# GLUBs: Games for Learning and Understanding Biology

Grigore D. Pintilie
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory
Cambridge, MA
gdp@csail.mit.edu

## ABSTRACT

This paper introduces GLUBs, which are games based on molecular mechanisms that take place in biological systems. Such games make use of interactive 3D graphics, and allow the player to interact with molecular components using simple 2D mouse gestures. They are similar to puzzle games, since the goal is to discover the correct interactions between molecular components. When the correct interactions are found, the player observes a molecular mechanism taking place. GLUBs thus have the potential to communicate much of the growing structural and biochemical knowledge in a fun and interesting way. In the game presented in this paper, the molecular mechanism involves the assisted folding of a protein by a chaperone.

## Categories and Subject Descriptors

K.3.0 [**Computers and Education**]: General; J.3 [**Life and Medical Sciences**]: Biology and Genetics

## General Terms

Algorithms

## Keywords

Games, 3D rendering, human-computer interaction, proteins, molecules, Brownian dynamics.

## 1. INTRODUCTION

Biological systems such as living cells are very complex entities. They are made up of numerous molecular components such as small molecules, proteins, deoxyribonucleic acid (DNA), and ribonucleic acid (RNA) molecules [1]. The study of such systems involves obtaining the structure of molecular components and discovering how the components interact with one another. Molecular components are extremely small, being on the scale of nanometers, and thus molecular structures are hard to obtain. Despite this, many molecular structures have been obtained through methods such as X-ray crystallography [14], and cryo-electron microscopy [5]. Moreover, biochemical experimental methods have elucidated how molecular components interact with one another to perform various molecular mechanisms [15]. Through games such as the one presented in this paper, we aim to bring such knowledge to life through the use of interactive 3D computer graphics and techniques from game design.


*FuturePlay @ Vancouver Digital Week 2010*, May 6-7, 2010 Vancouver, Canada.


## 2. OVERVIEW

A GLUB is based on a molecular mechanism, which is generally performed by a number of molecular components. Structural information is used to produce the geometries of such components. During the game, molecular components move naturally as if in their native environment. This native environment, in reality, is mostly a solution of water molecules. In such an environment, movement can be described using Brownian dynamics [9], which looks much like random displacements and rotations. To make the game interesting, the player can exert some control over the molecular components, by adding an artificial bias to the random displacements. A molecular component can thus be driven towards a particular location, for example towards contact with another molecular component.

When molecular components come into contact with one another in a particular way, they can bind to create a molecular assembly. Through binding, the shape of either one or both of the molecular components can change, and through such changes, certain mechanisms are performed. A GLUB is based on a known molecular mechanism, and the objective is for the player to uncover this molecular mechanism by directing the molecular components to move towards and to bind to each other. When the correct binding is obtained, the molecular mechanism can be observed. Through playing the game, the player thus learns a molecular mechanism and comes to understand its structural basis.

In the first GLUB presented in this paper, the mechanism used is that of assisted folding by a molecular complex known as a chaperone [15]. This mechanism is illustrated in Figure 1. The chaperone is made up of 16 proteins, which are arranged in two rings of 8 proteins each. The rings are stacked on top of each other, forming an overall spherical shape. Each ring has an open cavity, and the interior sides of this cavity recognize and bind unfolded proteins. When

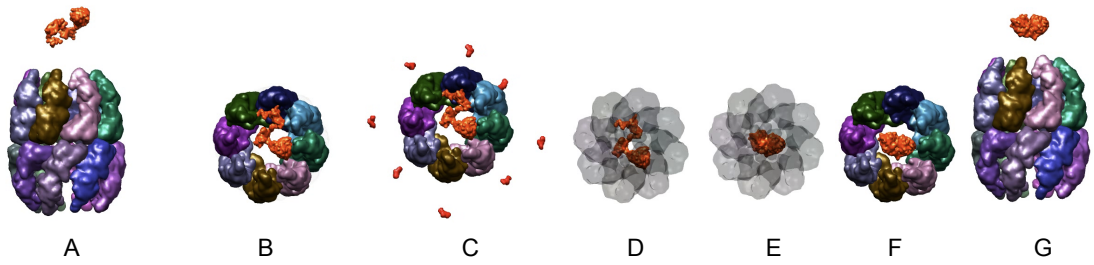


**Figure 1: Illustration of the molecular mechanism on which the first GLUB is based. The molecular components involved in this mechanism are: an unfolded protein (orange surface in A), a chaperone (made up of 16 proteins, each protein being shown with a randomly colored surface, A-G), and ATP molecules (red surfaces, image C). The mechanism starts when the unfolded protein enters the cavity of the chaperone (A,B). After ATP molecules bind to each protein in the top ring of the chaperone (C), the ring closes (D). Inside the closed chaperone, now shown using transparent surfaces, the protein folds (E). Then the ring opens (F), and the protein, now properly folded, exits the chaperone (G).**

an unfolded protein is bound inside the cavity, the chaperone changes its structure so that the cavity closes. This change in structure is driven by the binding of energetic adenosine triphospate (ATP) molecules to individual proteins. The closing of the cavity induces the unfolded protein to fold to its native state, which is also the functional form of the protein. Once the protein is folded, the cavity opens, and the folded protein exists the chaperone. This assisted-folding process is a very important mechanism in many living cells: it helps some proteins to fold to their native and functional forms, which are critical to the proper functioning of the cell.

# 3. METHODS

## 3.1 Geometry of molecular components

The visualization of each molecular component is a vital aspect of a GLUB. It is important for the visualization to be accurate, however also somewhat simplified so that it is easy for the player to clearly identify each molecular component, and so that the rendering of numerous molecular components can be performed at interactive frame rates.

The structures of molecular components such as proteins are available in public repositories such as the protein data bank (PDB)[8]. This structural information generally takes the form of a 3D coordinate for each atom in the protein. A protein can easily contain thousands of atoms, and in molecular assemblies, this number can be multiplied by another factor of 10 (e.g. in assemblies consisting of 10 proteins). To draw each atom is thus not only visually overwhelming but also computationally expensive. Instead, a GLUB uses a surface representation for each molecular component, in which a single surface encloses all the atoms that make up the molecular component.

### *3.1.1 Chaperone*

The atomic structure of the chaperone in the closed form was obtained by X-ray crystallography and is available in the PDB (PDB:1q3s) [10]. It is an assembly of 16 proteins, which are arranged in two rings, each ring having 8 proteins. The two rings are stacked on top of each other, as shown in Figure 2, forming a spherical shape. The inside of the sphere is an empty cavity.

Using a molecular visualization program, such as Chimera [6], there are several visualizations possible for such an atomic structure: each atom can be drawn as a sphere, as shown in Figure 2B, each atom can be drawn as a smaller sphere and bonds between atoms drawn as tubes, as shown in Figure 2C, or a single smooth ribbon can be drawn so as to interpolate atomic positions in the backbone of the protein, as shown in Figure 2D. These visualizations can be quite complex and may be unnecessarily slow to render.

Surface representations can also be generated using Chimera, by first turning the atomic structure into a density map. In this process, a 3D grid is created around the structure, and values from Gaussian functions centered on each atomic position are added together at all grid points. The marching cubes algorithm [4] is then used to extract the surface at a user-controlled threshold. The standard deviation of the Gaussian functions determines the resolution of the density map: the lower the standard deviation, the higher the resolution. At higher resolutions, individual atoms can be discerned in the density map, but at lower resolutions, atoms blend together in the smoother density map.

The resolution also implicitly determines the grid spacing that can be used for the density map. For higher resolutions, denser grids are required to capture the higher frequency variations in the density function. At higher resolutions, the resulting surfaces also have more triangles. The choice of resolution is thus a tradeoff between detail in the surface and the number of triangles obtained; less detail means faster rendering and smaller memory requirements.

Surfaces obtained for a protein in the chaperone at various resolutions are shown in Figure 2(E-G). For the geometry of the chaperone used in this GLUB, we chose the resolution high enough such that the overall shape of each protein is represented with a reasonable amount of detail. The resolution was also chosen to be not too high, to allow faster rendering and smaller memory requirements.

The structure of the chaperone in Figure 2 is only of the closed state as obtained by X-ray crystallography. Both the closed and open conformations have also been obtained using

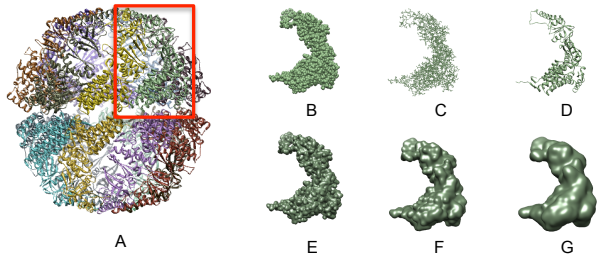


**Figure 2: All proteins in the chaperone are shown in A, each one drawn as a ribbon of a different color. The red box outlines a single protein, which is shown in B using a sphere for each atom, in C as spheres for each atom and tubes for bonds between atoms, and in D as a ribbon. Surfaces for the same protein are shown at decreasing resolution (increasing smoothness) in E, F, and G. In GLUBs, we use the surface representation, since it is simpler and faster to render.**

cryo-electron microscopy (cryo-EM) [15], as shown in Figure 3 (A,B). This method produces density maps, which only store a density value at regular intervals in a grid. Thus, information about individual proteins is not explicitly given by such density maps. To identify the different proteins, the density maps were first segmented using a multi-scale segmentation method [7]. This method produced regions corresponding to each protein in both the open and closed state. These segmentations are shown in Figure 3 (C,D).

Using the segmented regions as guides, the atomic structure of each protein in the closed state was fitted into the density map of the closed state. The structures were driven to the open state using molecular dynamics [12], to model the transition from closed to open state of each protein. The surfaces of each protein at 150 intermediate states were computed after the simulation was complete. The surfaces at 6 of these intermediate states are shown in Figure 4. These surfaces are used in the GLUB to smoothly animate the transition from a closed to open states, either forward or backward as needed.

### *3.1.2 Folded and unfolded protein*

A protein that is known to be bound by the chaperone in unfolded form is gamma-D crystallin, which is a protein that is found in the eye lens. The structure of gamma-D in the folded state has been obtained by X-ray crystallography, and is available in the PDB (PDB:1hk0). The structure of a large unfolded protein cannot be directly seen by any structural discovery method, since in an unfolded form, a protein is highly dynamic.

To obtain the structure of an unfolded protein, we again used a molecular dynamics simulation. During the simulation, artificial forces were applied to unfold the protein. A surface of the protein was saved at 150 intermediate steps between the initial folded state and the final unfolded state. Several of these surfaces are shown in Figure 4. These surfaces are used during game play to animate between unfolded and folded forms for the protein.

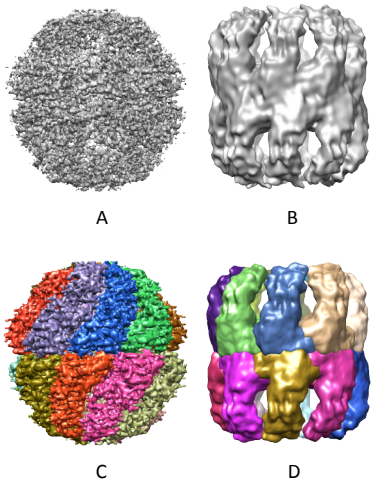


**Figure 3: Open and closed forms of the chaperone are shown, as seen in density maps obtained by cryo-electron microscopy. The density maps were taken from the Electron Microscopy Data Bank (EMDB), and have accession IDs 5137 (A) and 5139 (B). Images A and B show an iso-surface through each density map, drawn at a threshold such that the overall shape of the chaperone is captured. Images C and D show the segmentation of the density maps, with segmented regions drawn using different colors. Each region corresponds to a single protein in the chaperone.**

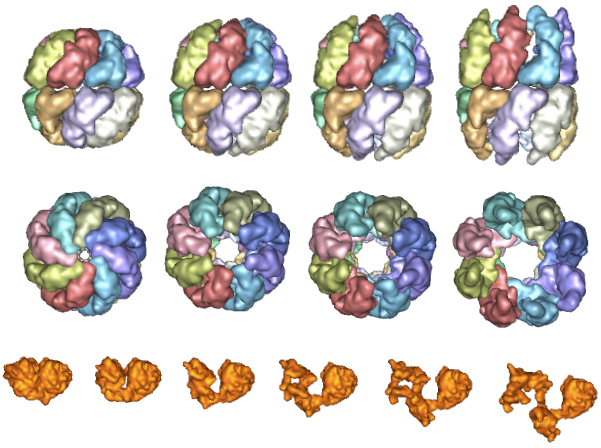

**Figure 4: The top and middle rows show the chaperone transitioning from closed to open (left to right). The bottom row shows the surface of a protein which the chaperone binds, transitioning from folded to unfolded.**

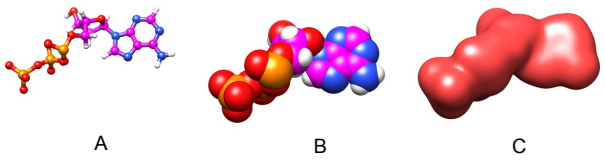


**Figure 5: An adenosine triphospate (ATP) molecule is shown using spheres for atoms and tubes for bonds in A, larger spheres for each atom in B, and a surface in C. In a GLUB, the surface representation is used because it is somewhat simpler and the level of detail is sufficient for illustration of a biological mechanism.**

### *3.1.3 ATP molecules*

Another molecular component that is part of the GLUB presented in this paper is the adenosine triphosphate (ATP) molecule. This molecule plays an important role in the molecular mechanism that this GLUB is based on. In particular, once an unfolded protein is bound by the chaperone, an ATP molecule must also bind to each of the proteins in the ring which has bound the unfolded protein. This drives the ring to change from the open state to the closed state. This process requires energy, which is taken from the ATP molecule through the breaking of one of its covalent bonds, yielding adenosine diphopshate (ADP) and phosphorus molecules. The structure of an ATP molecule is shown in Figure 5, using 3 different representations. Again the surface representation is used, mainly for its visually simpler appearance, and the fact that this level of detail is sufficient in this context.

## 3.2 Dynamics

### *3.2.1 Movement of molecular components*

To make a GLUB more interesting, molecular components move during game play much like they do in their cellular environments. There, molecular components are surrounded by water and other molecules. The motion of a molecule in such an environment can be modeled as a diffusion process using Brownian dynamics [13].

The movement of two molecules is illustrated in Figure 6. The position of a molecule, $i$, at a time $t$, $\overrightarrow{p}_i(t)$, is given by its position at the previous time, $t-1$, plus a random displacement:

$$\overrightarrow{p}_i(t) = \overrightarrow{p}_i(t-1) + a\hat{r} \quad (1)$$

In the above equation, $\hat{r}$ is a random unit vector, and $a$ is the displacement magnitude. The latter is drawn from a random normal distribution with the mean given by a displacement diffusion coefficient, $D_i$ and the time step, $dt$. In other words, over a number of steps, the generated magnitudes would be expected to have the mean:

$$\sqrt{\langle a \rangle} = \sqrt{6 D_i dt} \quad (2)$$

The diffusion coefficient of a molecule, $D_i$, depends on factors such as the size and shape of the molecule. For example, smaller molecules have higher diffusion coefficients, and move around much faster than larger molecules.

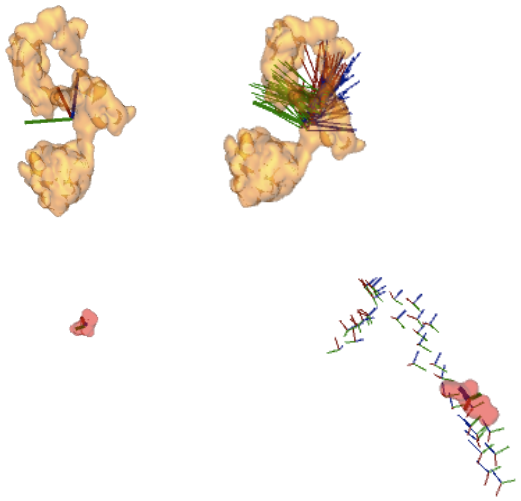

**Figure 6: Movement of two molecular components: an unfolded protein (orange) and an ATP molecule (red). The local coordinate axes are shown using three lines. On the left, the starting positions and orientations are shown. On the right, the axes are drawn at a series of positions and orientations that each molecule goes through while undergoing Brownian motion. Over the same time period, the smaller molecule moves more than the larger protein, since it has higher displacement and rotational diffusion coefficients.**

The time step, $dt$, should be extremely small for molecular diffusion: on the order of pico- to nano-seconds. For the purposes of the game, we avoid aiming for accurate time step and diffusion coefficients, and merely choose values that lead to reasonable movement during play. To be somewhat realistic, we choose diffusion coefficients for molecular components based on their size, so that smaller molecules have higher diffusion coefficients, and larger molecules have lower diffusion coefficients.

The motion of a molecule is also such that it rotates randomly from one time step to the next. This rotation, like the displacement movement, is also a result of the many random collisions between it and the other molecules around it. Each collision imparts some torque on the molecule, and at any given time, the net torque is not likely to be zero, and thus the molecule rotates.

In our implementation, the orientation of a component is maintained as a quaternion, which is updated from one time step to the next by multiplying it with a randomly generated unit quaternion. The latter is computed such that it represents a rotation around a random axis, with a rotation magnitude drawn from a random normal distribution with a mean based on a rotational diffusion coefficient (similar to Eqn. 2). Like displacement diffusion coefficients, rotational diffusion coefficients are set to be larger for small molecules

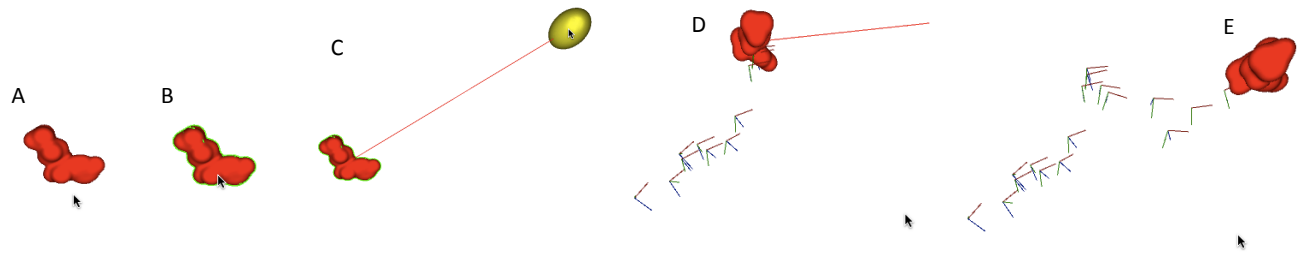


**Figure 7: Moving a molecule to an arbitrary point in 3D. The molecule is first shown in A. Once the player moves the mouse over it, it is selected, as indicated by a green outline (B). The player clicks and drags, and while doing so, a target point is moved by the mouse (yellow dot). A line is drawn between the current center of the molecule and the target point (C). The molecule continues to move as if diffusing, however a bias term drives the molecule towards the target point (D,E). The line between the position of the molecule and the target point is drawn at every frame, indicating the molecule is moving towards a target point (D). The axes in D and E are drawn here at intermediate positions of the molecule to illustrate its movement, and are not actually seen by the player during the game.**

and smaller for large molecules.

#### *3.2.2 Collision detection*

To maintain realism during gameplay, molecules are not allowed to interpenetrate. Collision detection is performed by checking triangle-triangle intersections between the meshes for different molecules. To make this fast, each mesh is first decomposed into an oriented bounding-box tree [3], which is used to prune out molecules and triangles that cannot possibly intersect when testing for molecule-molecule and triangle-triangle intersections. When a molecule is moved using Eqn. 1, the surface in the new position and orientation is checked for collisions with surfaces of other molecules. When a collision is found, the move is simply not allowed, and another move is tried.

### 3.3 Control by player

The interface in a GLUB allows the player to view the 3D scene from different viewpoints, by rotating, panning and zooming the camera. The player is also given control over the movement of the molecules: they can select a molecule, and specify a point in space it should move toward. This point can be an arbitrary point in space (these will be called *target points*), or a pre-specified zone where the molecule could potentially interact with another molecule (these will be called *target zones*).

#### *3.3.1 Selecting a molecule using the mouse*

The player can select a molecular component simply by placing the mouse over it. To indicate that the molecule has been selected, an outline is rendered around it (as described in [2]). This is illustrated in Figure 7 A,B.

#### *3.3.2 Target points*

When a molecule is selected, the player can click and drag the mouse, so as to specify anoter point in space the molecule should move toward. The location of the mouse in 2D is transformed to a 3D point, towards which the molecule starts moving when the mouse button is released (see Figure 7C). The 3D target point is found as follows: the 2D mouse coordinates are used to create a 3D ray, which starts at the camera position in the scene and passes through the mouse coordinates mapped to the viewplane. This ray is intersected with a plane that goes through the center of the selected molecule and is perpendicular to the view direction, and the 3D target position is set to the intersection point. This type of interaction has been commonly used in 3D interfaces before, and basically maps a 2D mouse position to a point in the 3D scene [11].

Once a target position has been specified for a molecule, a bias term is added to the equation of motion (1) for that molecule. The bias term is simply a vector, with direction determined at each time step. The direction is from the center of the molecule to the target position. The magnitude of the vector is set to 0.1 of the magnitude of the displacement for the molecule computed using Equation (2). The effect of this artificial bias term is that that the molecule continues to randomly move as if diffusing, but gradually makes its way towards the target point (Figure 7D,E).

#### *3.3.3 Target zones*

Using the method for specifying a target point as described above, it would be very tedious for the player to get a molecule to move to a specific position in the scene, for example inside the cavity of the chaperone. The camera position and direction would likely have to be adjusted multiple times, until the plane on which the target point is computed (perpendicularly to the view plane) would pass through the desired position, e.g. inside the cavity of the chaperone. This is a hard task, and would easily frustrate the player.

To make it easier for the player to specify the position of a molecule with respect to another molecule, for example inside a cavity, each molecule can be given one or more target zones. The target zones for a given molecule are places where another molecule can bind. For example, the chaperone has several target zones as illustrated in Figure 8.

As the player is drawing out a target point for a selected molecule, if the mouse is positioned over a target zone, the

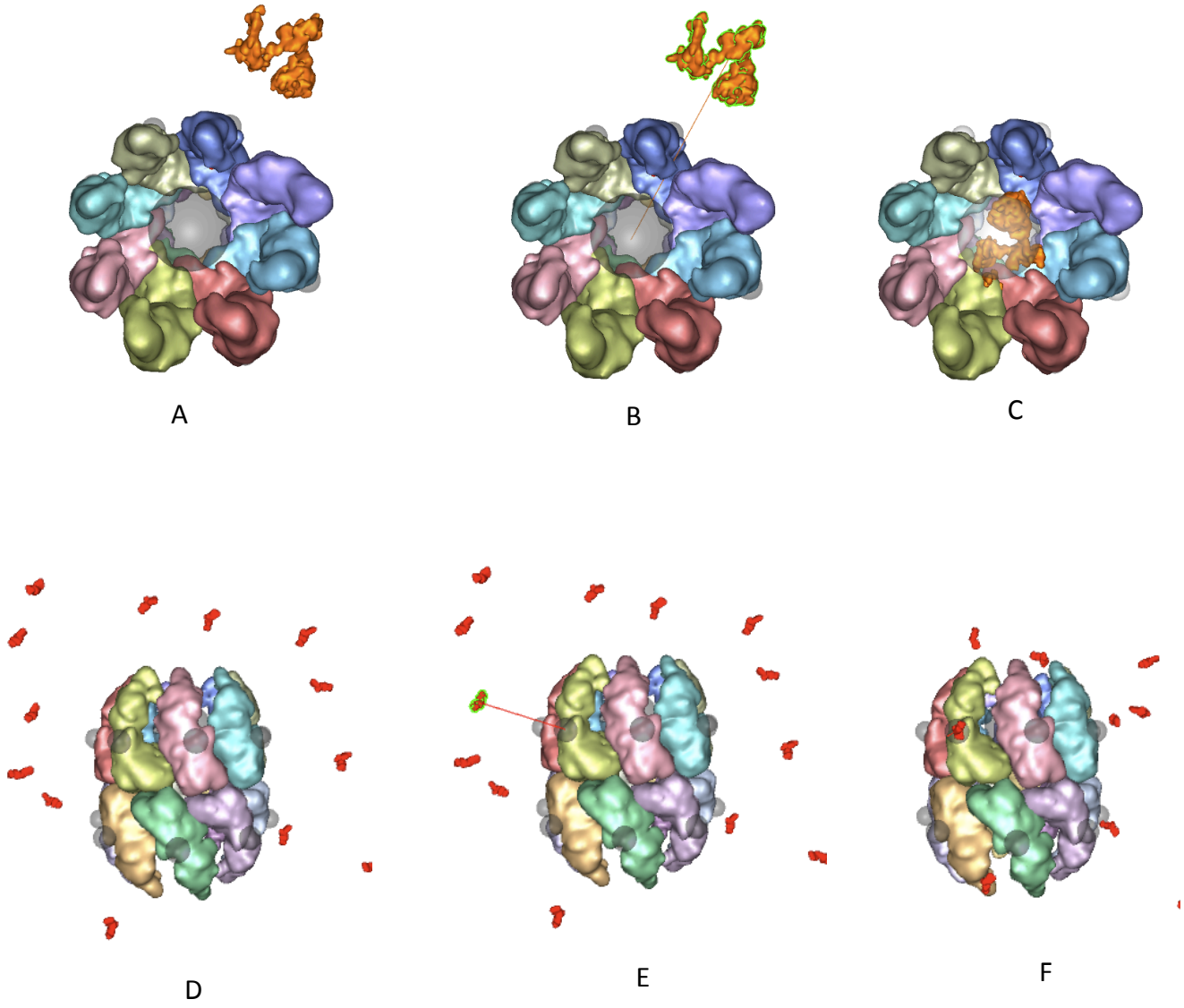


**Figure 8: Moving molecules into target zones. In (A), the target zone inside the cavity of the chaperone is drawn using a transparent sphere. In B, the player has selected a molecule (the unfolded protein), and dragged a line to the target zone. When the mouse is placed over the target zone, the target point automatically becomes the center of the target zone. The molecule then gradually diffuses towards the target zone, eventually reaching it, as shown in C. The chaperone also has target zones for ATP molecules, which are shown in D, also with transparent spheres which are adjacent to each protein in the chaperone. In E, the player selects an ATP molecule and drags a line to one of the target zones. Eventually the ATP molecule reaches the target zone, as shown in F.**

target zone is selected, and the target point becomes the center of the zone. The molecule will then randomly diffuse towards this position, and eventually make it so that its center is inside the target zone. This is illustrated in Figure 8. If this binding is one that is a part of the mechanism the GLUB is based on, the target zone starts to flash, so that the player knows that they discovered an important component of the mechanism.

## 3.4 Gameplay

### *3.4.1 Objective*

The objective of a GLUB is for the player to discover the molecular mechanism that the game is based on. Since the player will typically not know the mechanism ahead of time, the game is much like a puzzle, where the player has to put the pieces together from the structural information that is presented. This information takes two forms:

1. The structure of the molecular components. For example, as illustrated in Figure 8, the fact that the chaperone has a central cavity in the open state would indicate that something (perhaps an unfolded protein) can go inside this cavity.
2. Zones on molecular components, into which other molecules can be moved. Zones indicate where a molecular component might bind to another molecular component. For example, as illustrated in Figure 8, the zones next to each protein in the chaperone indicate that something might have to bind there in order to trigger some part of the mechanism.

The two types of information listed above can thus aid the player in solving the puzzle. A player can base their actions on this information, or as a last resort proceed by trial and error. They might, for example, move a molecule into a random target zone. If the molecule does not seem to fit, and if the zone does not start flashing, then the player realizes that this is not part of the mechanism. The correct placement of molecules in some cases also triggers transitions in the structures of the molecules, and when the player observes these, they also know that they discovered another part of the mechanism.

### *3.4.2 Gameplay for the first GLUB*

The GLUB presented in this paper is based on the molecular mechanism illustrated in Figure 1. The goal of the player is to first move the unfolded protein into the central cavity of the chaperone. Then, the 8 ATP molecules have to be moved into the zones next to each protein. Once all 8 ATP molecules are bound, the chaperone closes, and the unfolded protein folds. After a small duration of time, the chaperone opens and the folded protein is released.

### *3.4.3 Difficulty considerations*

An important consideration for any game is the difficulty level. There are at least three aspects in determining the difficulty level of a GLUB:

*Complexity of the mechanism.* In the GLUB presented in this paper, the player is only expected to correctly execute two types of actions: the first involves moving an unfolded protein into the cavity of the chaperone, and the second involves moving an ATP molecule so that it binds into a zone next to one of the proteins that make up the chaperone. A biological mechanism involving more components and more types of actions would thus be considered more difficult.

*Multiple simultaneous mechanisms.* Another way to make a GLUB more difficult would involve duplicating each component multiple times, so that the player would have to perform the same mechanism multiple times simultaneously. Since more components would mean more clutter, it would also be more difficult for the player to find the right component required for a particular zone or for a particular stage of a mechanism.

*Time considerations and hints.* Another concept often used in puzzle games is to limit the time the player has to reach the goal. This could also be used in GLUBS, however, since they are meant to be educational games, another alternative is to provide clues if, after some time, the player has not made any progress.

# 4. CONCLUSIONS AND FUTURE WORK

The information discovered by various biological experiments can be very complex. In this paper, it was demonstrated how such information can be used to create games that are interesting and hopefully fun to play. Such games can thus serve to educate as well as entertain. The game presented in this paper is based on a molecular mechanism in which a chaperone helps an unfolded protein to fold. In the future, we hope to more thoroughly study how players respond to and perform in such games, and to develop more GLUBs based on other molecular mechanisms.

# 5. REFERENCES


[1] Branden, C. and Tooze, J. 1999. Introduction to protein structure. Garland.

[2] DeCarlo, D., Finkelstein, A. et al. 2003. Suggestive contours for conveying shape. ACM Trans. Graph. 22, 3 (2003), 848-855.

[3] Gottschalk, S., Lin, M.C. et al. 1996. OBBTree: a hierarchical structure for rapid interference detection. Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (1996), 171-180.

[4] Lorensen, W.E. and Cline, H.E. 1987. Marching cubes: A high resolution 3D surface construction algorithm. SIGGRAPH Comput. Graph. 21, 4 (1987), 163-169.

[5] Ludtke, S.J., Baldwin, P.R. et al. 1999. EMAN: semiautomated software for high-resolution single-particle reconstructions. Journal of structural biology. 128, 1 (Dec. 1999), 82-97.

[6] Pettersen, E.F., Goddard, T.D. et al. 2004. UCSF Chimera–a visualization system for exploratory research and analysis. Journal of computational chemistry. 25, 13 (Oct. 2004), 1605-12.

[7] Pintilie, G., Zhang, J. et al. 2009. Identifying components in 3D density maps of protein nanomachines by multi-scale segmentation. Life Science Systems and Applications Workshop, 2009. LiSSA 2009. IEEE/NIH (2009), 44-47.

[8] Protein Data Bank. http://www.pdb.org.

[9] Schlick, T. 2002. Molecular Modeling and Simulation: An Interdisciplinary Guide. Springer-Verlag New York, Inc.

[10] Shomura, Y., Yoshida, T. et al. 2004. Crystal structures of the group II chaperonin from Thermococcus strain KS-1: steric hindrance by the substituted amino acid, and inter-subunit rearrangement between two crystal forms. Journal of molecular biology. 335, 5 (Jan. 2004), 1265–78.

[11] Strauss, P.S., Issacs, P. et al. 2002. The design and implementation of direct manipulation in 3D. SIGGRAPH 2002 Course Notes. (2002).

[12] Trabuco, L.G., Villa, E. et al. 2008. Flexible Fitting of Atomic Structures into Electron Microscopy Maps Using Molecular Dynamics. Structure (London, England : 1993). 16, 5 (May. 2008), 673–683.

[13] Truskey, G.A., Yuan, F. et al. 2009. Transport phenomena in biological systems. Prentice Hall.

[14] Woolfson, M.M. 1997. An Introduction to X-ray Crystallography. Cambridge University Press.

[15] Zhang, J., Baker, M.L. et al. 2010. Mechanism of folding chamber closure in a group II chaperonin. Nature. 463, 7279 (Jan. 2010), 379-383.