

List of Contents

Nomenclature	iii
1 Virtual reality – theoretical background	1
1.1 Definition	1
1.2 History	2
1.3 Hardware and software	4
1.3.1 Immersion	4
1.3.2 Sensory feedback- output	5
1.3.3 User's input	9
1.3.4 Software	11
1.4 Examamples of hardware	13
1.4.1 <i>Oculus Rift</i>	13
1.4.2 <i>Google Cardboard</i>	14
1.4.3 <i>Leap Motion</i>	14
1.5 Possible applications	16
2 Visualisation of Molecular Structures	17
2.1 Biomolecular Structure Determination	17
2.1.1 X-ray crystallography	17
2.1.2 NMR Spectroscopy	18
2.1.3 Electron Microscopy	19
2.1.4 Atomic Force Microscopy	20
2.1.5 Databases	20
2.2 Representation of Structures with Computer Graphics	21
2.3 Virtual Reality in Visualisation of Structures	23
2.3.1 Motivation	23
2.3.2 Examples	24
2.3.3 Problems and future	24
3 Molecule VR Viewer	27
3.1 Main Menu	27
3.2 Drawing Styles	28
3.2.1 Van der Waals	28
3.2.2 Lines	30

3.2.3	Balls and Sticks	30
3.2.4	Backbone	31
3.2.5	Ribbon	31
3.3	Colouring styles	32
3.3.1	CPK	32
3.3.2	Residues	33
3.3.3	Subunits	34
3.3.4	Secondary Structures	35
3.4	Visualisation Scene	35
3.4.1	Rotating and Zoom	36
3.4.2	Selecting mode	36
3.4.3	Exit	38
4	Molecule VR Viewer- implementation	39
4.1	<i>Unity 3D</i>	39
4.1.1	Editor	39
4.1.2	Scripting	40
4.2	<i>Implementation details</i>	40
4.2.1	MainMenu.cs	40
4.2.2	MoleculeViewer.cs	42
References		45

Nomenclature

Acronyms / Abbreviations

API	Application Programming Interface
AR	Augmented Reality
AV	Augmented Virtuality
CPU	Central Processing Unit
DOF	Degree of Freedom
EOG	Electro-Oculography
FOV	Field of View
GPU	Graphics Processing Unit
HMD	Head-Mounted Display
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
IR	Infrared
LOD	Level of Detail
mmCIF	macromolecular Crystallographic Information File
MMDB	Molecular Modelling Database
MR	Mixed Reality
NCBI	National Center for Biotechnology Information
NMR	Nuclear Magnetic Resonance
PDBML	Protein Data Bank Markup Language
PDB	Protein Data Bank

RE	Real Environment
SDK	Software Development Kit
VE	Virtual Environment
VR	Virtual Reality
WOW	Window on World

*Reality is merely an illusion, albeit
a very persistent one.*

Albert Einstein

Chapter 1

Virtual reality – theoretical background

1.1 Definition

Nowadays, thanks to constant technology development, accessibility of Internet and computer-related innovations, people tend to live parallelly in two worlds- real and virtual. As the latter one offers less boundaries and constraints, and the only limit there is, is people's imagination, it is not surprising that these days one can observe growing trend in improving the quality of man's immersion in artificial world. This possibility leads to the new idea of perceiving the surrounding reality, which is called Virtual Reality (VR) or Virtual Environment (VE).

The concept of VR has never been unified, therefore it cannot be defined in one way. One of the most general definition explains it as "an immersive and interactive system that provides users with the illusion of entering a virtual world" [1]. In other papers one can find more specific description. Most of them state that three-dimensional graphics is an essential part of VR [2], moreover, that it needs to provide a multi-sensory computer-mediated environment [3]. Despite the multitude of possible definitions, all of them agree that VE is immersive, interactive and that it creates an experience in simulated world. It is well systematized by Sherman and Craig [4], who describe four essential elements in VR, which are:

- **Virtual world**– set of objects in a space, relationship between them and laws that govern this entirety.
- **Immersion**– impression of being present in showed environment instead of looking at it from the outside. This phenomenon will be described more precisely in section 1.3.1.
- **Sensory feedback**– collection of sensory data about the environment based on user input, for example his position which determines the given view and perspective.
- **Interactivity**– responsiveness to user actions.

Those features of Virtual Reality draw it away from traditional media technologies and thus VE gives new possibilities which has never been considered before.

In this section it is worth mentioning other notions, closely related to VR, which are Real Environment (RE), Mixed Reality (MR) Augmented Reality (AR) and Augmented Virtuality (AV). For this purpose it is convenient to introduce the concept of Reality-Virtuality Continuum, which is presented below, in Figure 1.1.

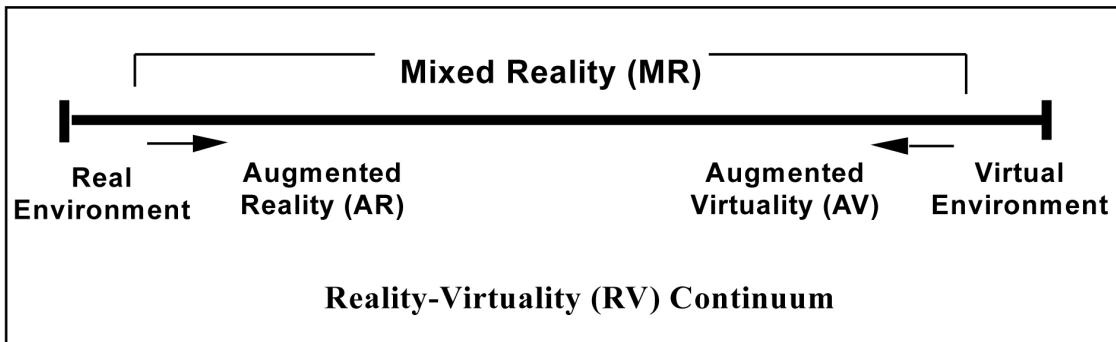


Fig. 1.1 Representation of Reality-Virtuality Continuum [5].

As shown, the Real Environment is an opposite idea to the one that was already explained, which is Virtual Reality. Thus one can understand it as an environment, which consists of the real objects exclusively, observed either in person or through some kind of display or window. Between those two extrema of RV, the Mixed Reality is placed. It is a general notion, which describes surrounding, where both real and virtual objects are presented together. While going into details, two subconcepts occur. First one, Augmented Reality, may be defined as "augmenting natural feedback to the operator with simulated cues" [5], what practically means that the user display is transparent, so it shows the view of a real world, but additionally enriched with coexisting virtual objects. *Ipsso facto*, Augmented Virtuality is very close to AR, as the only difference between them is stated by proportions of virtuality and reality presented to the user, what results from Figure 1.1. It follows, therefore, that those two concepts interpenetrate.

1.2 History

It may seem that Virtual Reality is an idea that originated with a recent rapid technological development. In fact, VR's origins date back to 1950s at the latest, when the first devices for that purpose were designed.

In 1957 the machine called *Sensorama* (Figure 1.2) was constructed by Morton Heiling. This multi-sensory stimulator contained prerecorded film in color and stereo (six to choose), binaural sound and, additionally, scent, wind and vibration effects. Although it was not interactive, *Sensorama* is considered as the first Virtual Environment system. Four years later, Philco Corporation produced another device- the *Headsight*, which was the first example of Head-Mounted Display (HMD) with motion tracking. It provided a vision of a film that was recorded in a real time by cameras, what makes it an Augmented Reality rather than VR system. Subsequently, in 1965, Ivan Sutherland, proposed the concept of *The Ultimate*

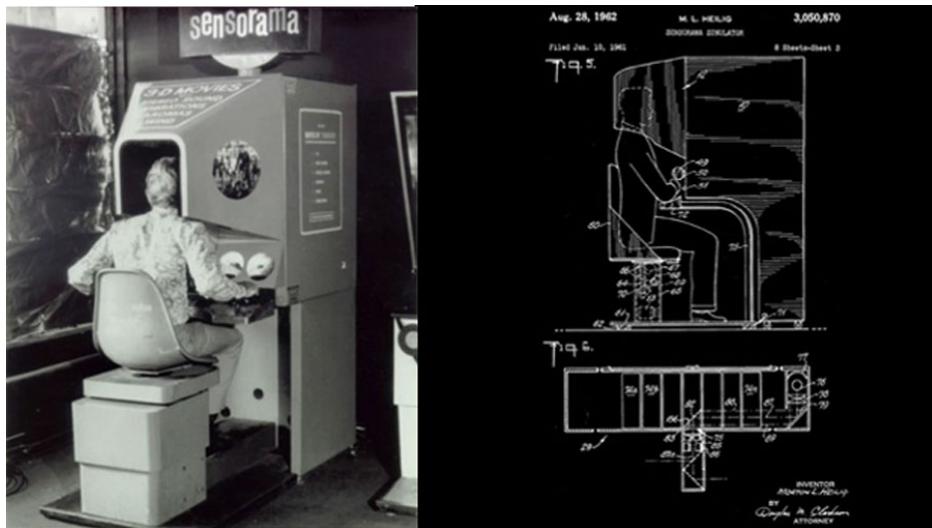


Fig. 1.2 *Sensorama*, patented in 1962 [6].

Display. The assumptions he declared, encompass the Virtual Reality idea until today. These were virtual world maintained in real time through a computer hardware, user's ability to interact with it and a tactile feedback, all with usage of HMD and augmented 3D sound. On this basis he created *The Sword of Damocles*, device with Head-Mounted Display and head-tracking. Although its large size and weight, it was invested in by NASA and CIA [6, 7].

1970s brought another new, innovative ideas, like *GROPE*- prototype of a force-feedback system or *VIDEOPLACE*- artificial reality, where position-tracked silhouette of a user was projected on a screen as an avatar. But the most rapid development occurred in the next decades, when the miniaturisation went hand in hand with expenses reduction. In 1985 and 1988, the VPL company introduced first commercially available VE devices- *DataGlove* and *EyePhone* HMD. Next, in 1989, Fake Space Labs manufactured *BOOM*, which was a small box with two CRT monitors creating the stereoscopic view and a mechanical arm measuring the position of a box, what allows the user move through a virtual world. Another concept, that did not base on Head-Mounted Display, was presented three years later as the *CAVE*-Cave Automatic Virtual Environment. Here, the stereoscopic images were projected on the walls of a room and could be interpreted properly thanks to LCD shutter glasses worn by a user. Comparing to HMD solutions, *CAVE* assured wider field of view and better quality of presented images. In the decade of 1990s the strongest interest was shown in games industry as a VR's application. This resulted into several systems, like *Virtuality*, *Sega VR* or *Virtual Boy*, which often showed poor image quality or even caused adverse health effects (nausea or headache) to the user [6, 7].

After the strong 3D graphics development, in 2012, young constructor Palmer Luckey created a new HMD-based device called *Oculus Rift*, that met with a great commercial interest. Two years later, Oculus VR company was bought for a 2,5 billion of dollars, what accelerated a present growth of Virtual Reality development [6]. Current used software and hardware are described in section 1.3 and 1.4.

1.3 Hardware and software

First of all, to understand how does the Virtual Reality really works, one has to become familiar with the basic components of those systems and their architecture. Generally, in VE systems, user is connected as part of the input/output loop, providing input through devices for interaction and experiencing the result of this input through variety of output, immersive devices. VR engine (computer) is responsible for processing loop information in real-time. Figure 1.3 presents the scheme of those dependencies.

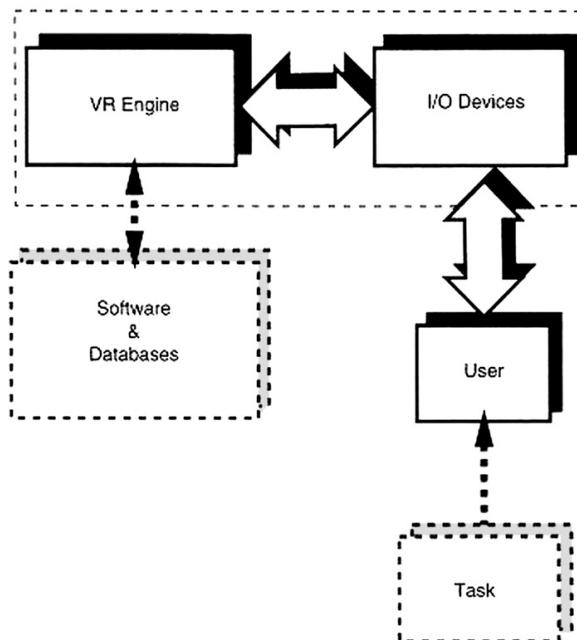


Fig. 1.3 Architecture of Virtual Reality system. I/O stands for input/output [8].

1.3.1 Immersion

Sensation of immersion can differ qualitatively, therefore, on this basis, Virtual Reality is divided into three groups [7]:

1. Non-immersive systems
2. Semi-immersive systems
3. Immersive systems

The first one which is called, interchangeably, Desktop-VR systems or WOW (Window on World), is a simplest type of VE. It does not employ special devices, since it is based on working of one or more computer screens (mostly monoscopic). It allows interaction of user, but does not give him an impression of immersion and barely blocks out the real

world. Therefore in some quarters it is not considered as Virtual Reality technology. Semi-immersive systems (Fish Tank VR) are improved versions of Desktop-VR, as they use not only screens, which are often stereoscopic in this case, but also head-tracking devices, that intensify immersion of a user. As they do not provide sensory feedback, they cannot be recognized as the third category of systems. Fully immersive systems, besides supporting stereoscopic vision, employ Head-Mounted Display which enables to, according to user input (position, orientation), change the view of the scene. Thanks to that, user may look and move around, so simply navigate in the virtual world which, additionally, is presented in full scale. Moreover, the immersion effect may be upgraded by sensory, auditory and haptic interfaces [7, 9].

Depending on the application and developer invention, user may achieve different types of immersion, which are listed below [7]:

- Sensory immersion- feeling of entering into the virtual world and being stimulated intellectually by it. Experience of time and space unity because of the fusion with image medium.
- Spatial immersion- occurs when the simulated world is perceptually convincing.
- Narrative immersion- experience of being gripped in a story presented in virtual world.
- Tactical immersion- occurs when user performs tactile actions that involve skill.
- Strategic immersion- analytical involvement associated with mental challenge.
- Psychological immersion- confusing the virtual surrounding with real life.

1.3.2 Sensory feedback- output

In simulating artificial world, the most important role plays stimulating user's senses. Thus, knowing the contribution of each one for information passed to humans brain, is very useful. Figure 1.4 explains why the visual part of VR has become a focus of research and why the sense of sight appeals to people imagination the most. Despite this fact, other senses are also taken into consideration of Virtual Environment systems research, what is presented below in this section.

Sight

Human vision is stereoscopic, as each eye can see one two-dimensional image which partly overlap. Those images are being processed by a human brain obtaining a parallax effect-visual phenomenon of depth perception, based on a difference in position of each eye. This phenomena is mimicked by devices simulating three-dimensional vision, the most important ingredient in Virtual Reality- they generate different images for each eye performing natural dimensions and high resolution.

The simplest gear to obtain stereoscopic vision are anaglyphic glasses. Each lens there has a different color, which acts as color filter, dividing an image in two sub-images. Another

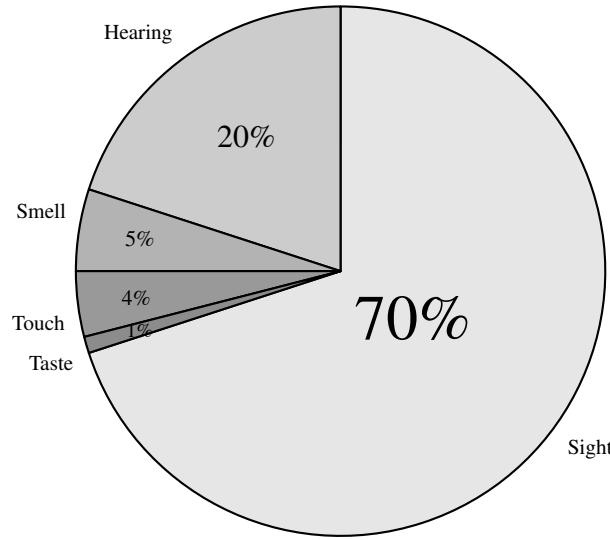


Fig. 1.4 Five human senses contribution in information transferring to a brain [10].

device is called shutter glasses, which have LCD screen in front of each eye. They are switched on interchangeably, corresponding with an image presented on a screen resulting in a 3D image [13]. Next example, Fish Tank VR systems, use 3D glasses and standard desktop monitor. In this case, stereoscopic images are created with usage of the off-axis projection, the method that generates two asymmetrical (not centered on the main projection axis) projections. Nevertheless, in the subject of Virtual Reality, the most often used devices nowadays

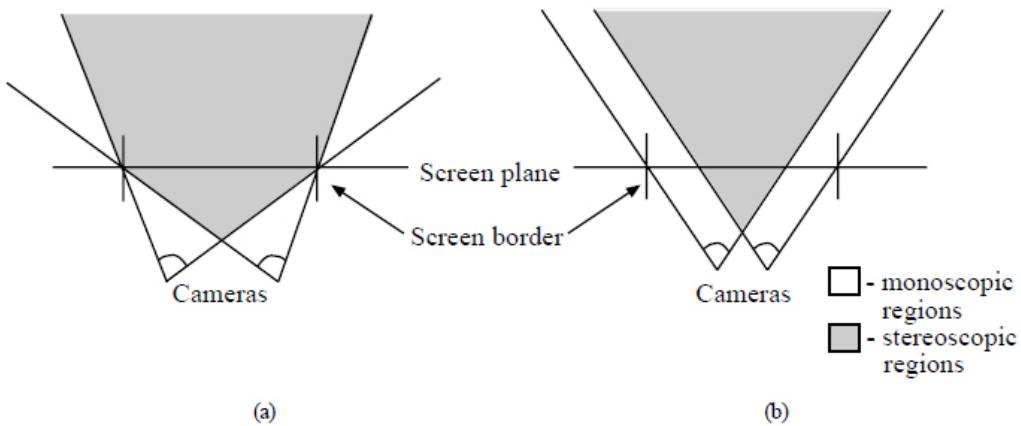


Fig. 1.5 Two centers of projection transformations: a) off-axis projection b) on-axis projection [10].

are immersive systems, like Head-Mounted Display, where different images displayed for each eye result in a stereoscopic effect. It may be obtained both by on-axis projection (when both image positions are on the symmetry axis of the view plane) or off-axis projection. The

latter one is often needed since the tracked head position is not defined only by the axis of view plane. In modern systems the choice is up to the application developer [10, 11]. The difference between those two approaches is shown in Figure 1.5.

Another issue discussing the vision is the field of view (FOV). Theoretically, human eye retains FOV of 180° both horizontal and vertical. Practically, this value is reduced to 150° because of the obscurement with cheeks, eyebrows and nose. Total horizontal viewing range binocularly overlaps in 120° (when focused on infinity). Since the task of Virtual Reality is to give sensations that are close to real ones, FOV is a parameter of high importance and it may be noticed in HMDs development. As example, current devices use the barrel distortion (Figure 1.6)- technique that distorts image and enables lenses to present wider FOV, mimicking spherical shape of the eye [14]. Typical headset in 1996 supported 40°-60° field of view, while *Oculus Rift* or *HTC Vive* in 2015 covered 110°. For comparison with traditional visualisations, a 21" monitor observed from distance of 50 cm reaches 48° [10, 12].



Fig. 1.6 VR Demo rednnered in stereo with *Oculus Rift* using barrel distortion [14].

Hearing

When vision does not provide enough information, auditory channel grows in importance. It may give an extra information (enabling perception of data outside the FOV) and may help in spatial orientation. Indeed, 3D sound in VR increases the level of immersion by simulating distances, directions and material information about the surrounding.

Spatial hearing in humans occurs by appraising both monaural (the same for both ears) and binaural (different for each eardrum) clues. The latter occur, if the sound source is

placed outside the median plane- the distance between it and each ear is different, causing interaural differences of time (ITD) or intensity (IID). Those phenomena are mainly used for directivity perception, which is an accurate tool (up to one degree), but only works in case of laterally localizing. On the other hand, monaural cues provide information about elevation: amplifications and attenuations in the frequency bands (intervals in frequency domain). These clues are needed in front/back localization. To recognize the direction in this case, humans rely on spectral modifications of sound caused by the shape and size of a head, neck, torso and outer ears. It is possible since the sound interacts with the body geometry differently, depending on its direction. All these subtle signals received by humans brain through the ears generate the effect of spatial sound perception [10, 15, 16].

Thus to create a good immersion audio effect, one has to take into account not only the dependencies presented above, as they refer to directional recognition only. Introducing sound changes while moving past listener (doppler shifts), considering environment geometry, distances (usually as a consequence of loudness) and material properties of objects (which may generate echoes) should also be done. Finally, sound and visual events need to be synchronized [10, 15]. Therefore, to achieve successful acoustic simulation, three basic steps are required [10]:

1. Sound generation- creating sounds, either by sampling or synthesizing, needed in the virtual world.
2. Spatial propagation- calculation of movement of sound waves through the environment.
3. Mapping of parameters- mapping calculated parameters from second step onto sounds that will be delivered through headphones or speakers to the user, with consideration of monaural and binaural clues to achieve the directional effect.

According to the *Oculus Rift* documentation, to accomplish second and third step in *Oculus* device, the Head-Related Transfer Functions (HRTF) referring to directional recognition are applied. HRTF data may be obtained both experimentally (from human tests) and by head model simulation. Since they are not perfect and often need interpolation of data, mimicking the real spacial sound system is still in a field of study [16].

Smell

Although olfactory simulation is not commonly used in Virtual Reality, it may play an important role in some applications of VE, like surgical simulations or simulations designed for emergency medical personnel operating in the field. Human can detect smells in concentration from one part per million up to one part per billion. Additionally, it is much easier to recognize increases than decreases in concentration, therefore VR hardware with olfactory output should enable diffusing odours when needed and, when it is not longer required, purify and filter the air.

The method that makes delivering odours possible is simply odourant storage. They can be stored as liquids, gels or waxes and usually are microencapsulated or compressed on flat surfaces. Metering of the exact amount smell is done by capsules scratching. The whole

system should be miniaturised and have low power requirements, as the most convenient would be placing it in HMD [15].

Modern olfactory devices for VR are still in the testing phase. One of them, called *The FeelReal*, is supposed to work as an add-on to different kinds of HMD available on the market. It does not only support seven smells (which may be chosen while buying the whole system), but also its temperature, generating hot (e.g. mimicking fire) or cool (e.g. mimicking cool water) scented air [17].

Touch

The last sense discussed in this section is touch, as it may be introduced both by output and by input devices. Although still not commercially popular, they can be useful in a wide range of applications, enabling the introduction of new features and improving the immersion impression.

Haptic sensations experienced by humans may be divided into two groups. First, called kinaesthetic or force feedback, stands for forces sensed by muscles, joints and tendons. The second group, tactile feedback, means touch, temperature, texture and pressure feeling on the skin surface. Generally, both types of sensing occur simultaneously.

Tactile simulation may be generated in different ways, like with usage of pneumatic systems, piezoelectric crystal, shape-memory alloy technologies, electrical stimulation etc. It is also possible to provide temperature feedback. On the other hand, kinaesthetic forces, which may be introduced both as an input or output, are provided using three components: first, measurement of a movement and forces exerted by user, second, calculation of their effect on virtual object together with resultant forces, and third, presentation of those calculated resultant forces to the user. It may be achieved inter alia with using of electrical stimulation, electromagnetic motors, hydraulic or pneumatic systems [10, 15]. As it is already proven that force feedback increases efficiency of manipulation and placing tasks, it broadens the spectrum of possible applications of VR.

Although the most obvious type of device used for providing haptic stimulation is glove or hand-controller, recent reports show other possibilities, like shoes or even whole suits. Despite of their form, all of them contain large amount of sensors and trackers. Albeit still under development, some already presented examples provide user's hand visualisation on the visual display [18].

1.3.3 User's input

Input devices enable interaction with virtual world and, for the best immersion effect, they should be as intuitive and natural as possible.

Position and orientation tracking

Positional tracking is one of the most essential components in Virtual Reality. Usually it refers to head tracking, but it may be also introduced by e.g. object or hand tracking (the latter was shortly described in section 1.3.2). Regardless of the type of tracked object, all

of them have six degrees of freedom (DOF): translation coordinates (in x, y and z axis), which refer to position and rotation angles (pitch, yaw and roll), which refer to orientation. Tracking systems should provide data describing as many as possible. Generally, one can divide tracking devices into two categories: those, that deliver total position values (absolute data) and those, that achieve that relatively to, for example, last state. Another division is based on the technology that tracking mechanism may use and the most common examples are described below.

Magnetic tracking relies on measuring strength and angle of magnetic field. To achieve this goal, transmitter , that generates magnetic field, receiver, that picks it up and control unit, that calculates position and orientation of a sensor, are needed. Transmitter, known also as emitter, may generate alternating or direct current excitation and receiver determines distance and rotation on the basis of strength and distribution of the measured field. Despite of the good accuracy, magnetic trackers have crucial disadvantage, which is possibility of interference not only from magnetic fields generated by other devices but also from conductive or ferromagnetic materials, which can occur nearby.

Inertial tracking systems make use of accelerometers and gyroscopes, which measure linear acceleration and angular velocity, respectively. Since those measurements are integrated afterwards, obtaining velocity, position and angular position, inertial tracking method is not considered as accurate, due to error which may occur with those calculations, especially for small changes of position.

Optical tracking devices work basing on one or more camera, which detects measured item thanks to pattern recognition. They may use set of markers placed on the object like HMD (in a geometry which enables to recognize whether it is rotated or not) and algorithm, which compares known marker position and the one obtained from a camera, resulting in position and orientation measurements. The algorithm should also be able to fill gaps in the data in case of marker's view obstruction. Typically, markers are constituted by infra-red lights that flash periodically, or retro-reflectors, which reflect the light back from the source of IR (e.g. camera), as this wavelength prevents interference with other activities. Other types of optical tracking systems use predefined patterns (which may be visible markers or known 3D models).

Acoustic tracking is based on measurements of the time that acoustic signal emitted by transmitter needs to reach a receiver. Obtained values are subsequently processed resulting in a distance that wave, which is typically ultrasonic (above 20 kHz), has covered. Since the data may be received between two points only, usually acoustic tracking systems contain multiple emitters and multiple microphones placed on tracked object, and the rule of their arranging is similar to the one applied to markers in optical tracking. Nevertheless, this solution is susceptible to noise interference and errors caused by differences in speed of sound in the air or by echoes.

The last of presented types, **mechanical tracking**, is one of the oldest. It uses joints connecting the remote object and a point of reference. This construction together with, for example potentiometers, allows to measure position values. Although mechanical tracking is resistant to any kind of interference, it does not allow the user to move freely due to mechanical restrictions [10, 15, 19].

All kinds of technologies can be combined to create Virtual Reality system that fulfils required standards. In resulting combination, introduced approaches may be responsible for different types of tracking and cover their disadvantages mutually.

Eye tracking

Direction, at which the user's eyes are pointing, is not always identical with head position. Therefore, many devices are equipped with an eye tracking system. Moreover, it may decrease rendering costs, as the image resolution does not need to be the highest in places out of the line of sight.

Approaches used to introduce eye tracking obviously differ from those responsible for position and orientation tracking. First, optical systems, work basing on reflections of beam of light (e.g. IR LED) from the convex cornea surfaces analysed by photo-transistors. Due to differences in user's eyes shape, tear fluids and corneal astigmatism, this type of eye tracking system requires complex calibration. The second approach is image tracking, that uses camera and image processing to determine user gaze. Next, electro-oculography (EOG) solution, use the fact that retinal epithelium generates the standing potential between cornea and retina. It can be measured by electrodes placed next to the eyes, but it may be disturbed by electric interference. The last approach, electromagnetic, is based on measurement of voltage induced magnetically on a coil attached to a lens on the eye [10, 15].

3D and desktop input devices

To move, select or modify objects in virtual world, additional input devices may be introduced. To this group belong gloves, mentioned in section 1.3.2, more dexterous manipulators or controllers, sensor devices (like *Leap Motion*, which is described in 1.4), joysticks but also simple desktop mouse or keyboard. While 3D devices are intuitive, they provide 6 DOV control and do not affect negatively on immersion effect, two-dimensional ones, like desktop devices, are simple and easily accessible [10]. Nevertheless it is worth mentioning, that using the latter is connected with partial detachment from virtual world and causes problems with using while wearing HMD which covers vision of reality.

1.3.4 Software

To create the illusion of presence in virtual world, convincing models and simulation should be performed. Nevertheless, good resolution and huge physical calculations demand more resources, what increases computational cost. Although current CPUs (Central Processing Units) and GPUs (Graphics Processing Units) are constantly developed and become more and more powerful, VR applications require quality that mimics the real world, therefore their demands also continue to increase. Despite this, it is important to remember, that high computational cost may affect overall performance e.g by hindering interactive framerates. Therefore used algorithms and data structures should be constantly developed to receive as good quality and as low computational expenses as possible.

Level of Detail (LOD) is an important facet of data structure construction. It allows to define accuracy of geometric representation, due to perspective of an object, its movement, importance or eye tracking (mentioned in section 1.3.3). From the algorithm point of view, it is also worth remembering to manage memory properly and, for example, load objects in advance, before they are visible. Besides modelling objects, laws that rule the physics of presented environment should also be introduced. It may mimic the real ones, like Newton's laws, or be completely different. As physical phenomena are very complex when coming into details and they need to be performed in real time in VR, usually their subset should be applied.

What is more, is that developer of VR application may introduce different ways of interaction of the user with virtual world. Scene view may differ depending on application and presence of head tracking. Navigation of user could also vary, and may be hand directed, gaze directed, performed through input devices or with virtual controls. Similarly selection manipulation of objects and many other interaction possibilities can change and be performed in different ways.

Finally, to render an image of a created world, the following calculations of display transformations need to be done: eyes position in the tracker's sensor, position and orientation of a sensor in a tracking system, position and orientation of tracking emitter in the room as well as position and orientation of this room in the virtual world. Moreover, the stereoscopic image needs to be generated, using either on-axis or off-axis projection (described in section 1.3.2) and proper image distortion. Additionally, models present in the virtual scene need to be rendered, including their performance (amount of polygons per second), shading, lighting and texturing capability [10].

Engines and frameworks

Nowadays large amount of software allowing VR creation is available for developers. It can be divided into four groups [14]:

- **Native Software Development Kits (SDK)**- the lowest level of development, device drivers and software libraries used with operating system's libraries.
- **Game engines and frameworks**- known as middleware, already cover rendering, physics and interfacing to devices. Often provide set of tools- integrated development environments (IDE).
- **Web browsers**- developing VR with usage of web technologies with its whole infrastructure (data, hyperlinking between VEs etc.).
- **Video players**- enable creating virtual world, which is not fully interactive, from recorded video.

1.4 Examamples of hardware

Thanks to recent rapid evolve in a field of Virtual Reality (mentioned in section 1.2), more and more manufacturers start producing devices for that purpose. They differ on the details, but all of them take part in a technological race, where the prize states the biggest commercial success. Nevertheless, for the purpose of this work, only two of devices available on the market are described. Those HMDs are considered as leaders, and they are *Oculus Rift*, which is a desktop VR device and *Google Cardboard*, which becomes mobile VR equipment in combination with a smart phone. First one, as previously discussed, is a precursor in modern HMD development in which the high hopes are placed. On the other hand, *Google Cardboard*, with it's simplicity and low cost, provides a convincing argument that VE will soon become a part of humans everyday life.

1.4.1 *Oculus Rift*

Generally, *Oculus Developer Kit* device in version *DK2* (Figure 1.7) consist of stereoscopic display with a head tracking sensors, position tracking camera and HDMI wire splitting into HDMI and USB cables, which connect it to the computer. It may be used with both, laptop or desktop, on either *Windows*, *Linux* or *Mac OC* system. However, in case of GPU, described device is not that peripheral, as it has high-end graphics card requirements, what induces development in this department as well. *DK2* was released in 2014 as the successor



Fig. 1.7 The *Oculus Rift* *DK2* Head-Mounted Display and position tracking camera [14].

of *DK1*, with better resolution (1920 x 1080 pixels, 960 x 1080 per eye) and positional tracker, which was not introduced in a previous model. Therefore, it enables the user not only to look around, but also to move around a virtual scene. For the head tracking purposes, *Rift* combines inertial and magnetic tracking in so-called Inertial Measurement Unit (IMU) what results in a 3 DOF effect. 6 degrees of freedom are introduced by optical tracking, with usage of micro-LED markers on the headset and IR camera. Although this solution improves

immersion effect, it is still not perfect, as the user needs to be in front of the tracking camera, what limits his area of movement. This issue is going to be solved soon with introducing third generation of *Oculus*, called *Crescent Bay*, which has 360-degree head tracking and higher resolution of an image.

What is more, *Oculus* producer provides needed software, *Oculus Runtime* and *Software Development Kit* (SDK) for developers sake [14, 16]. Additional facilitation constitutes availability of engines that are strongly focused on developing applications for *Rift*, like Unity3D.

1.4.2 *Google Cardboard*

Introduced in 2014, *Cardboard VR* hastens Virtual Reality experience possibility for an average consumer. Since it makes use of almost every smart phone (which are owned by a large part of society nowadays), it allows to immerse into VE without expensive hardware- neither HMD nor computer. *Google Cardboard* is simply a box with two lenses, which should be set by a user (Figure 1.8). The stereoscopic image is created through the Cardboard-ready



Fig. 1.8 The *Google Carboard* VR viewer [14].

application on the phone, which should be placed inside the device. It provides 90° FOV and head tracking, which is generated by the phone's accelerometer and compass. Obviously, impression of immersion in this case is much smaller than in previously described device, but availability of this solution makes it more and more popular. In number of applications as well- hundreds of them are already accessible.

Similarly to *Oculus Rift*, software for developers is also provided in this case by a producer. It exists in two options- an Android SDK and an add-in to, already mentioned, Unity3D engine [14].

1.4.3 *Leap Motion*

Although it is not a Head-Mounted Display, like the previous examples, *Leap Motion* provides user's input tool which is suitable for VR purposes. As this example of sensor device plays the key role in this work, it is described here as the only one of its kind.

Presented controller is a small peripheral USB device that uses two CCD IR cameras and three LED emitters, which generate IR light. It tracks the position of predefined objects like palm and its finger tips in Cartesian space, relatively to the center point of the device, which is placed at the position of the central IR emitter [20]. Due to the difference in positions in which the controller may be used, it is worth mentioning that it employs right-handed Cartesian coordinate system, which is shown in Figure 1.9.

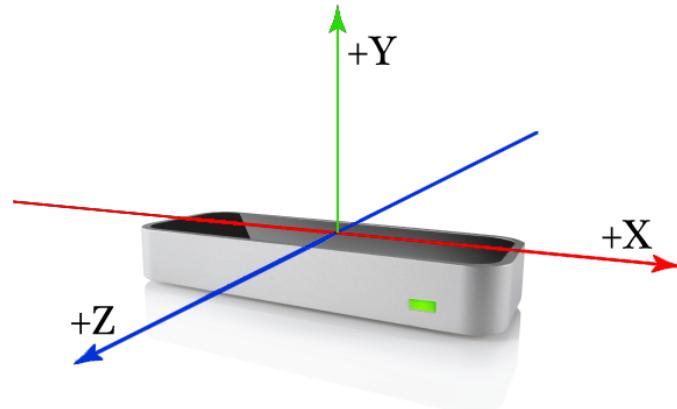


Fig. 1.9 *Leap Motion* controller and its coordinate system [21].

According to the manufacturer data, the accuracy of the sensor in detection of fingertip position is approximately 0.01 mm, and the cameras frame rate varies from 20 to 200 fps depending on the user's settings and available computing power. *Leap Motion* may be placed on a physical desktop or mounted onto Virtual Reality HMD. Its field of view is stated as 150° with a volume of approximately 0.23 cubic meters and the effective range varies between 25 and 600 mm above or before the device.



Fig. 1.10 *Leap Motion* in use- hand models on the screen are corresponding with the tracked real hands of the user [21].

The *Leap Motion* API (Application Programming Interface) measures distance, time, speed and angle of the objects recorded by cameras. The software uses an internal model of

human hand, what enables predictive tracking in case of temporary visibility disturbances. Moreover, the model contains all anatomical finger bones. Thanks to that, visualisation of hands in the three-dimensional VR scene is more immersive, as the model's movements are very close to its real equivalent (Figure 1.10). The most recent hand tracking software release, called *Orion* indicates variety of improvements, dedicated mostly for VR purposes. Additionally, for the developer's use, *Leap Motion* offers its SDK with libraries needed for the application development. It is available in several languages and enables integration with Unity3D engine.

1.5 Possible applications

There is a large variety of applications in which Virtual Reality is and may be involved. List of possibilities constantly grows with the VE development, where entertainment constitutes the group of one of the strongest commercial potential. Therefore, many developer tools are mainly focused on creating video games, which may become more and more involving thanks to or because of VR. What is more, it creates a new way of thinking about social media and web browsing. The same happens to live entertainment- some of the famous artists have already broadcast their shows in the VE versions, making attending it possible to larger group of fans.

But not only amusement makes Virtual Environment such an interesting and unique platform. Another industry is widely understood education. By providing interactive learning, VR may contribute to the increase of effectiveness in educating process. Thus simulations, such as medical training scenarios, scientific laboratory, flight or military reproductions, which are already created and developed, may become indispensable part of different training processes, preparing for the most unusual cases in a safe and inexpensive way. Moreover, simulations that simply create 3D VR representations may also be a valuable tool, immersing the user and facilitating understanding of presented image. That issue may be used in scientific sector, what is strongly connected with the topic of this work and described later in Chapter 2. What is more, medical applications of VR are not only connected with the personnel's education. Creating the virtual world provides a platform for a new type of therapies, especially psychotherapies (like anxiety treatment) or even diagnosis, comparing reactions of possible patient with a healthy group. This topic is of great interest in a scientific world nowadays, resulting in a number of publications describing new treatment ideas.

Another field where VR is already applied is an architecture together with real estate. Creating visualisations in three dimensional view is not only a feature for designers purposes, but most of all to customers. They may experience their presence in presented properties quickly and easy, both by watching projects or video presenting already existing places. This function is also used in tourism, where showing hotel room or aeroplane interior in VR may be important for the sale success. Additionally, for those who cannot travel, VE may offer tours without getting out from their home. [14, 22]

Examples presented here are just a few of the many other possibilities, that are constantly created. In all of them Virtual Reality facilitates or even enables humans life, and with the growing availability of VE gear it soon may become part of their every-day reality.

Chapter 2

Visualisation of Molecular Structures

Modern biotechnology, and thus biochemistry, genetics, molecular biology and bioinformatics as well, is based on understanding the mechanisms of molecular actions. Therefore, one of the most important information needed is a structure of molecules, such as proteins, nucleic acids, lipid layers etc. What is important to stress, is that in case of biological molecules not only general sequence is of great interest. Sequence of amino acids, in example, gives one only information about primary structure of proteins. What is meaningful as well, is their secondary (forming of α -helices, β -sheets and turns), tertiary (spatial structure) and quaternary structure (combination of several protein chains), as those features enable protein molecules to perform their various tasks. Generally, these data make simulations and biomolecular engineering possible, enabling explaining of biological phenomena and creating new ways of influencing them.

2.1 Biomolecular Structure Determination

To create the simulation of biomolecular mechanisms, one can use structural data of certain molecules, previously obtained and now collected in available data banks. Nevertheless the knowledge of techniques used to retrieve these data is useful to know their limitations and be able to choose the most adequate one. Therefore, major methods used for that purpose are listed and described below.

2.1.1 X-ray crystallography

Currently, X-ray crystallography provides the most detailed atomic structures. The whole process consists of three main steps: first, obtaining and growing crystals of a pure molecule, second, placing it in a beam of X-rays, which is diffracted into a pattern of spots, and third, computer analysis and interpretation of arisen map of electron densities in the crystal lattice. The result depends mainly on quality of crystals, which affects the resolution of created image. Described method enables obtaining high-resolution data- best crystals may result in a map providing image of elements separated by less than 1 Å. Though, in the vast majority of studies the resolution of 1.5 Å up to 3 Å is reached. While the resolution of 1.5 Å enables

distinguishing each atom in a structure, the 3 Åresolution is harder in interpretation and usually requires additional knowledge of covalent structure of the given molecule. In these cases the biggest difficulty state mobile areas of a molecule and regions on its surface, and thus the interpretation is not unambiguous. To indicate the level of trust of the result, one introduced temperature factors, known also as B-values, which provide information about the disorder of atomic positions. B-values control the shape of Gaussian distribution, which is used for describing model atoms around an atomic center. Therefore, value of temperature factor, which is equal to 10 stands for an atom, which position is sharply described, whereas 30 or more indicates high disorder of the interpreted position [23].

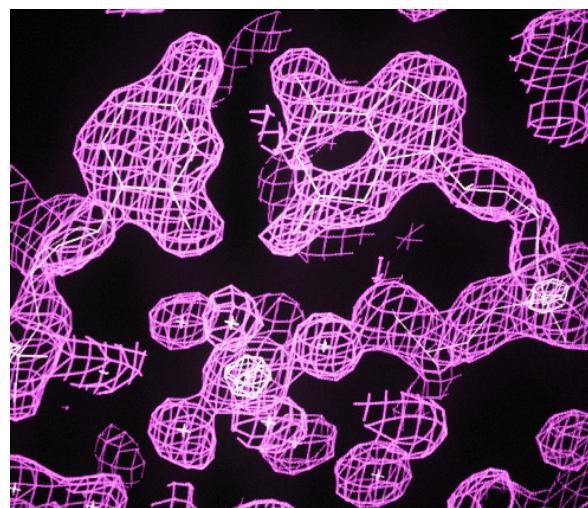


Fig. 2.1 Electron density map of DNA crystal [23].

Besides problems with resolution, it is important to remember that the highest limitation of X-ray crystallography is simply its need of crystals, which represent one of the possible conformation, conceivably (but not necessarily) different from the one occurring in natural environment. Therefore, to overcome this issue, crystals obtained under varied conditions must be studied. This approach makes studying functional aspects of biomolecules possible.

2.1.2 NMR Spectroscopy

Another way of obtaining structural data of biomolecules is nuclear magnetic resonance (NMR) spectroscopy. It provides information about local environment of atomic nuclei. It is possible thanks to the fact, that atomic nuclei have magnetic moment, which may be aligned in a strong magnetic field. This state may be disrupted by a radio frequency pulse, which exhibits proper wavelength. What actually is measured by a spectroscopy, is the radio frequency radiation which occurs during relaxation of atomic nuclei to the basic state. It is strongly influenced by a local environment of the atom, therefore obtained data gives an information about distances between atomic nuclei and their local conformation. Usually, atomic model is developed by using a list of constraints concerning pairs of atomic neighbours and bonds between them. Its interpretation results in an ensemble of structures,

which may represent all possible conformations that molecule might adopt freely in a solution or the whole range of conformations, where only one describes the actual model. In case of proteins, they differ mostly in side chains, retaining main protein chain structure.

Huge advantage of NMR technique is its applicability for molecules in a water environment, what is convenient while determining structure of biomolecules. Nevertheless, NMR spectroscopy may be used only for small proteins and nucleic acids. Even improvements in this method, such as usage of multiple radio frequency pulses which perturb multiple nuclei, enables to characterize the structure of a protein with a maximum length of 250 amino acids [23].

2.1.3 Electron Microscopy

Electron microscopy methods state the approach that is the most intuitive. Nevertheless it allows to observe instead of single atoms in a molecule, only its general morphology. This issue arises from practical problems that occur while applying the technique, like imperfections of optics, problems with specimen preparations and low contrast. Therefore, the exhibited resolution is not higher than 2 nm. However, electron microscopy is often applied combined with other methods, like X-ray crystallography, NMR spectroscopy or molecular modelling. It is extremely useful not only in case of large molecules, but also to observe changes of biomolecules in different conditions.

Transmission Electron Microscopy

Transmission electron microscopy consists in illumination of a thin sample by the electron beam, whereas the microscope resolves the relative transparency of specimen regions. This method allows even to obtain a three-dimensional image by tilting the sample at a range of angles and combining resulting images (so-called electron tomography). Though, the technique is not perfect and additional proceedings are needed to improve contrast. One of them is staining the specimen with salts of heavy metals, what on the other hand may introduce artifacts. To counteract it, one may freeze the sample (cryoelectron microscopy), but it brings new problems with contrast. Thus, commonly applied approach is analysing and averaging of separate particles, which merged together give an image of whole molecule with good contrast.

Scanning Electron Microscopy

Scanning electron microscopy slightly differs from its predecessor. In this approach, sample is fixed and dried and then coated with a thin metal (often golden) layer. Thereafter it is scanned with a narrow electron beam, which are scattered or emitted from a surface. This process results in a 3D image, but with relatively low resolution of 10 nm, introduced by the metal coat. Nevertheless it still may be used in research of large assemblies of biomolecules [23].

2.1.4 Atomic Force Microscopy

The last method of determining biomolecular structures is the atomic force microscopy. It results in a topographic map of a surface of the molecule. Sample is moved under a tip both laterally and vertically (what is controlled by piezos scanners) applying a constant force on a tip. Detection of forces between specimen and cantilever is done by illuminating a laser beam on its back. Occurring reflection in different places is then caught by photo-diodes. Moreover, to reduce high shear forces that arise between tip and a sample and to prevent the latter one, the tip works in a tapping mode. Thanks to that oscillating movement, contact is shortened what minimizes shear forces. Additionally, the whole apparatus is placed in a solvent, what enables mimicking biological, natural environment and therefore give the real structure of biomolecules.

Atomic force microscopy is a really accurate tool. Its resolution depends on the sharpness of a tip, though usually is in the range between 0.1 and 10 nm. What is more, it may be applied for measuring forces between molecules or even forces that conduct protein folding [23].

2.1.5 Databases

There are many kinds of bioinformatical databases, providing information about nucleotide and protein sequences, protein families, structures, their functional annotation etc. For this work the most important type constitute structural databases, therefore they are shortly described in this section.

PDB

Protein Data Bank (PDB) database was established in 1971 in USA as the tool needed for crystallographers to freely exchange protein structures between laboratories. Nowadays PDB constitutes the greatest collection of spatial structures of proteins in the world, especially since 2003, when it has been united with European Macromolecular Structure Database and PDB Japan.

In the described database one can find structures obtained with a large variety of methods, mainly by X-ray crystallography, NMR spectroscopy and electron microscopy, which were described earlier. Before publishing of the results sent by laboratories, structure is checked for correctness and data format. Data may be presented in three kinds of files: PDB, mmCIF (macromolecular Crystallographic Information File) and PDBML (Protein Data Bank Markup Language). In the end, the structure is tagged and becomes its unique identity number.

It is worth mentioning the details of PDB data format, as it is mainly used by visualisation software. First of all, those files employ so-called method of chemical rules in generating bonds between atoms. Practically, it means that there is a need of tables with types and typical lengths of bonds between certain atoms. Thanks to that, for example, if two carbon atoms are 1.5 Å apart, it is assumed that they are bonded with single bond and thus the amount of information in one file is reduced. On the other hand, this solution is problematic in unusual cases, when such estimations are incorrect. What is more, data in the PDB files

is divided into two groups. First one, explicit sequence, is marked with the lines with a SEQURES symbol, where one can find sequence of amino acids in certain chain of protein. However, to reconstruct the whole structure it is not enough, therefore implicit sequence is needed. In this type every single atom is represented separately with its position in Cartesian coordinate system in the space (x, y, z), its name, number in the structure, the number of the residue that atom belongs to, number of possible conformations etc. Those lines are assigned with an ATOM (usually for atoms belonging to nucleic acids or proteins) or HETATM (for small molecules) symbol [24]. The whole format description and the contents guide is found on the website of Worldwide Protein Data Bank.

MMDB

Molecular Modelling Database (MMDB) which belongs to National Center for Biotechnology Information (NCBI) is strictly connected with the PDB structures, but it is extended with Entrez system, which enables the access to all biological data concerning the molecule. In this case, PDB file is translated to the ASN.1 (Abstract Syntax Notation) format, where one can additionally find such information as unified definition of secondary structure or the spatial domain structure. In contrast to PDB files, ASN.1 contains data about every bond specifying its category (explicit bonding approach). Nevertheless it uses so-called dictionary of the residues, where atoms and bond of 20 amino acids and 8 nucleotide groups are described [24].

2.2 Representation of Structures with Computer Graphics

To explore and understand the working mechanisms of biomolecules, the already known structure needs to be visualized. Obviously, direct imaging in this case is impossible, as molecules are significantly smaller than the wavelength of visible light. To overcome this issue, a large variety of representations have been developed and are useful for different purposes. All of them should have two things in common: clarity - being understandable for the user and maintaining the properties of a molecule that are directly related to the nano-scale world. Here, a few most popular ones are described.

First one, which is so-called "Ball-and-Stick" representation is the most fundamental one, presenting atoms as "balls" and bonds between them as "sticks". Although this approach is typical for education purposes, it does not provide relevant information. Most of all, it is misleading in case of the volume that molecule occupies, as the spheres are smaller than atomic radii to obtain the clearer view. Certainly, bonds presented as sticks give a comprehensive image for the user, nevertheless it is distant from the reality. Secondly, wireframe diagram (Figure 2.2 A) is simply representation of covalent bonds between atoms, presenting them as lines. It gives, similarly to "Ball-and Stick" approach, a view for the geometry of a whole molecule and finds an application in investigating linkings of proteins with cofactors or structures of active sites. On the other hand, spacefilling view (Figure 2.2 B) represents each atom as the relative sphere (for example Van der Waals sphere). This approach is the most relevant for size and shape representation, what is needed for

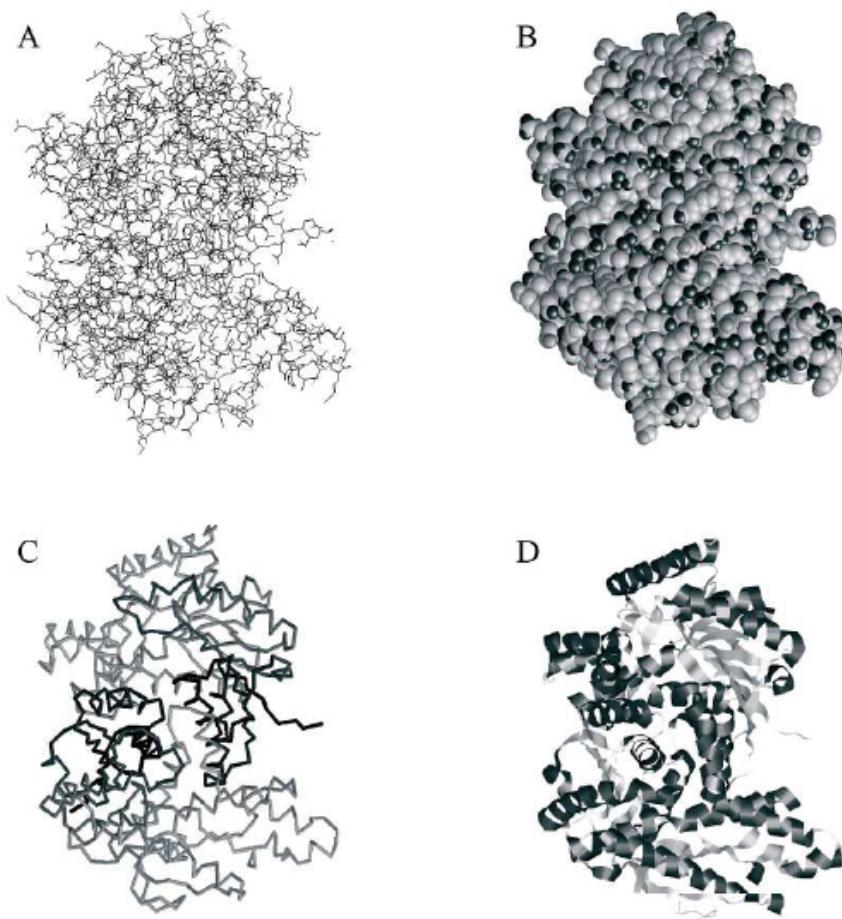


Fig. 2.2 Lactate dehydrogenase presented in four ways: A- wireframe (bond diagram), B-spacefilling, C- backbone (ribbon), D- secondary structure (cartoon) diagram [24].

visualisation of interactions between molecules. Then, ribbon diagram (known also as backbone) shows the topology of protein chain to the user and it is based on linkings between α -carbons of each amino acid. The last representation, called "cartoon" is focused on the secondary structure of the protein, presenting α -helices as twisted ribbons and β -sheets as arrows. Those two last examples (Figure ?? C and D) are commonly used for topology, protein folding analysis and its comparison between different molecules [23, 24].

Modern software used for visualisation is able to read the structural data and image it in a way that user needs. Obviously, to fulfil their task and be useful for users, such programs need to make rotation and zooming of the models possible. Moreover, additional features are helpful, such as measurements of distances and angles between atoms, identification of certain features of the structure and colouring, which significantly increases clarity of presented view. Generally user may choose from a large collection of colouring systems. In example, CPK, where each type of atom is coloured differently (carbon- grey, oxygen- red, hydrogen- white, nitrogen- blue etc.), shapely colouring, where each amino acid or

nucleotide has got a different shade, chain, which, instinctively, differentiates each chain, group, where C- and N- or 3' and 5' ends are coloured differently, temperature, which helps to identify vibration degree of freedom for each atom etc [24]. What is more, some of the software is not only visualisation tool and often provide a large engine for molecular modelling simulations, like program called VMD. On the Protein Data Bank Website, one can find a list of 41 programs used for biomolecular visualisations, some of them with open source or academic license.

2.3 Virtual Reality in Visualisation of Structures

Nowadays, a large amount of model organisms proteins have been researched and their structures are currently available in the data banks. This improvement coincides with the lastly observed computer processing power and memory growth. With addition of steady decrease of hardware prices, those occurrences create a great conditions for development of biomolecular visualisation, which leads to application of Virtual Reality for chemical and biological sciences.

2.3.1 Motivation

The most important investigation which should be done, is considering, whether engaging VE to bioinformatics is worth trying, and if so, what are the potential benefits from that solution. To answer these questions, it is convenient to describe issues that currently scientists have to deal with. First of all, for better understanding of proteins functions, their interactions and conformational changes should be known, and that is possible with the dynamically explored visualisation. Although classical approach results in still improving programs, it creates the 3D representation in the two-dimensional space, where depth perception and immersion is often not sufficient. Moreover, when several models are needed to be imaged at the same time, the view becomes unclear and structure manipulating becomes more difficult. Problem appears as well when small changes occur, such as in transient state interactions, which should be easily observable for better phenomena explanation. Furthermore, in the bioinformatics field, the human factor still plays an important role, for example in the structure interpretation and solving (often by comparative modeling) or in receptor docking simulations, when the algorithms cannot perform sufficient visual pattern recognition. Those subjects are highly important in drug design. For example, if the protein binding site is unknown, searching for possible sites need to be performed by experienced researcher and this case requires as best graphical representation of a structure as possible. Additionally, thanks to precise observations of a protein binding site, scientist may form a good hypothesis for ligand design, what is one of the most interesting tasks of molecular modelling [25].

Those issues can be overcome or at least become easier thanks to Virtual Reality, due to its immersion effect, its intuitiveness and possibility of real-time visualisations creation. Three-dimensional image enables direct manipulation of models and interactions between them. It facilitates observation from different points, looking around and distinguishing of each structure. Although this idea does not mimic any known real surrounding, it creates

an exact representation of nano-reality and enables discovering its features. Molecular modelling, molecular dynamics and bionanotechnology tools may be implemented in VR environment, creating a powerful combination that facilitates a large variety of bioinformatics tasks.

2.3.2 Examples

The idea of combining Virtual Environment with biomolecular simulation is not new. Already described in Chapter 1, *GROPE* and *CAVE* systems were used for those purposes in 1990s. Both were applied to drug-enzyme docking procedure, while the latter was a visualisation system, which actually had its input in discovery of new molecular features. Nevertheless, it was really expensive and, what is more disturbing, it did not give the user a reference point where he or she is in the virtual world. It hinders image manipulation and gives user a feeling of disorientation [10, 26]. Then, in 2000, Abraham Anderson and Zhiping Weng created program called *VRDD*, which was actually a drafting table-style projection system with additional wand with buttons, which was a program controller and remote. Additionally, the concept allowed to use a keyboard for more complex tasks. *VRDD* could present different display models, enabled docking simulations, real-time calculation of binding free energies and searching for binding sites by restricting it to possible regions [25]. Next, the very specific tools for VR were made, like the one designed for nucleic acids only, which can create long chains and therefore generate huge amount of data [27], or one for bionanorobotics prototyping, which main task was to invent protein-based molecular machines with usage of molecular dynamics methods [28].

The last example constitutes *Molecular Rift*, the program implemented in 2015 for *Oculus Rift* by Magnus Norrby et al. What is more, it makes use of *MS Kinect* gaming sensor for gesture steering, which has its equivalents in keyboard steering as well. *Molecular Rift* is able to read three structure file formats: PDB, MOL2 and SDF and to perform calculations with several forcefields thanks to usage of *Open Babel* chemical tool kit, make representations in different styles and colours and label specific atoms or amino acids. Additionally, the authors made user tests, which encourage to further improvements, as, despite some minor problems, the tool was rated as useful and interesting by the group of scientists [29].

2.3.3 Problems and future

Despite all the advantages of VR application to bioinformatics, it also generates difficulties, which need to be overcome to make it a standard technology in molecular visualisation. Performance limitations constitute one of them. Complexity of the vast majority of biomolecules generates high computational cost if visualized, increasing the demand for CPU and GPU processing power of the computer. Although technological development causes new solutions that provide better efficiency of computers, the overall performance may be improved on the software site as well. In example, developers of *ADN-Viewer*, program dedicated for nucleic acids imaging, implemented the idea of decrease of image quality with the distance from the user. It means, that the closest objects were represented in great details, whereas further

ones- with lower accuracy. Moreover, they applied algorithm that deletes currently hidden parts of the structure from the data, what saves computational expenses [25].

Next problem that occurs while developing for VE is the issue of navigating tools, which are necessary for the proper exploration of biological structures. In three-dimensional space it is problematic to place 2D control panel. What is more, as currently most common gear for VR is a Head-Mounted Display, which is not transparent, limitations of external input devices (mouse, keyboard, joysticks) usage occur. Nevertheless this difficulty decreases, thanks to new possibilities that appear on the market, like VR gloves, eye-tracking devices, cameras etc. However, need of such specialised equipment generates additional costs. Anyway, the rule which need to be followed is that the program should be made in a way that minimizes the possibility of user disorientation [25, 29].

It is important to remember, that creating visualisation programs for VE requires co-operation of programmers and biologists, to make it as effective and useful as possible. Although there is still no software available commercially for described purposes, many papers reports new solutions in this topic, what indicates the need of visualisation improvement. Nowadays, biotechnology development and the speed of this development is strongly dependent from bioinformatics, thus one can conclude, that the potential market is relatively large. Additionally, next step in this issue for the future development is the visualisation of whole cells or even biological pathways and organelles, as Norby et al. suggest the idea of such representations, with the function of simulating and steering the biological regulation. Furthermore, it should be stressed out that scientific usefulness of VR is followed by educational one. Understanding the contents of big data largely depends on the way it is presented. Surely "gamification" and interactiveness of biochemistry could make it more attractive and assimilable for students, what may happen to be a good investment in science in the future [29].

Chapter 3

Molecule VR Viewer

This chapter describes all functionalities of the core and, simultaneously, result of presented work, which is the Molecule VR Viewer program. Description of methodology and implementation details are discussed in Chapter 4.

Generally, Molecule VR Viewer is an application that presents structure of the molecule in Virtual Reality, basing on the data contained in PDB file. As the program provides several modes and options, functionalities are described step by step, starting from the main menu. For its full performance, application requires *Oculus Rift* HMD with attached *Leap Motion* device.

3.1 Main Menu

As main menu window is created in 2D and requires keyboard and mouse input, it is not rendered for *Oculus*. Starting panel is presented in Figure 3.1.

In the first input field, user is asked to enter the PDB ID- the four-character unique code that stands for a single molecule structure, which may be found in Protein Data Bank. Next, in settings panel, user may influence three properties of the generated visualisation. Toggle box "Hide Hydrogens" enables to choose whether to take hydrogen atoms into account while generating the structure or not. This function is purely practical, as the big number of hydrogen atoms (which often constitute around 50% of the protein) may affect clarity of the view. However, if the PDB file does not provide data about hydrogen atoms (due to the method of obtaining the structure), toggle does not have any influence on the created view. Then, user may choose drawing and colouring style from the two drop down lists, but the choice of the representation style affects on the availability of colouring methods- for each drawing method different colouring settings may be applied. Application provides five different drawing settings: Van der Waals, Lines, Balls and Sticks, Ribbon and Backbone and four colouring styles: CPK, Residues, Subunits and Secondary Structures. All of them are described in the following subsections.

On the right side of the menu window, short user's guide with pictograms presents the ways of rotating ad zooming the view, switching between modes and how to use the pointing mode.

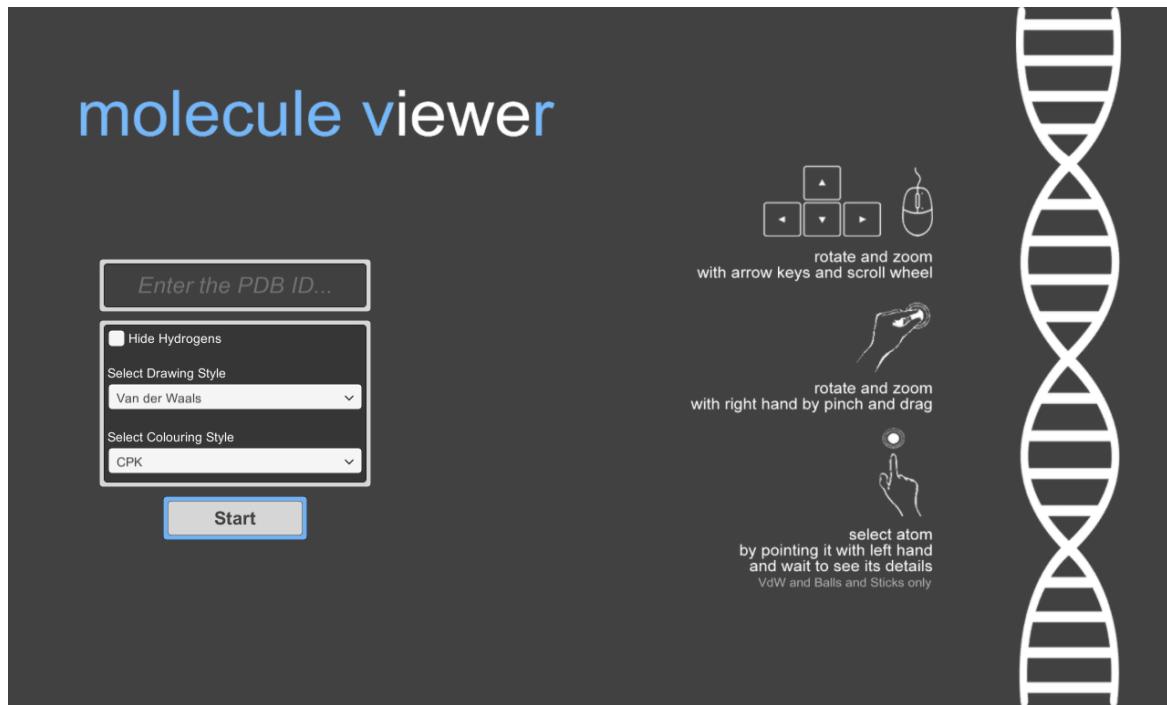


Fig. 3.1 Starting window of the Molecule VR Viewer program.

After clicking the "Start" button, correctness of the entered PDB ID is checked. In case of problems, error message occurs, allowing the user to recognize the problem. If the PDB code was found in the database, program presents "Wait..." message and analyses the selected structure to draw it. From this moment, user may put on the head-mounted display.

3.2 Drawing Styles

3.2.1 Van der Waals

Van der Waals representation creates for each atom a sphere of Van der Waals radius, which is different for every single element. It illustrates the imaginary volume that is occupied by the atom, in which it may interact with other atoms. When the atoms are bounded covalently, their Van der Waals spheres overlap. Thus, the representation shows the approximate shape of the whole molecule and its potentially accessible places, like active sites, which are highly important in case of ligands binding, their design etc. Example of such shape obtained with Molecule VR Viewer is shown in Figure 3.2.

In addition to CPK colouring style presented in the example, Van der Waals drawing style enables Residues and Subunits colouring.

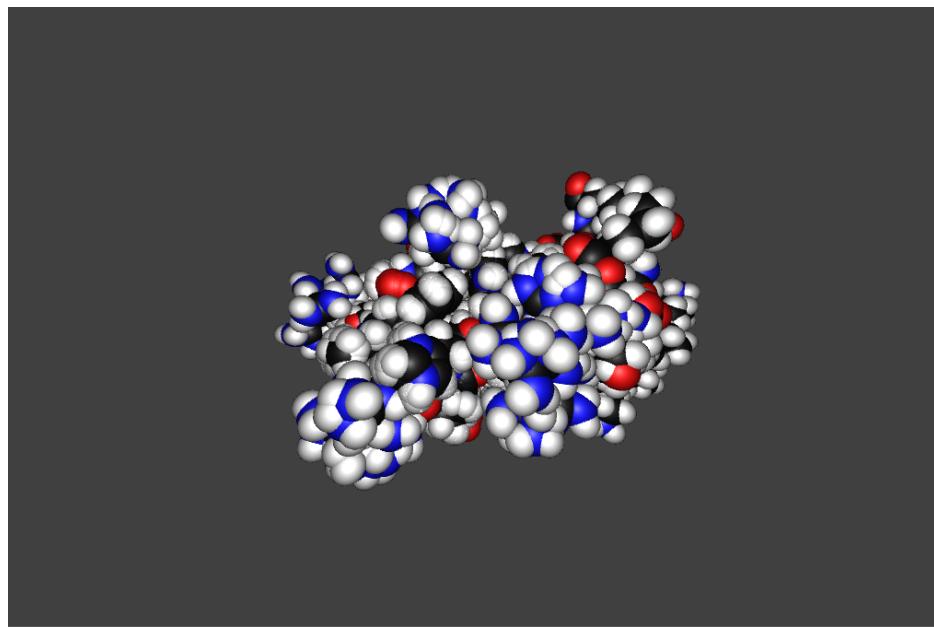


Fig. 3.2 Van der Waals representation of yeast transcription factor ADR1 (PDB ID: 1ARD) with CPK colouring and showed hydrogens.

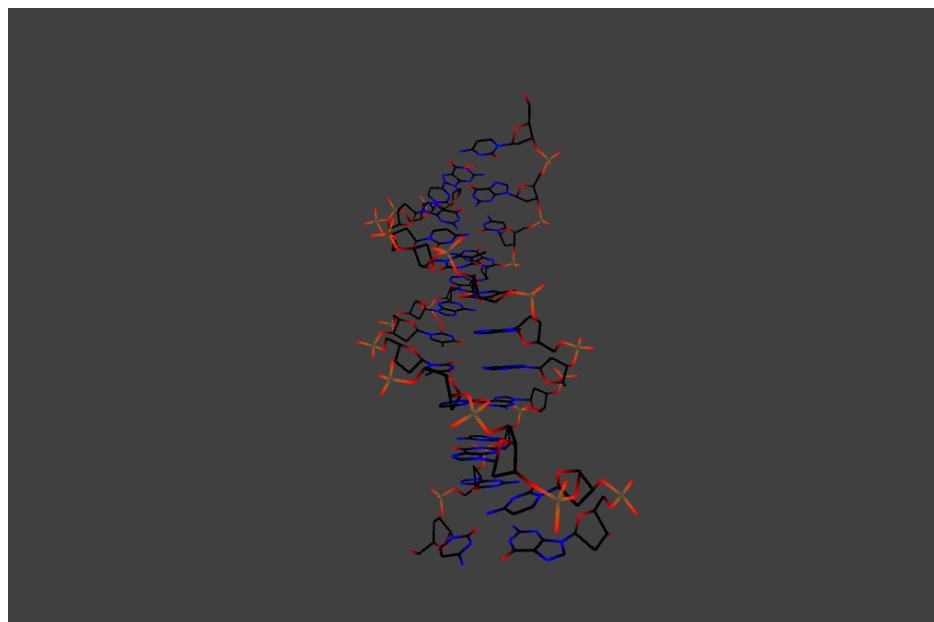


Fig. 3.3 Structure of the B-DNA dodecamer (PDB ID: 1BNA) obtained with Lines style and CPK colouring. The double helix structure of this nucleic acid is clearly visible, as well as belonging nucleobases.

3.2.2 Lines

Another visualisation style is showing the covalent bonds existing in the structure and because of that is called Lines representation. Resulted wireframe is clear and, for the experienced scientist, easy to understand and to distinguish individual residues. Comparing to Van der Waals representation, Lines show the skeleton of the molecule, resulting in not crowded view, where one can easily find, for example, chemical rings. The exemplary structure made with bond diagram is presented in Figure 3.3.

As shown, CPK colouring in this representation is made by smooth transition of colour between bonded atoms. The colouring options in this case are same as in Van der Waals setting.

3.2.3 Balls and Sticks

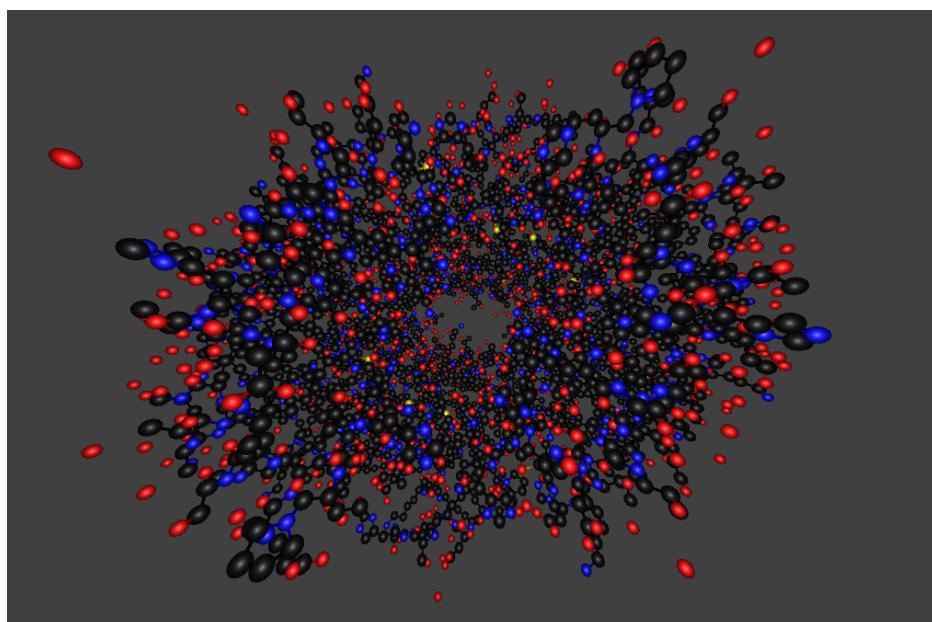


Fig. 3.4 Deoxy human hemoglobin (PDB ID: 1A3N) - important transporter of oxygen, shown with Balls and Sticks setting and CPK Colouring.

Balls and Sticks is simply fusion of Lines and Van der Waals representation. To make the clear view, radii of the atoms are twice reduced. This drawing style is widely used in the molecule visualisation for better understanding, especially for education purposes. Nevertheless it is important to remember, that real atoms are not spheres bonded with sticks, and such idea is only a convention, which does not provide proper information about molecular shape. Illustration of Balls and Sticks representation demonstrates Figure 3.4. As the presented example is constituted by relatively big protein (tetramer- consists of four chains), the general view seems to be unclear. This case proves, that magnifying the view and looking around freely is crucial in the context of useful visualisation.

3.2.4 Backbone

As protein is, generally saying, chain of amino acids, which are connected with peptide bonds, it does have a main backbone trace. It is one of the most simplified type of representation and is obtained by omitting amino acid side chains. In case of protein, the backbone is created between alpha carbon atoms (carbon atom before the carbonyl carbon in peptide, which bonds with amino acid residues). What is more, nucleic acids are biopolymers as well, and their backbone is created by phosphate and deoxyribose molecules, and in simplified form- by linked phosphorus atoms.

This representation style puts emphasis on the secondary structure, ignoring the primary one somehow. In this approach is not the single residues that matters, but the structures that they form. Additionally, this drawing setting is great for showing the tertiary structure- protein geometric shape or even quaternary one- with all the subunits of molecule, as presented in Figure 3.5. Because of all those properties, the only colouring method applicable here is the Subunits style.

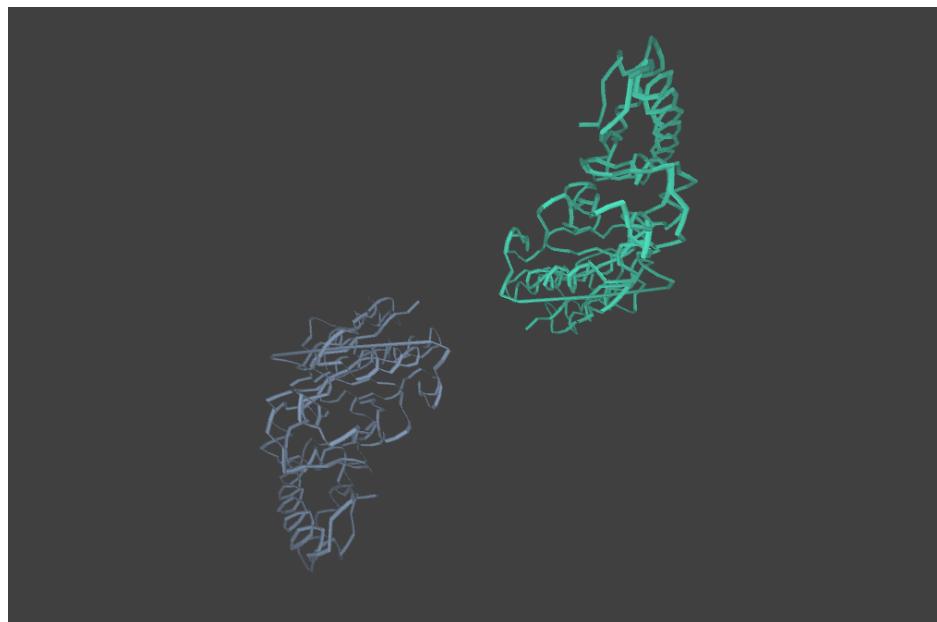


Fig. 3.5 Papain dimer (PDB ID: 4QRV) presented in Backbone style with Subunits colouring. One can recognize alpha helix structures occurring in each chain.

3.2.5 Ribbon

The last drawing style supported by Molecule VR Viewer is Ribbon diagram. It is one of the most commonly used in both scientific and educational world, since it illustrates the whole organisation of protein structure. One can say, that it is improved Backbone representation, with even bigger emphasis put on secondary structure. It is achieved by presenting α -helices and β -strands, the most common and important secondary structure types, as cylindrical spiral ribbons and arrows, respectively, both following the backbone of the peptide. β -sheet

arrows, except just their distinguishing from other structures, play another important role-they show the direction of protein chain, from amino to the carboxyl end. These are the crucial elements of ribbon diagram and they are supported by Molecule VR Viewer, however, in more complex diagrams, other structures, such as loops, turns or even disulfide bonds may be presented.

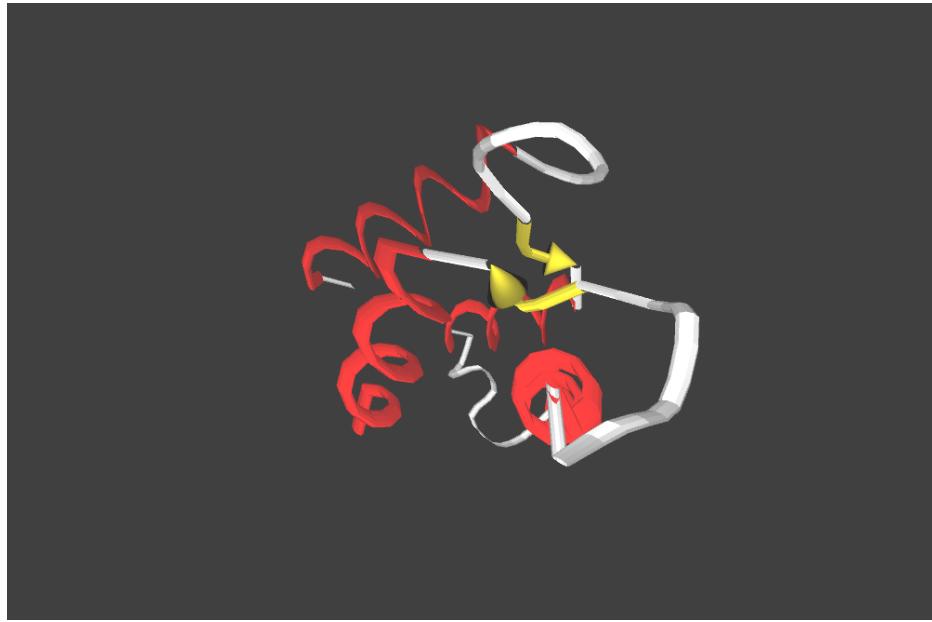


Fig. 3.6 Ribbon diagram of calbindin D9k from *Bos Taurus* (PDB ID: 2BCA) with noticeable four α -helices and two β -strands

Exemplary protein visualised with Ribbon representation is shown in Figure 3.6. In this diagram the most useful colouring style was set- Secondary Structure. However, it is possible to apply Subunits colouring as well.

3.3 Colouring styles

3.3.1 CPK

Colouring style called CPK from it's authors names (Corey, Pauling, Koltun) is a standard convention of colouring atoms on basis of elements they represent [30]. As the original patent of Koltun does not mention all of the elements that may occur in biomolecules, they have been set for this work purposes, and the result is shown in Table 3.1.

The big advantage of this style is its clarity and the easiness of recognizing e.g. metal cofactors in metalloproteins. Such example is presented in the Figure 3.7

H	
C	black
N	blue
O	red
F, Cl	green
P, Fe	orange
S	yellow
Br	dark red
I	purple
Be, Mg, Ca, Sr, Ba, Ra	dark green
Li, Na, K, Rb, Cs, Fr	dark purple
Ti	grey
other	magenta

Table 3.1 CPK colour assignments used in Molecule VR Viewer.

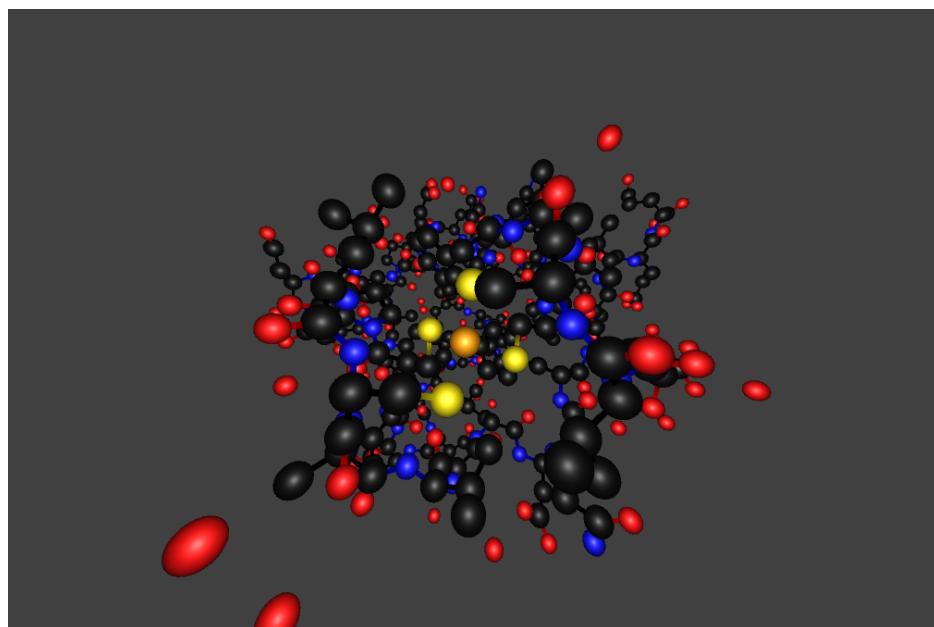


Fig. 3.7 Fe^{3+} ion (orange) in the binding pocket made out of 4 cysteine residues in the structure of rubredoxin (PDB ID: 1B2J) presented in CPK colouring mode and Balls and Sticks representation.

3.3.2 Residues

Both proteins and nucleic acid consist of monomers- amino acids and nucleotides, respectively. Although, as mentioned before, experienced scientist may recognize the residue directly from its structure, it is more convenient to apply the distinguishing colouring style. Since there are 20 basic amino acids that build proteins, their colours were assigned following to the scheme

presented in Table 3.2, which comes from the *RasMol*, molecular visualisation program [31]. All of the other residue types have random color assigned. The example shows Figure 3.8.

ASP,GLU		Red
LYS,ARG		Blue
PHE,TYR		Dark Blue
GLY		Light Gray
ALA		Gray
HIS		Medium Blue
CYS, MET		Yellow
SER,THR		Orange
ASN,GLN		Cyan
LEU,VAL,ILE		Green
TRP		Purple
PRO		Brown

Table 3.2 Amino acid colour assignments used in Molecule VR Viewer [31].

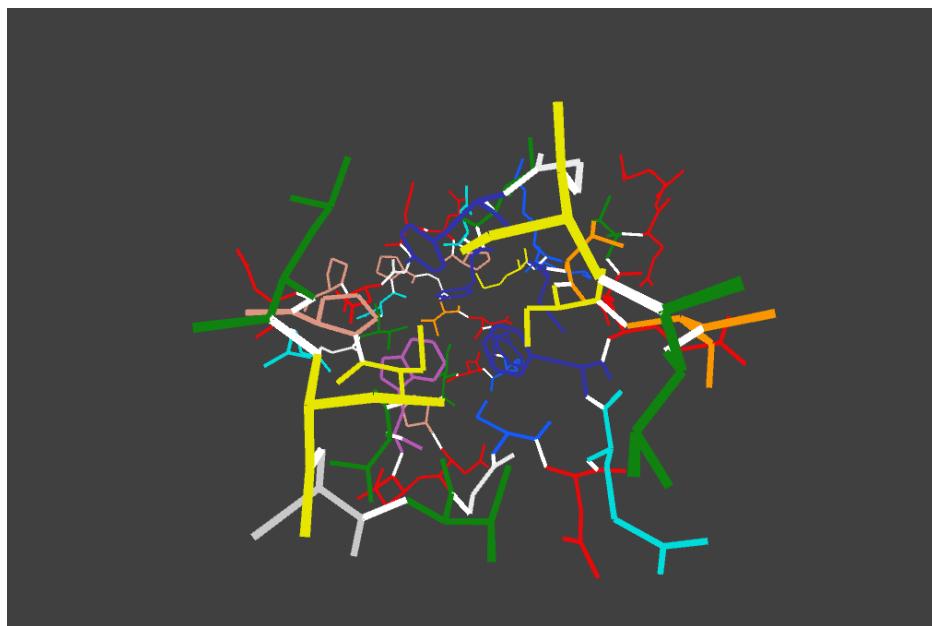


Fig. 3.8 The same rubredoxin (PDB ID: 1B2J) as in Figure 3.7, presented with Residues colouring and Lines representation. In this case one can recognize 4 cysteine residues directly (yellow), not from the sulfur atoms.

3.3.3 Subunits

Since protein or nucleic acid may be an assembly made of multiple chains, forming quaternary structure (in case of peptides), it is sometimes of high importance to recognize which atom

belongs to which subunit, whether subunits are identical (homomers) or not (heteromers). Usually chains are named with consecutive letters of the alphabet, nevertheless, it is not a standard. Therefore, in Molecular VR Viewer, the colour of each subunit is assigned randomly, resulting in the image like in Figure 3.9.

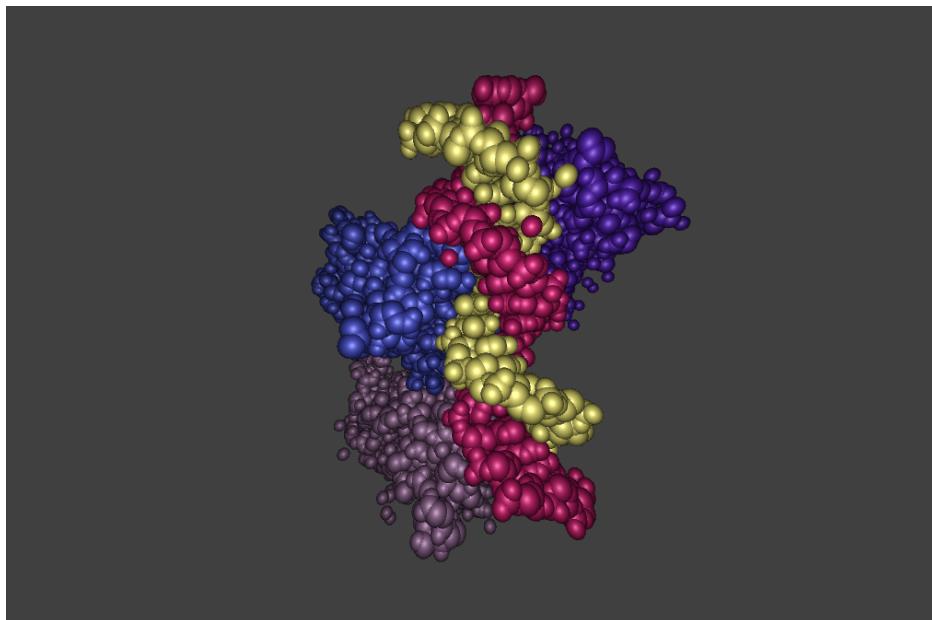


Fig. 3.9 Tumor suppressor p53, which is of high interest in scientific world, complexed with DNA (PDB ID: 1TUP) presented with Subunits colouring and Van der Waals representation. It is not hard to distinguish two DNA chains (pink and yellow) forming double helix, from the protein units.

3.3.4 Secondary Structures

The last colouring setting, dedicated for Ribbon diagram, is the simplest one. As shown in Figure 3.6, each type of secondary structure has different colour assigned. α -helices are red, β -sheets are yellow and the rest of the backbone is white. This style intensifies the effect of distinguishing secondary structures introduced in Ribbon drawing style.

3.4 Visualisation Scene

After clicking the "Start" button, required structure is loaded and user may start to watch and explore it. At the beginning, the molecule spins around its center, what may be stopped by pressing any key or what stops automatically after 15 seconds. Moreover, the name of presented molecule is displayed on the upper side of the user's view, which disappears after 15 seconds as well. With the *Oculus* HMD on, user may look around this simulated nano-world.

3.4.1 Rotating and Zoom

Rotating and magnifying of the molecule is one of the most important feature for the properly designed visualization tool. In Molecule VR Viewer it may be accomplished in two ways. First one is simply pressing arrow keys on computer's keyboard for rotation up-down and left-right or scrolling the mouse wheel for zooming in and out. Nevertheless, with a head-mounted display on, user may find pressing the proper key hard, as the *Oculus* headset does not provide transparency and the whole vision is covered with VR. Therefore, the second way is introduced.

Rotating and zooming of the image may be obtained by the Leap Motion Controller, the device already described in subsection 1.4.3. If it is attached to the HMD, it requires only to extend right arm forward. Then, on the viewer scene the hand model should appear, which is the exact equivalent of the user real hand.

To start moving the scene, user needs to make a pinch gesture (with index finger and thumb). Without releasing it, movement of the hand up-down and left-right rotates the molecule. For zooming in, user should pinch and pull it to yourself, for zooming out- pinch and push forward. The hand model making pinch gesture on the Molecular VR Viewer scene illustrates Figure 3.10.

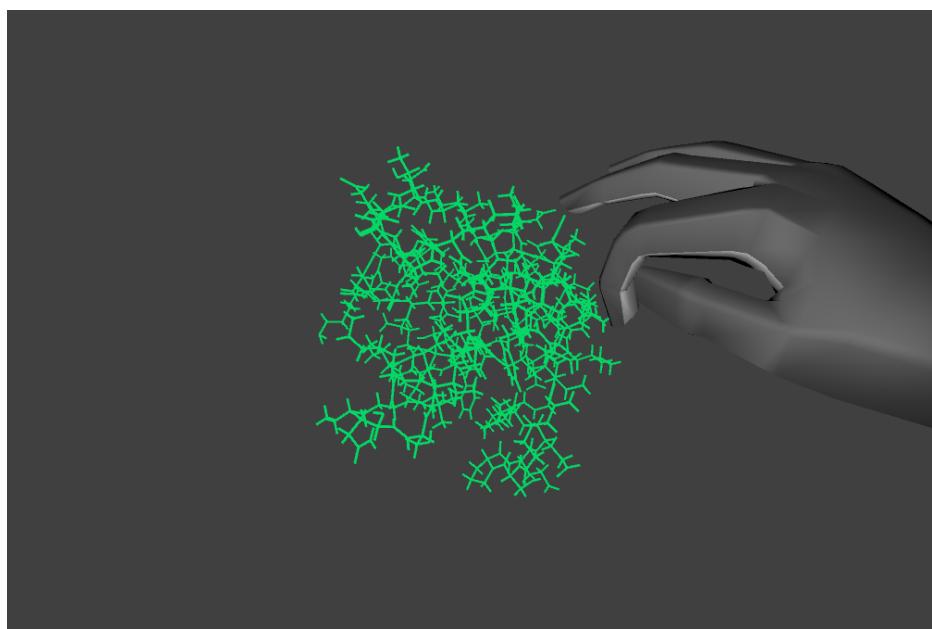


Fig. 3.10 Pinching gesture enables rotation and magnifying of the structure.

Manipulating with *Leap Motion* usage requires a bit of practice. However, this solution is perfect for the Virtual Environment, since it provide another dimension of user's immersion.

3.4.2 Selecting mode

In the 3D VR world, traditional, two-dimensional pointers (like computer mouse) do not find an application, obviously due to the lack of one dimension. However, in desktop programs

for molecular visualisation, user has the ability of selecting single atom and checking its details. As the new solution should not be less functional, Molecule VR Viewer has another feature that solves this problem. The selecting mode with usage of *Leap Motion* constitutes the solution.

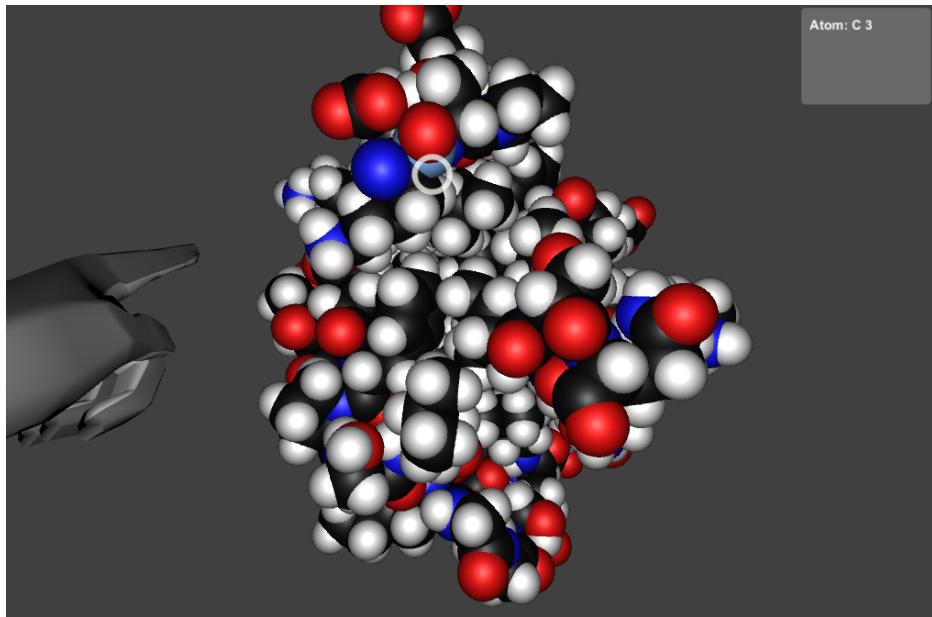


Fig. 3.11 Pointing with the left hand at the atom. As seen, the cross hair is aimed at the carbon atom number 3, what changed the colour of the sphere from black to bright blue.

Since this feature is mostly needed for pointing on an atom, it is available only for Van der Waals and Balls and Sticks representations, as these are the only modes that in fact create objects for each atom. After loading a structure in one of these two styles, user may turn on the Selecting mode by extending left arm forward. In other cases, the left hand does not appear on the screen and the mode is inactive.

Selecting is held by pointing with an index finger at the chosen atom. Pointed sphere changes its color to the bright blue and the atom's name and number is displayed in the right upper corner. That feature enables searching for the requested fragment. If user wants to become familiar with details of the chosen atom, he or she should point at it for few seconds. Targeting is facilitated by a cross hair, which is simultaneously a progress bar indicating the time that left for the atom details presentation (Figure 3.11). The time buffer protects the user from accidental pointing on other atoms, which could cause chaotic performance. After successful selection of an atom, its name, element, number, residue and chain is displayed in the upper right corner for 5 seconds (Figure 3.12). In this time user cannot select another sphere.

Escaping from the selecting mode is possible by hiding left hand out of *Leap Motion* controller's vision.

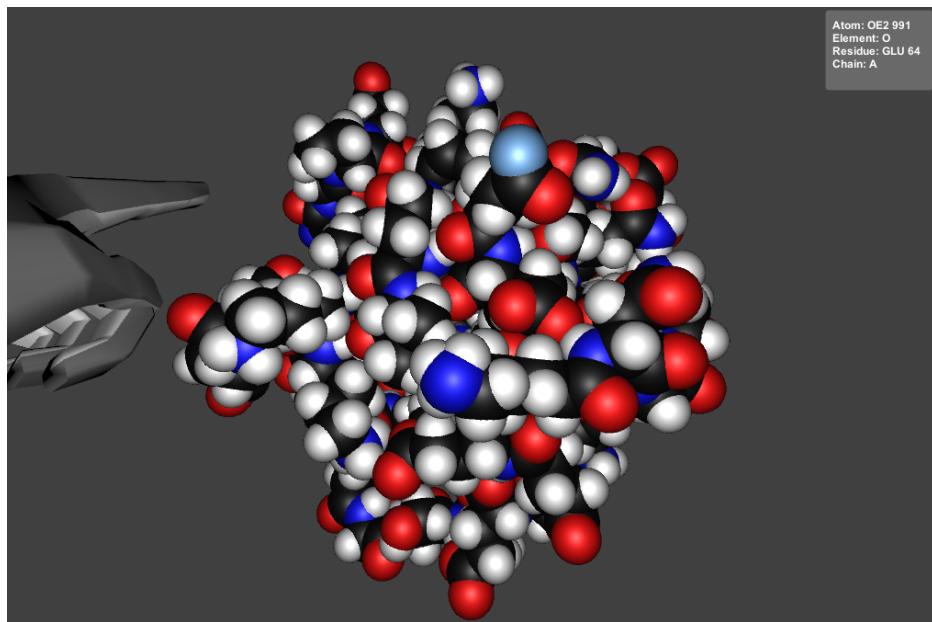


Fig. 3.12 Selected atom and its details shown in the right upper corner. The cross hair is hidden, as time needed for details presentation has passed and pointing at other atoms is temporarily inactive.

3.4.3 Exit

As the re-typing of PDB ID code etc. requires taking off the HMD, there is no way to get back to main menu in single session. Changing of molecule code or chosen setting requires restarting of the program.

Closing the Molecule VR Viewer may be done by simply clicking the escape key on the keyboard.

Chapter 4

Molecule VR Viewer- implementation

As previous chapter outlines the result of this work, it does not provide description of methodology needed for creating Molecule VR Viewer. Therefore, this chapter is devoted to programming details, used algorithms and implementation of the project.

4.1 *Unity 3D*

Unity 3D game engine provides IDE (Integrated Development Environment) for games and other interactive materials, both in 2D and 3D. In version 5.3, it enables writing scripts in two languages: UnityScript and C#. The latter one was used for this work's purpose.

For better understanding of Molecule VR Viewer implementation, the following subsections describe the components needed for creating materials in *Unity 3D* and explain how do they work.

4.1.1 Editor

Generally saying, *Unity 3D* products are created on two tracks: Editor and Scripting. The first one is editable in main window of application, showing the scenes of the project, their environments and objects.

Objects in *Unity 3D*, called GameObjects, are containers that hold Components and Scripts, which may influence their appearance, properties and actions. For example, every GameObject has the Transform Component, which represents position and orientation in space. The engine serves a huge variety of editable Components, which may be attached to the object, like its physical properties, its material etc. Moreover, some of the general GameObject types containing needed Components are provided by *Unity 3D*, e.g. Cameras, which are responsible for observing the scene, Lights, 3D basic models, UI (User's Interface) elements etc. Additionally, the GameObjects can be stored as templates called Prefabs, from which one can create new instances of the GameObject [32].

Summing up, Editor mode is the core of the whole project, which enables the developer to combine all of the needed objects, create Scenes, change player settings (like enabling Unity VR Support) and manage it all to create the final product.

4.1.2 Scripting

The second track of developing programs in *Unity 3D* is Scripting. Scripts may be responsible for arranging events and responds in the Gameplay mode, they can influence Components of GameObjects, create and destroy GameObjects etc. As they play huge role in whole mechanism of game or material creating, it is worth knowing their manner of working.

First of all, *Unity* scripts derive from the built-in class MonoBehaviour, which is the base class connecting the script with internal workings of *Unity* engine. By default, the script defines two methods: Start, which is responsible for all initialisations before the gameplay begins, and the next one, Update, which is the place for code that is going to be called once per frame during the gameplay.

Secondly, it is worth mentioning, that scripts created for *Unity* do not contain constructor functions. This feature is supported by the Editor mode, when script is attached to some GameObject as its Component, which is equal to creating an instance of the class. What is more, variables declared in the class field, which are public, are editable in Editor mode and can be even linked with another GameObject.

Thirdly, scripts are able to access other components that belong to the same GameObject or even to different GameObjects [32].

Event Functions

Start and Update functions mentioned above are in fact just two examples of the big group of event functions, which are activated in certain moments of a gameplay. They are the consequence of the fact, that *Unity* does not run the code continuously, script by script. Instead, scripts take control intermittently, when their certain functions are called by *Unity*. After the function's execution, control is passed back to the engine [32].

The remaining examples of event functions and their order of execution are shown on the flowchart in Figure 4.1.

4.2 *Implementation details*

Molecule VR Viewer program consists of two scenes- menu and viewer scene- and each one contains the core script that manages all of the functionalities. In this section they are described in details.

4.2.1 **MainMenu.cs**

MainMenu script is attached to the Main Camera of Menu scene, where the two-dimensional UI (User's Interface) configured in Editor mode, is presented (Figure 3.1). Each UI element has its reference in the script, what enables pulling the data out of it.

Start function is responsible only for adding the listener to the "Start" button, which is the ButtonFunction method. It is responsible for setting of the feedback message and checking the correctness of entered PDB ID Code, which means its length (PDB codes

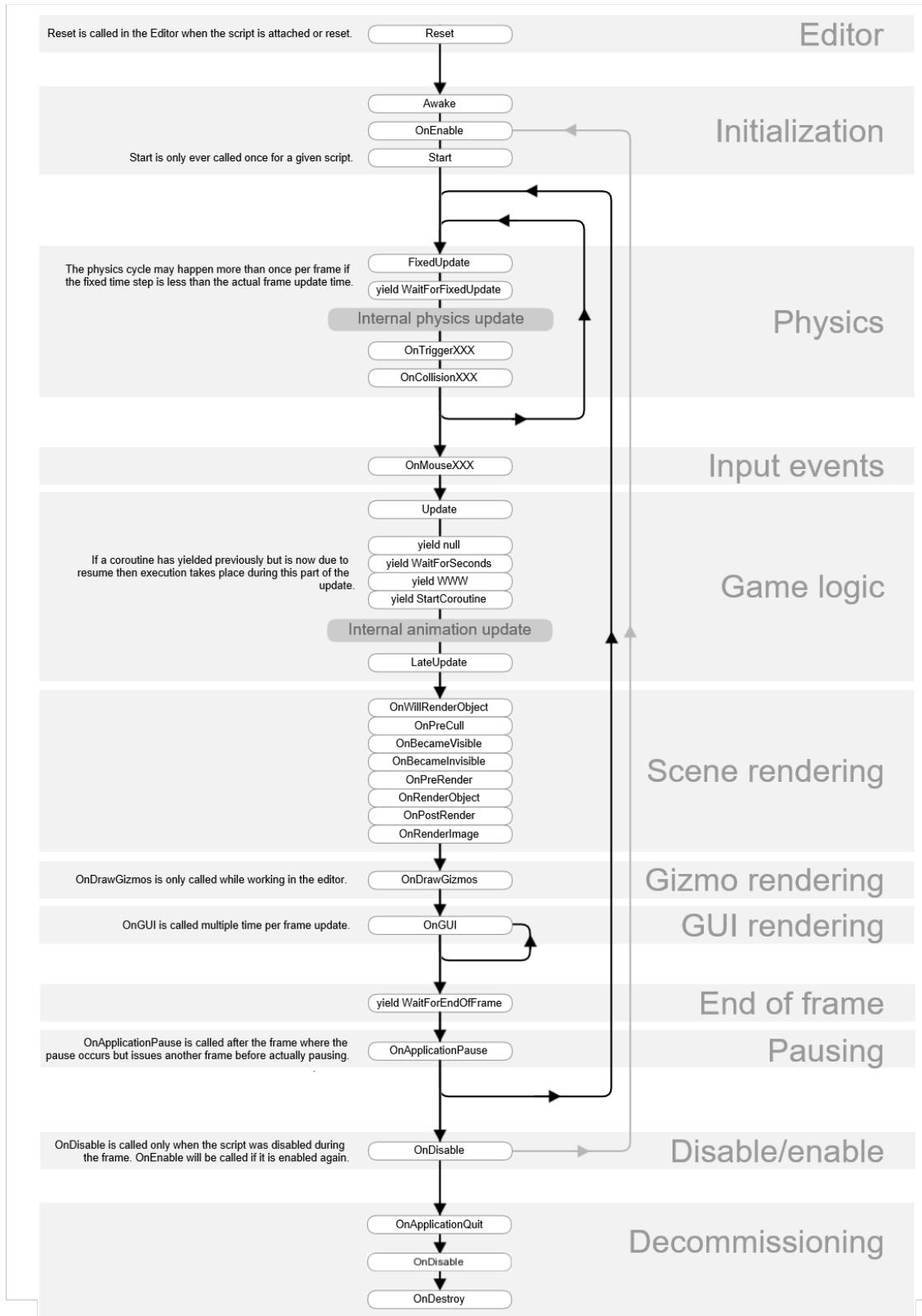


Fig. 4.1 Script lifecycle flowchart with repetitions included [32].

always have 4 characters) and its actual existence in PDB data base. The latter is done with the `HttpWebRequest` and `HttpWebResponse` classes from `System.Net` library. If the URL address responses, the Viewer scene is loaded with *Unity's* `SceneManager`. If not, and the `WebException` occurs, method displays the `WebException` message on the feedback message presented to the user.

The `Update` function contains elements responsible for assigning the data inserted by the user and for changing the set of colouring styles depending on which drawing style was chosen. Variables, which are obtained from UI, are held by the `Configurator` class.

Configurator.cs

This class, which does not inherit from `MonoBehaviour`, initialises four static variables, which are `pdbID`, `hideHydrogens`, `representationStyle` and `colouring`, with corresponding get and set methods.

As the user's settings are needed during the entire gameplay, they need to be passed between scenes, which is more challenging than, e.g. communication between objects. The `GameObject`, where they are set, is no longer needed, therefore using the *Unity's* method `DontDestroyOnLoad`, which prevents the object from being trashed while loading another scene, is not the best solution. This is why the other way was chosen, which is the usage of static attributes, that are available as long as the application is running.

4.2.2 MoleculeViewer.cs

Viewer scene, which is loaded after successful request from the PDB server, is managed by the `MoleculeViewer` script, which constitutes the core of the whole program. It is the only component (except, obviously, `Transform` component) attached to the `GameObject` called `CameraParent`. The reason of that setting is explained later in this section.

In this case, `Start` function contains all the components needed for creation of structure representation in a way chosen by the user in main menu. First of all, the object of the `FileReader` class is initialized and its method `ReadFile` is called.

FileReader.cs

This class does not inherit from `MonoBehaviour` and provides the engine for extracting needed data from PDB database. Therefore, at the very beginning it downloads the PDB file and reads it line by line.

This is a good moment to introduce and describe the PDB file format. Its main property is its linearity- the content is presented in number of lines, with 80 columns per each. First six columns contain a record name, which defines the record type of the line. This is a very useful feature, as every record type is divided into fields, that have standardised positions in the line. Knowing the exact location of interesting data is very convenient in terms of file analysis. Because PDB file may contain a large variety of record types, only those, that were used in this work, are described below [33].

- COMPND— macromolecular contents of an entry presented in a form of specification list. Specifications that were used in Molecule VR Viewer are MOLECULE, where the macromolecule name is defined, and CHAIN, which provide list of subunits identifiers, separated by commas.
- SEQRES— listing of the linearly linked chemical components that form a polymer. For proteins these are amino acids, presented in a form of three-letter codes, and for nucleic acids- their residues.
- ATOM- atomic coordinates of an atom that belongs to a standard residue, its name, element symbol, serial number, occupancy, temperature factor, charge and belonging to subunit and residue. XYZ coordinates are in Angstroms. What is important, these records are listed in an ordered way, for proteins from amino to carboxyl terminus, and for nucleic acids- from the 5' to the 3' terminus.
- HETATM- atomic coordinates of an atom that does not belong to the polymer, its name, element symbol, serial number occupancy, temperature factor, charge and belonging to subunit and residue. Useful especially for prosthetic groups, inhibitors, solvent molecules, metal ions etc.
- CONECT- specifies connectivity between atoms, mostly for non-standard atoms and disulfide bridges, by assigning to the atom's serial number all the atoms that are bonded with it.
- HELIX- specifies the position of helices in a molecule by indication of the initial and ending residue. Helix type is also described by the number corresponding with helix class.
- SHEET- specifies the position of sheets in a molecule similarly as in HELIX.

References

- [1] Heim M.: *Virtual realism*. Oxford University Press, New York, 1998.
- [2] Fuchs H., Bishop G. et al.: *Research Directions in Virtual Environments*. NFS Invitational Workshop, Univ. North Carolina, 1992.
- [3] Cruz-Neira C.: *Virtual Reality Overview*, SIGGRAPH'93 Course, 23, 1.1-1.18, 1993.
- [4] Sherman W., Craig W.: *Understanding Virtual Reality. Interface, Application and Design*. Morgan Kaufmann Publishers, San Francisco, 2003.
- [5] Milgram P., Takemura H., Ustumi A., Kishino F: *Augmented Reality: A class of displays on the reality-virtuality continuum*. SPIE proc. Telemanipulator and Telepresence Technologies, 2351: 282-292, 1994.
- [6] Olszewski P.: *Historia wirtualnej rzeczywistości*, 2015. Retrieved January 20, 2016, from [<http://www.komputerswiat.pl/centrum-wiedzy-konsumenta/gaming/wszystko-o-wirtualnej-rzeczywistosci/historia-wirtualnej-rzeczywistosci.aspx>].
- [7] Mandal S.: *Brief Introduction of Virtual Reality & its Challenges*. International Journal of Scientific & Engineering Research, 4, 4: 304-309, 2013.
- [8] Lange B.S. et al.: *The Potential of Virtual Reality and Gaming to Assist Successful Aging with Disability*. Physical medicine and rehabilitation clinics of North America, 21.2: 339-356, 2010.
- [9] Brooks Jr. F.P.: *What's Real About Virtual Reality?* IEEE Computer Graphics and Applications, 19, 6: 16-27, 1999.
- [10] Mazuryk T., Gervautz M.: *Virtual Reality History, Applications, Technology and Future*. Institute of Computer Graphics, Vienna University of Technology, 1996. Retrieved January 20, 2016, from [<https://www.cg.tuwien.ac.at/research/publications/1996/mazuryk-1996-VRH/TR-186-2-96-06Paper.pdf>].
- [11] Wikibooks contributors: *Cg Programming/Unity/Projection for Virtual Reality*, Wikibooks, The Free Textbook Project, 2015. Retrieved January 25, 2016, from [https://en.wikibooks.org/w/index.php?title=Cg_Programming/Unity/Projection_for_Virtual_Reality&oldid=2988608].

- [12] Oscillada J.M.: *Comparison Chart of FOV (Field of View) of VR Headsets*, 2015. Retrieved January 20, 2016, from [<http://www.virtualrealitytimes.com/2015/05/24/chart-fov-field-of-view-vr-headsets/>].
- [13] Lacrama, Dan L.; Fera D.: *Virtual Reality*. Ann. Univ. Tibiscus, Computer Science Series V, 1: 137-144, 2007.
- [14] Parisi T.: *Learning Virtual Reality*. O'Reilly Media, Sebastopol, 2015.
- [15] Gobbetti E., Scateni R.: *Virtual reality: Past, present, and future*. Virtual Environments in Clinical Psychology and Neuroscience: Methods and Techniques in Advanced Patient-Therapist Interaction, 3-20, IOS, Amsterdam, 2015.
- [16] *Oculus Rift Documentation*, 2016. Retrieved January 27, 2016, from [<https://developer.oculus.com/documentation/>].
- [17] *FeelReal official website*, 2016. Retrieved January 27, 2016, from [<http://feelreal.com/smells>].
- [18] Marco J.: *List of Haptic Controllers under Development for Virtual Reality*, 2015. Retrieved January 27, 2016, from [<http://www.virtualrealitytimes.com/2015/03/13/list-of-haptic-controllers-virtual-reality/>].
- [19] *Overview of Positional Tracking Technologies for Virtual Reality*, 2014. Retrieved February 06, 2016, from [<http://www.roadtovr.com/overview-of-positional-tracking-technologies-virtual-reality/>].
- [20] Weichert F., Bachamn D., Rudak B., Fisseler D.: *Analysis of the Accuracy and Robustness of the Leap Motion Controller*. Sensors (Basel, Switzerland), 13(5): 6380-6393, 2013.
- [21] *Leap Motion Documentation*, 2016. Retrieved May 06, 2016, from [<https://developer.leapmotion.com/documentation/>].
- [22] Singal N.: *Magic Realism*, 2014. Business Today, November 8, 205-208, 2015.
- [23] Goodsell D.S.: *Bionanotechnology : lessons from nature*. Wiley-Liss, Inc., Hoboken, New Jersey, 2004.
- [24] Gruca A.: *Bioinformatyczne bazy danych*. Wydawnictwo PJWSTK, Warszawa 2010.
- [25] Anderson A., Weng Z.: *VRDD: Applying virtual reality visualization to protein docking and design*. Journal of Molecular Graphics and Modelling, 17: 180-186, 1999.
- [26] Schulze J.P. et al.: *Advanced applications of virtual reality*. Advances in Computers, 82: 217-260. Academic Press, Burlington 2011.
- [27] Hérisson J. et al.: *3D visualization and virtual exploration of genomic sequences*. Data Science Journal, 4: 82-91, 2005.

- [28] Hamdi M. et al.: *Prototyping bio-nanorobots using molecular dynamics simulation and virtual reality*. Microelectronics Journal, 39: 190–201, 2008.
- [29] Norrby M. et al.: *Molecular Rift: Virtual Reality for Drug Designers*. Journal of Chemical Information and Modeling, 55: 2475-2484, 2015.
- [30] Koltun W.L.: *Space filling atomic units and connectors for molecular models*. U. S. Patent 3170246, 1965.
- [31] *Rasmol v2.6 Manual*, 2016. Retrieved May 06, 2016, from [<http://life.nthu.edu.tw/~fmhsu/rasframe/TOC.HTM>].
- [32] *Unity 3D Documentation*, 2016. Retrieved May 27, 2016, from [<http://docs.unity3d.com/Manual/index.html>]
- [33] *Protein Data Bank File Format Documentation*, 2016. Retrieved June 7, 2016, from [<http://www.wwpdb.org/documentation/file-format>]