Acoplamento de Proteínas em Ambiente Imersivo

João Miguel Santos
joao.b.dos.santos@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

Outubro 2017

Abstract

In molecular biology there are several challenging problems, in specific, a very complex problem which the name is ”protein docking”. This problem tries to understand how two proteins interact with each other, this makes it possible to understand if they can be docked to each other while still remaining functional. In order to solve this challenge there are several manual solutions, however, recently none of them has a good manipulation technique, since they are based on devices that can only function in 2D space. This work presents a solution that allows professionals and scientific community to perform protein docking in a more natural and intuitive way, in order to do this we use the virtual reality to achieve a superior visualization of the proteins and also to be able to implement more intuitive manipulation techniques. The results we obtained were quite encouraging, as all users were quite satisfied with the experience of doing protein docking in a more efficient way than the current solutions offers.
Keywords: Docking, Proteins, Virtual Reality, Interaction

1. Introduction

Proteins are macromolecules which are present in all living beings, these are constituted by amino acid chains. An amino acid consists of an amine group (which are composed of one nitrogen atom and two of hydrogen), a carboxylic group (which are compounds by one carbon atom and two by oxygen) and one R group. All of them have always an amine and carboxylic group however the R group is different from amino acid to amino acid and causes different combinations that generate different types of amino acids.

There are a number of problems in molecular biology, but two of the most important are protein docking and protein folding. The first problem is about how two proteins interact with each other so that it is possible to dock them with each other and the resulting protein keeps functional. This is very important because more and more new proteins are known and how they interact with each other is essential to design drugs to cure certain types of diseases. The second problem has to do with perceiving and discovering how the protein is in nature, so that it is understood in what form it is functional, the amino acids need to be in a specific position within the protein so that it works perfectly, so this problem consists of changing amino acid positions to try to find the right way for the protein to work.

In this project we addressed the problem of protein docking with the aim of improving the resolution process in order to assist professionals and scientists in the biology field. So far there are two approaches to this problem, one approach is automatic and the other is manual. Automatic solutions rely on algorithms that usually try to couple two proteins in every possible way to find the solution, so they have to test many cases, and the larger the protein complexity, the longer it takes time to solve. Manual solutions are those that imply that a professional has to try to solve the problem with the aid of the software, these recent solutions are not the best because they do not yet take advantage of the new technologies like virtual reality, which consequently make the manipulation of proteins less practical and accurate.

As mentioned previously we present a solution to help molecular biology professionals solve the protein docking problem. With this idea in mind we developed a game using virtual reality, in order to be able to implement manipulation techniques that are more intuitive and to achieve a superior visualization relative to what exists today.

2. Related Work

Proteins[7] are macromolecules where the various biological functions are performed. They are part of the structure of all the cells of our body. As an example we have the figure 1 representing a small protein.

The problem of protein docking [5] is a geometric search problem where the degrees of freedom

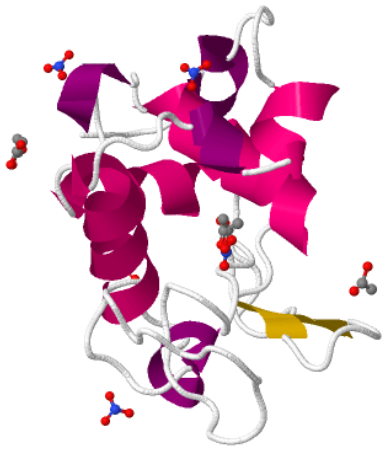

Figure 1: Atomic resolution of a protein called lysozyme

to be considered are the relative orientation of the two molecules as well as their conformations. This problem is very important because understanding how proteins interact can allow, for example, to develop and design drugs to cure diseases. The figure 2 shows how the process of docking two proteins is done in a simplistic point of view.

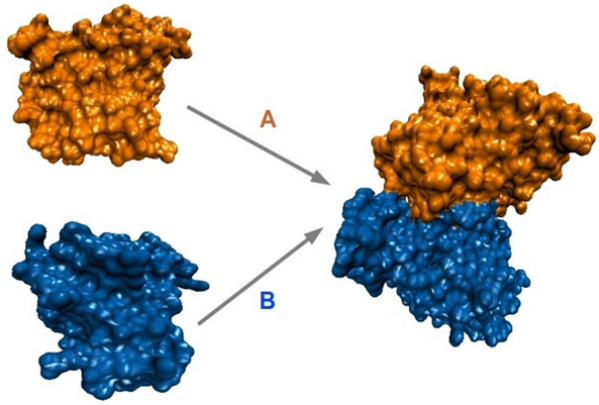

Figure 2: Protein docking between A and B to form a new rigid body [9].

In order to solve this problem there are automatic and manual approaches. Automatic approaches use algorithms that, by trial and error, try to find a solution with minimal energy. As an example of these algorithms we have the HADDOCK [4], ZDOCK [17] and ClusPro [3]. According to Ehrlich [5] the manual approach is one in which the researcher has to make the docking of the two proteins interactively and determine the chemical and geometric adjustment by means of computer graphics tools. Preliminary structures of manually generated complexes can be refined to remove bad contacts by minimizing energy. Manual docking is usually done using experimental data on the effects of mutations as guides to generate plausible structures of complexes.

Solutions such as STALK[14], VRDD[1] and DockPro [2] are examples of automatic approaches to address the protein docking problem. These three projects came up with the idea of using virtual reality to make the protein docking an experience more efficient and overall better than using the traditional mouse and monitor. However, in the year of 1997, which is the year of project STALK[14], there were still no powerful and portable devices that could make the user feel that he was part of the virtual world. The devices at that time were large and they needed projectors to send the image to a surface and the user had to wear glasses to see the 3D effect, also the way to manipulate, although it was better than the 2D mouse, was still used a mouse 3D however the DockPro [2] introduced a much more intuitive way of manipulating proteins, without using any kind of mouse, using a magnetism system to be able to track the user's hands.

Knowledge about the structure of protein-protein complexity allows scientists to better understand the key mechanisms in interaction between proteins. This is an important scientific bottleneck, both in theoretical terms (understanding of protein functions) and applied research (specific inhibition of protein functions for drug design). Current methods for protein-protein docking include automated phases, which take into consideration protein topology as well as energy (ie, physicochemical) properties and phases for molecular visualization, allowing for evaluation of the results.

The automatic phase is expensive in terms of processing time, and it produces a large number of docking configurations that can not be separated using only objective and automatic parameters. Therefore, a manual phase analysis is also required, requesting the visualization of the results of the algorithms by a specialist.

However, this visual analysis requires large amounts of information to be processed simultaneously by the docking specialist including manipulation of three-dimensional objects, physicochemical data, biological data and so on.

In the project Combination of Sensorimotor Renderings for the Immersive Analysis of Results [6], the hypothesis is that the use of virtual reality technologies and related interactions, which depend on multiple sensory and motor channels, help the professional to do the docking process.

The next solutions addressed the concept of gamification in the context of the protein docking problem. According to Huang [10] and G.

Kiryakova [12], gamification is a technique that uses the challenges and the fun component of a game to the learning of concepts in order to motivate the users to learn. This process is used, in most cases, to motivate an audience which is not so much interested in a certain subject to learn. The Udock [13], BioBlox [15] and ChremPreview [21] games try to apply this concept of gamifying a problem to reach a larger number of users and at the same time help the scientific community to solve scientific problems.

The Udock [13] and BioBlox [15] projects are very similar games in that the objective is to do protein docking in order to arrive at a result that they consider the best to pass level. The Udock [13] interface is much simpler than BioBlox pace[15] and allows for fewer operations, the latter being more suitable for users who are not as knowledgeable at the level of Atoms and concepts of molecular biology in general, given that it presents much more information as type of atoms and their charge. The characteristic of the two that are the same is the fact that the manipulation technique is quite limiting, since the user can only perform two types of manipulation, rotation about the protein axis or grab two hooks to atoms, one of each protein, to be able to move them. This causes the user to be frustrated as the accuracy of this manipulation is poor.

The game ChremPreview [21], although not made for protein docking, had the idea of creating a system that could be used in the hands of the user. It used augmented reality so that the user could create or draw atoms and proteins anywhere. The shortcoming of this solution is the usage of the real world as a background, but it can distract and even disturb the user when trying to visualize the atoms.

2.1. Interaction and Virtual Reality

The introduction of virtual reality into problems, such as protein docking, is very important given the fact that the user has a totally different experience in terms of visualization and interaction with the virtual world. Mainly because of the new virtual reality devices allow various types of tracking, such as head, body and hand movement, the user can feel that it is part of the virtual world and thus the interaction, in this case, with the proteins becomes similar to reality and intuitive as if manipulating another real-world object.

Since this work focuses heavily on the manipulation of objects in immersive environment, it was important to study in depth the techniques of interaction. According to Poupyrev [18] manipulation techniques can be categorized as exocentric and egocentric. Exocentric manipulation techniques are those that when the user is interacting with the virtual environment from outside, using points of view not limited to users. In other words, exocentric techniques do not make users feel totally immersive in the virtual environment and give the impression that they are not part of virtual reality when using this technique. Examples of these techniques are works such as STALK [14], VRDD [1]. The egocentric techniques allow the user to feel inside the virtual world and to be able to realize interactions. These are the most commonly used techniques for immersive environments because the manipulation is based on human intuition and does not require any training. There are two basic metaphors for egocentric manipulations that have been well defined to this day that are the virtual hand and virtual pointer. With the virtual hand, users are able to execute manipulation commands, including hand tracing, gesture recognition, pointing, observation and hand movements to designate the parameters of interaction tasks. With the virtual pointer, users can manipulate objects by pointing to them, and the direction, shape, and methods that check the selected objects are the parameters of the interaction task.

3. Methodology

In order to develop the game we have taken into account the most popular game engines, however the one we decided to use was Unity given that it has a lot of advantages over others like having a very good integration with HTC Vive and also because we have previous experience in using it. Although is not the best for molecular rendering since it does not handle well having many objects (thousands of atoms, which are 3D spheres) in the scene at the same time, which can cause it to fall below thirty frames per second however with some optimizations at the mesh level used we can achieve a good performance in our game.

3.1. Architecture

To implement this game we used the following architecture (figure 3).

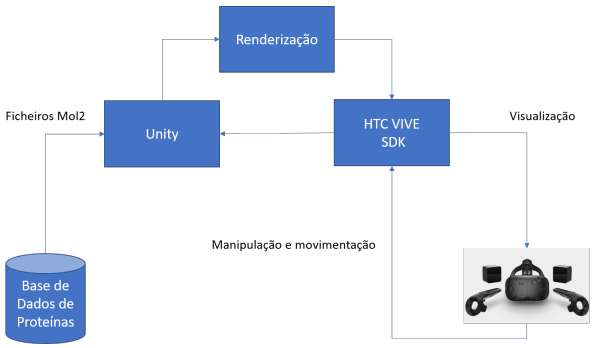

Figure 3: Architecture of the solution.

To initialize the game it is necessary to provide two proteins, the game accepts any kind

of protein in the MOL2 file. The contents of the .MOL2 file may contain multiple records such as ”MOLECULE”,”ATOM”,”BOND”, ”SUBSTRUCTURE” and ”SET”. In this case the records that interest us are the ”molecule”, ”atom” and ”bond”. In the record ”ATOM” are described the characteristics of the atom, the most important are the type of atom, which is given in the second column, the next three columns are the coordinates x, y and z and in the last column is given the electric charge of the atom. Therefore, for the creation of the three-dimensional objects that represent the atoms in the Unity, these three characteristics are necessary, and the type is used to determine the size of the atom, the coordinates for the position and the charge to be able to calculate the result of interaction between the proteins. The register ”BOND” represents the bond between the atoms, the second and third columns indicate the id of each atom. Although the bonds between the atoms are not visible to the user, they are useful for calculations and also for the physics of the game.

Then the scene is rendered and sent to the virtual reality device where it can be viewed by the user. Whenever the user makes a move with his arms and hands or moves itself is sent to Unity so that he can process the data and render the scene again to reflect the movements made by the user, also whenever the user manipulates the proteins Unity calculates the result of the interaction between them and renders the game again.

3.2. Implementation

For the representation of the proteins in the game we have studied several ways of implementing it since they can be represented in several ways. In order to make visualization, manipulation and later docking easier, we think that a more complete object, i.e. without empty spaces (as would happen if we used the point and stick representation) would be more advantageous for the user. So we decided to use the representation of the Van de Waals spheres to have a rigid body kind of look.

The first objective was to read an amino acid (a small part of a protein) since it was simpler to test if the parsing of the file was being well done. The most complex part to render the amino acid was to implement the R. Sayle [19] algorithm in order to create bonds between the atoms, since there are many rules to be able to know if an atom, within the same amino acid, has a connection or not. The algorithm was implemented and a three-dimensional representation of the amino acid was correctly generated and in order to be sure that it was well generated it was compared to other very popular three-dimensional visualization software called Avogadro[8].

There are two types of physics that we could use, conventional and molecular through a force field. The advantage of conventional physics is that it is simpler to calculate and it is already implemented by Unity itself which causes the game to be more fluid and less prone to lag. On the other hand, molecular physics as its name indicates is the best to represent all the forces involved in the interaction between proteins, such as the forces of repulsion and attraction, which is not the case with conventional physics. Although it was the best choice for the game, there were some doubts about the performance since it needs to be calculated in real time so as not to create lag.

In order to use an implementation of force fields that is already tested, we used a library called OpenBabel[16], because has the implementation of the most popular force fields used today. Before using any of the force fields a search was made to determine which is the best suited to our problem, and we decided to use GAFF (General Amber Force Field[20]) which was the best for our needs as it is most suitable for animal proteins. After we implemented the code needed to use OpenBabel[16], we found that the movement using the force field was very responsive and pretty close to what was said above. The part of the game engine was complete, yet it lacked the part of reading proteins with more than a thousand atoms and optimizing the meshes of the three-dimensional objects in order to improve the performance. To optimize meshes, a 3D modeling software was used to create a sphere with far fewer vertices than the primitive mesh used in Unity.

When we started using a protein with more than a thousand atoms, there was a problem that was the question of lag while the user was moving the protein using the molecular force field. From this point on we realized that the problem would only get worse since the protein used was one of the smallest, in order to solve this problem we tried several ways to improve the performance but we ultimately couldn’t do it which led us to use the physics engine for the interaction of proteins and using only the force field to calculate the result of the interaction between the proteins.

We started to think about the interaction and interface of the game, as already mentioned the game had to have a minimalist interface and also should be easy to interact with. For the interaction with protein we chose to follow an egocentric interaction technique through direct manipulation since it allows the user to perform intuitive movements and so does not need to learn anything, in addition this technique allows the user to perform tasks that require good precision.

To get real examples of proteins that are studied

in how they interact with each other we used a site where it has tables[11] of various proteins. In this project we chosen two simpler proteins to be able to demonstrate and run user tests more easily, because more complex proteins can have more than ten thousand atoms which would make difficult the visualization and manipulation.

3.3. Prototype

The first prototype was created after we started to parse the proteins and had also a basic manipulation technique with the mouse, so we could test any performance issue before start developing code to support the virtual reality device. In this prototype it was possible to test the game in order to find any problems or possible bugs, our concern was more about the problems related with collisions between the proteins, so in order to make sure the collisions would be more stable we made some adjustments in the size of the atoms. The interaction at this stage was still not the focus since it did not matter to improve it because later we would implement the manipulation using the commands of HTC VIVE.

The second prototype had already all the code needed to implement the virtual reality device. The focus of this prototype was to test some manipulation techniques that we thought that could be right for this game and improve the interface and visualization to be more adjusted to virtual reality. As we said we tested some manipulation techniques in order to know which one best suits our needs, after testing them we decided to go with a egocentric direct manipulation technique using the HTC VIVE controller, because this kind of manipulation allows the user to move the proteins without previous knowledge in how to do it and also because its very precise which is needed for this problem. The interface on virtual reality is a really complex problem because we need to position it in the right place so that all users can see it well, we can not simply use text and other elements that follow the user's head movement or that are too close to the screen of the device because the user will not be able to see it properly as the interface will be blurred, like we trying to read letters in a paper that are two centimeters away of our eyes. The solution to this phenomenon was to create an interface that does not move according to the user's head, but rather that it stays in a specific place of the scene and that the user can see whenever he wants it but at the same time do not distract him or the or actions he have to do.

The final prototype was achieved following user testing. After processing the suggestions, feedback and criticisms of the game, we made the adjustments that we decided were necessary to correct the flaws and also to improve aspects of the game that were not yet correct, especially with regard to the interface.

The interface to an immersive environment has several aspects that have to be taken into account so that it does not affect gameplay, such as text getting behind or in front of the objects that the user wants to see, and in our case manipulating . In order to solve this problem it was improved the positioning of the result and the best result to a position where the user could see whenever he wanted but without disturbing the game.

After implementing all the suggestions, which we found reasonably good to be used in the game, and fixing bugs, the final prototype was ready (figure 4).

4. Tests

The users chosen to carry out the tests were professional users in the field of biology such as Bioengineering, Biotechnology, Biological and Biomedical engineering, all having more than one year of experience. The decision to choose only biology professionals came about because we wanted to focus only on the part of visualization, interface and manipulation and in order to receive only feedback about these three aspects we needed users who already had a good knowledge base on atoms, amino acids and proteins, so we can get users to focus more on these aspects. This also makes the interface more minimalist rather than providing a lot information, which for professionals, would be unnecessary.

Although they are professional users only a few of them have had some experience with virtual reality and manipulation in an immersive environment, about 14% of users never had used virtual reality devices. About half of users who had already experienced virtual reality had also played some kind of game, especially puzzles and action. Another focus of the visualization was also the concern of the colors to be used so that user with color blindness did not have to strain to see whatever it was. Although in our tests there were no tester with any degree of color blindness, we decided to use a background color quite different from the colors of the atoms in order to create a good contrast.

4.1. Tasks

Before proceeding with tests, small instructions and information were given individually to each user. The information in this case was to inform about the purpose of the work and the tests that were to be carried out and also, if the user in question did not know the protein docking problem, briefly explain what it was and therefore explain the mechanics of the game.

The first task that the user was asked to perform was to make himself to get comfortable to the glasses and controllers of HTC VIVE. We created a

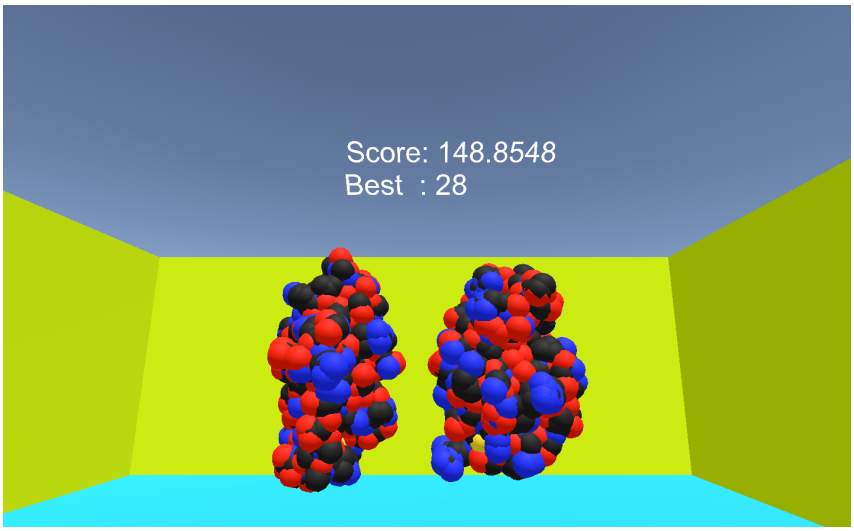

Figure 4: Final Prototype

level where there was no goal but to learn to move within the game and manipulate the proteins. To begin the test the user was asked to put himself at a specific point in the room so that when the game started he could see the two proteins away from him, as soon as he started he was supposed to walk close to the proteins and start to manipulate them. All users had five minutes to figure out how to handle and manipulate the proteins and also to see if they did not have any problems like nausea or vision.

As the users got more confident using the controllers, we then proceeded to the task more interesting for both the user and us, since in this task we wanted to be more attentive to the comments they made.

The second task was to try to dock two proteins together so that they could reach a minimum result of 20 energy in a maximum time of 10 minutes. During these ten minutes we always tried to tell the user and explain what he was trying to do. In the first tests we realized that the users were a bit confused with what really needed to be done so that the docking would generate a good result since it was very unstable. To avoid the user be confused about how to play we explained a little more at the beginning of the tests the question of having to try various positions and that there was also a better result indicator to help them choose a zone of the protein more subject to better results.

4.2. Feedback

In order to receive more comments after the game, a short interview was made with five questions, which we thought caused the user to reflect a little more to answer, so that we had more validation elements about the game.

In the end, the user was also asked to answer a series of questions related mainly to the in-game manipulation, ease and fluidity of the game in a questionnaire. In order not to allow the user to give neutral answers questions we've created the questionnaire using the Likert scale method 6 where it provides a response on a scale between 1 and 6, which corresponds to not agreeing at all to fully agree. Having all the questions a space for the user to detail the answers and also to give suggestions.

5. Results & discussion

In order to determine if the user liked the gaming experience, the question was asked immediately if they thought the game had been played, ie if the experience of manipulating and visualizing the proteins had had a problem in order to solve the problem . All users agreed the game had a good flow, with no major flaws.

As previously mentioned, the problem that was proposed to the users was an easier example of protein docking due to its size, but nevertheless the position required in order to have a result below 20 (which was the value that we proposed to users to pass the challenge) was more difficult to find, however, even with the somewhat difficult position to find all users said that the docking was somewhat easy to solve or approach the result 20.

In order to see if the user, due to the movement they could have done or even just by the colors and scene used, we wanted to know if our game created some kind of motion sickness. This is a concern for games that use virtual reality because if the game causes headache, dizziness or even motion sickness it means that users will not be able to play for a long time or even not play at all given that it causes users to get sick.

So two questions were asked so that the user could tell if had any type of fatigue and/or if him was indisposed. Only one user felt a slight indisposition

after playing the 15 minutes in a row while the rest did not have any indisposition, as for fatigue none of the users got tired after playing.

Due to the minimalist interface of the game we were in doubt whether the user would know whether or not the problem had been solved and also because the game does not have a predefined end since there is no absolute result, so we asked the users if they thought had success in solving the problem being that everyone agreed that yes but with some doubts, also because there was no feedback that had been solved.

It should be noted that all the users who did the tasks managed to get at least the result of 20, which gave us a good idea that the game was easy to play and that finding the position was not very difficult. Another note that one of the users mentioned was that the experience had not been completely fluid given that the proteins, due to the physical forces of the game, could get pushed each other far way and out of the user's reach causing them to restart to continue playing . This problem was solved by using a bounding-box around the area where the user was manipulating the proteins.

5.1. Interface and Visualization

Regarding the interface and visualization of the game, users were asked about various aspects such as position of interface texts, colors and size of proteins. In general, all the questions that were asked had positive feedback, which indicates that the direction we took was the right one.

So that the user can analyze the protein and its atoms well we found it necessary to adjust their size so that he could see the two proteins well and only then could come and see in detail any aspect he wanted. All users thought that the size of the atoms they viewed was adequate for that protein. It should be noted that if the protein had more atoms, such as five or six thousand atoms, it might have been necessary to reduce their radius a little so that one could see the whole detail of the protein.

Another important factor when making a protein docking is that it has all the information available, either through the interface or through another representation where the user can understand. We have decided to give atoms the color of how they are represented in all three-dimensional visualization programs of molecules, this makes it possible for professional users to extract information on how an atom interacts with another, more specifically whether atoms are repulsed or attracted to each other, and it is convenient for the nearest atoms to attract and further away those who repel. For this we wanted to know if it made sense to have or not another color map for proteins, such as a color gradient depending on whether the atom has positive or negative charge, and all users thought it would be quite interesting to add the option to change color map.

In several programs for visualizing proteins in three dimensions there are various options for rendering them, such as point-stick, cartoon, wireframe, Van der Walls. The representation of proteins is quite important so that the user can more easily dock the proteins to this we think that a representation of Van de Walls was the best for this game given that each atom has a specific radius, which causes the visualization to be more realistic, and also makes the protein appear complete, that is, without zones without atoms. All users also agreed that this rendering makes a lot of sense.

Given that the interface that the game is minimalistic in that it offers only the result and the best result to the user, we wanted to know if the users thought that it needed more information or not in order to facilitate the docking of the proteins. We did not enter anything else in the interface, as explained above, to avoid a large clump of information in the scene. Users thought that the given information was enough to solve the problem.

However there were a lot of suggestions so that when the user got a new best result, there was some kind of feedback for the user to know, since we only updated the best result. The suggestion was to introduce some kind of sound or even a flash so that the user would realize that he had reached a new result, however we thought that the audible warning was the best option since the flash could cause it to affect people with epilepsy.

In order to be able to help the user a little, even if they are professionals, we have implemented a better result indicator. This is intended to make the user know in which specific place and which has achieved its best result up to that time and also in which area is most likely to have the best result of the docking, ie, serves as a guide to help the user. According to users this feature is very important so that they do not get lost during the interaction, because after some time it may become difficult to know specifically in which areas of the two proteins is that the best result is found. While everyone thought the functionality was helpful there was a comment that the sphere used could be distracting at times.

6. Manipulation

The more intuitive the movements the user has to make easier the manipulation because they do not need to learn new movements of arms and hands to get something they just need to be close to the object where the arm and hand catch up. This is precisely what we want the person to do in the game, start at a position away from the proteins so they do not stay on top of them, and then walk

up to them and begin to interact.

In the walk-to-protein part no user realized that he could walk to them, perhaps for lack of explanation, experience with virtual reality devices or because they thought that if they could grasp the protein from afar, there could have been more context in the game mode that the user realized immediately that he could go to the protein.

However in order to understand if the users were satisfied with the movements they had to do we did a few more questions. The translational movement of the protein, i.e. from the moment the user grasps the protein until releases it, is sufficient to be fluid. All users agreed that the movement of the game was easy to perform and that it was quite fluid with no lag or other problems.

To support the rotation of the protein we think of some methods to facilitate this movement to the user, but we decided to use the total rotation of the pulse, that is, the user can rotate the protein up to a maximum angle allowed by the rotation of the pulse . This may not be very good because the user to achieve a rotation in 360 degrees would have to rotate the protein twice, but we nevertheless consider that the rotation thus is more realistic to what we are accustomed and, more importantly, its precise which helps a lot during the interaction between proteins. Most users agreed that the rotational movement of proteins was fairly easy to achieve.

7. Discussion

After the evaluation of both the questionnaires and the interviews, where the user who tested the prototype had the opportunity to talk about the experiment, we detected several aspects that needed to be improved mainly in the interface. Generally speaking, however, most users did appreciate the experience of manipulating and docking the proteins, especially since virtual reality allows the user to concentrate more and more easily manipulate proteins as wanted. In addition, it can more easily analyze details in the protein that it could not use using a monitor.

8. Conclusions

In this work we have analyzed the existing solutions and developed a solution to help solve the problem of protein docking so that it is possible to be played in an immersive environment where a much more natural and intuitive experience is obtained than using traditional methods such as the use of a mouse and/or keyboard.

With the implementation of the game in an immersive environment, we have been able to provide a more practical way to aid the process of protein docking, contributing to the improvement of the user experience. All users who participated in the tests agreed that it is a very positive experience and showed satisfaction of being able to perform the docking through virtual reality, allowing them a more realistic experience possible.

To conclude, the objective of this work was achieved through the creation of the game in an immersive environment and responds to the existing challenges of protein docking. This work is able to give users a more convenient and realistic experience, increasing the efficiency of the process and providing a new way to help the protein docking problem.

8.1. Future Work

To make the game more predictable and more realistic, it is necessary to add support for functions based on molecular force fields. For players who are not professionals in the field it is necessary to add additional information, such as name of the atoms and help throughout the game so that they understand everything they need to be able to play. In order to be able to render proteins with high complexity is required to have a higher optimization work for the game to continue fluid, it will also be necessary to adjust interface parameters and the size of the atoms themselves so that the protein does not occupy too much space in the scene.

References

[1] A. Anderson and Z. Weng. Vrdd: applying virtual reality visualization to protein docking and design. *Journal of Molecular Graphics and Modelling*, 17(3):180–186, 1999.

[2] S. Cakici, S. Sumengen, U. Sezerman, and S. Balcisoy. Dockpro: a vr-based tool for protein-protein docking problem. In *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, page 11. ACM, 2008.

[3] S. R. Comeau, D. W. Gatchell, S. Vajda, and C. J. Camacho. Cluspro: an automated docking and discrimination method for the prediction of protein complexes. *Bioinformatics*, 20(1):45–50, 2004.

[4] C. Dominguez, R. Boelens, and A. M. Bonvin. Haddock: a protein- protein docking approach based on biochemical or biophysical information. *Journal of the American Chemical Society*, 125(7):1731–1737, 2003.

[5] L. P. Ehrlich and R. C. Wade. Protein-protein docking. *Reviews in computational chemistry*, 17:61–98, 2001.

[6] N. Férey, J. Nelson, C. Martin, L. Picinali, G. Bouyer, A. Tek, P. Bourdot, J.-M. Burkhardt,

B. F. Katz, M. Ammi, et al. Multisensory vr interaction for protein-docking in the corsaire project. *Virtual Reality*, 13(4):273, 2009.

[7] A. Gruber, A. Durham, C. Huynh, and H. del Portillo. Bioinformatics in tropical disease research: a practical and case-study approach. *Bethesda (MD): National Library of Medicine (US), NCBI*, 2008.

[8] M. D. Hanwell, D. E. Curtis, D. C. Lonie, T. Vandermeersch, E. Zurek, and G. R. Hutchison. Avogadro: an advanced semantic chemical editor, visualization, and analysis platform. *Journal of cheminformatics*, 4(1):17, 2012.

[9] I. Hashmi and A. Shehu. Hopdock: A probabilistic search algorithm for decoy sampling in protein-protein docking. *Proteome science*, 11(1):S6, 2013.

[10] W. H.-Y. Huang and D. Soman. Gamification of education. *Research Report Series: Behavioural Economics in Action, Rotman School of Management, University of Toronto*, 2013.

[11] H. Hwang, T. Vreven, J. Janin, and Z. Weng. Protein–protein docking benchmark version 4.0. *Proteins: Structure, Function, and Bioinformatics*, 78(15):3111–3114, 2010.

[12] G. Kiryakova, N. Angelova, and L. Yordanova. Gamification in education. Proceedings of 9th International Balkan Education and Science Conference, 2014.

[13] G. Levieux, G. Tiger, S. Mader, J.-F. Zagury, S. Natkin, and M. Montes. Udock, the interactive docking entertainment system. *Faraday discussions*, 169:425–441, 2014.

[14] D. Levine, M. Facello, P. Hallstrom, G. Reeder, B. Walenz, and F. Stevens. Stalk: An interactive system for virtual molecular docking. *IEEE Computational Science and Engineering*, 4(2):55–65, 1997.

[15] I. C. London and G. College. Bioblox. `http://www.bioblox.org/`, 2015. [Online; accessed July-2017].

[16] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison. Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3(1):33, 2011.

[17] B. G. Pierce, K. Wiehe, H. Hwang, B.-H. Kim, T. Vreven, and Z. Weng. Zdock server: interactive docking prediction of protein–protein complexes and symmetric multimers. *Bioinformatics*, 30(12):1771–1773, 2014.

[18] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst. Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In *Computer graphics forum*, volume 17, pages 41–52. Wiley Online Library, 1998.

[19] R. Sayle. Pdb: Cruft to content. *MUG 2001*, 2001.

[20] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, and D. A. Case. Development and testing of a general amber force field. *Journal of computational chemistry*, 25(9):1157–1174, 2004.

[21] M. Zheng and M. P. Waller. Chempreview: an augmented reality-based molecular interface. *Journal of Molecular Graphics and Modelling*, 73:18–23, 2017.