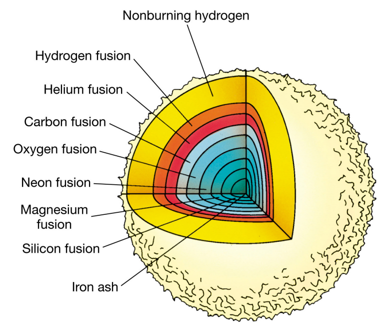

Fig. 2. An example image of a massive star near the end of it's lifecycle, as may be used to illustrate the stellar evolution cycle. Copyright 2005 Pearson Prentice Hall, inc.

Usability attribute	Measurement	Value to measure	Min	Goal	Max	Measured
Operation speed	Navigate to specific time	Time (s)	6	3	1	1.3
Operation speed	Change mass of star	Time (s)	8	4	1	2.8
Error tolerance	Navigate to specific time	Number of errors	2	1	0	1
Error tolerance	Change mass of star	Number of errors	2	1	0	0
Suitability	Questionnaire	Average score	5.5	8	10	8.2
Satisfaction	Questionnaire	Average score	5.5	8	10	8
Educational value	Questionnaire	Average score	5.5	8	10	8.2
Learnability	Questionnaire	Average score	5.5	8	10	8.2

Fig. 3. Usability specification with the lowest acceptable scores (Min), the best possible scores (Max), the target scores (Goal), and the measured values for these attributes.

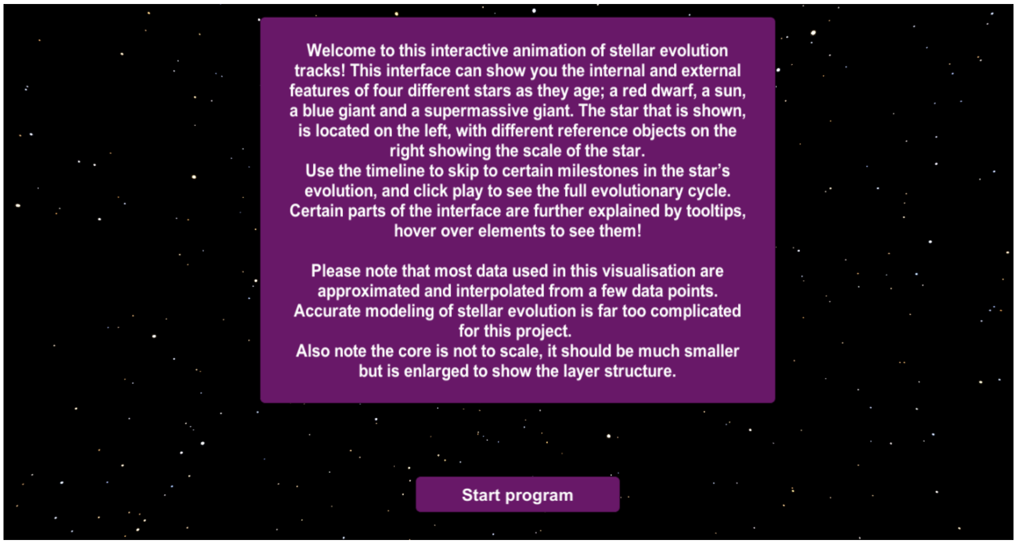

Fig. 4. The introduction screen of the created program.

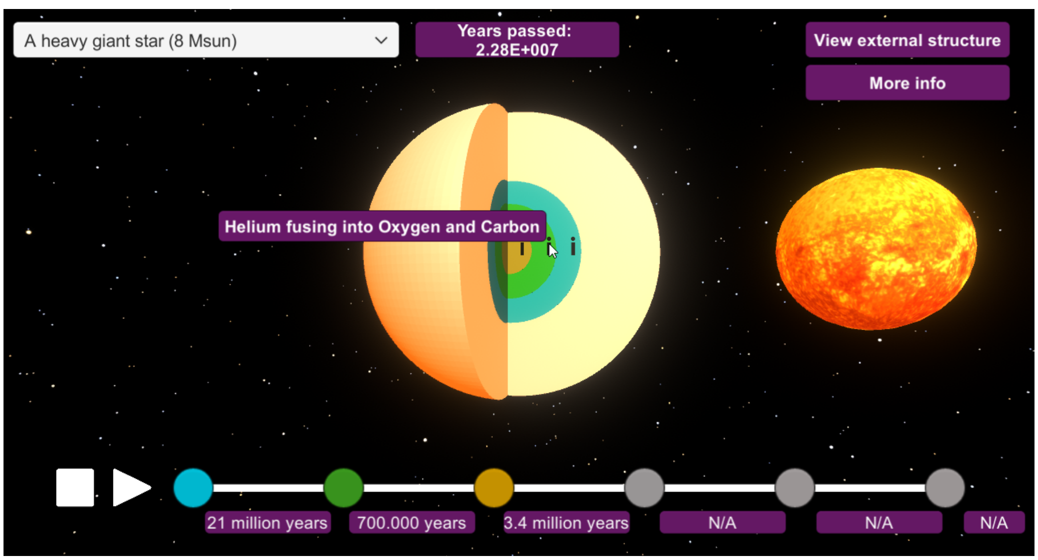

Fig. 5. The interface for a heavy giant star during the third phase of the life cycle. By hovering over the i icons on the layers, the user can get information on the fusion reaction that takes place.

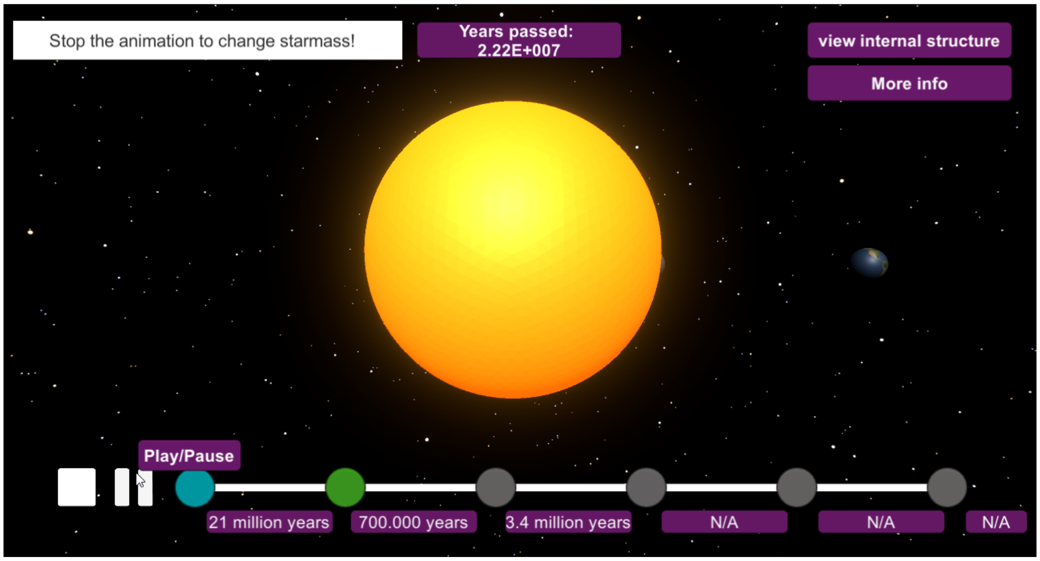

Fig. 6. The interface for a heavy giant star during the second phase of the life cycle (while the animation is playing).

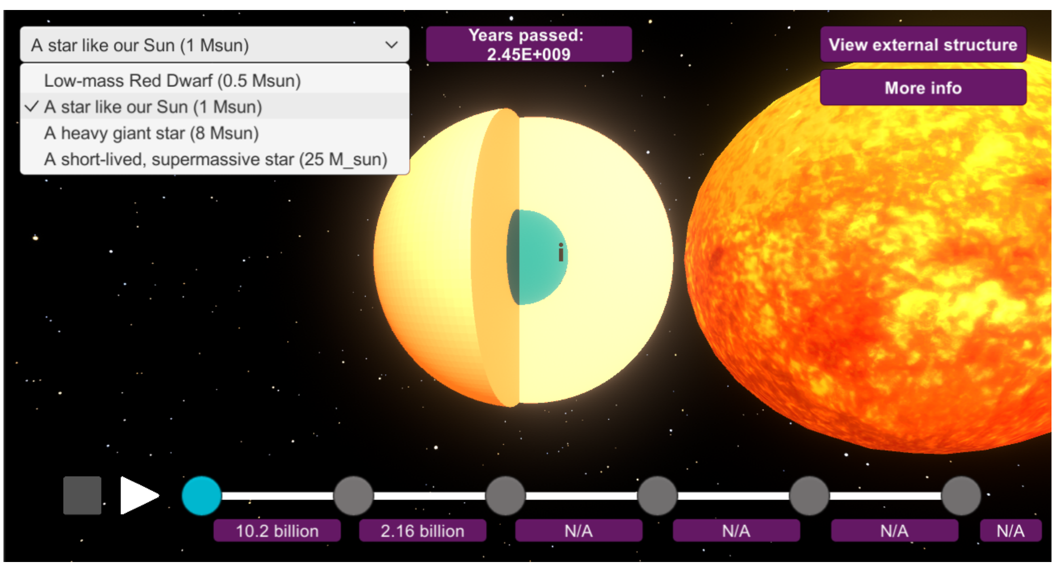

Fig. 7. The interface for a star like our sun during the first phase of the life cycle (while the animation was stopped). By clicking on the drop down menu, the users can change the type of star they want to look at.