University of Bergen
Department of Informatics

Exploring the potential of citizen science for de novo sequencing of MS/MS spectra

Author: Ingrid Liabakk Eriksen
Supervisors: Harald Barsnes and Yehia Farag

UNIVERSITETET I BERGEN
Det matematisk-naturvitenskapelige fakultet

June, 2022

Abstract

Proteomics is the large-scale study of the function and structure of proteins expressed in a given sample. It generally uses mass spectrometry-based technologies to detect and identify changes in the proteome. One of the common methods is de novo sequencing of MS/MS spectra, which can be used to determine a sequence of amino acids from a mass spectrum. However, while an algorithm using de novo sequencing has a specific approach and would thus give the same solution every time, people on the other hand, would vary more in their approach to solving such a problem and therefore come up with a broader range of solutions. This thesis presents a prototype exploring the feasibility of developing a citizen science game based on de novo sequencing of MS/MS spectra. Furthermore, the main benefits and limitations of such a game are outlined.

Acknowledgements

First and foremost, I would like to thank my supervisors, Harald Barsnes and Yehia Farag, for all their help, support, and advice. I would also like to thank them for regular meetings with close follow up of my progression and interesting discussions.

Secondly, I would also like to thank my fellow students at the Department of Informatics, University of Bergen. It has been a truly motivational environment for studying, not to mention procrastinating.

Lastly, I would like to thank Tobias for his support and for always brightening my day.

`Ingrid Liabakk Eriksen`
Monday 13th June, 2022

Contents

List of Figures

List of Tables

Listings

Thesis Objective

Explore the feasibility of developing a citizen science game based on de novo sequencing of MS/MS spectra, and discuss the main benefits and limitations of such a game.

Chapter 1

Background

This chapter introduces the theoretical background for the thesis. In particular, it introduces basic biology concepts, the technology used to process the relevant data, and how to analyze it. Finally, the concepts of crowdsourcing and gamification are discussed, including examples of how they can be used in citizen science.

1.1 DNA, Genes and Proteins

Molecular biology is the field of biology that studies the biological activity at a molecular level, such as the structure and interactions of cellular molecules that carry out processes essential for a cell's functions and maintenance [7]. Especially important is the study of deoxyribonucleic acid (DNA). DNA has two main functions; it determines how an organism should look and function, and it contains the genetic material inherited from one generation to the next. The DNA consists of the four nucleotides, A (adenine), C (cytosine), T (thymine), and G (guanine), where a set of three nucleotides make up a codon, which in turn correspond to one amino acid.

The central dogma of molecular biology (**Figure 1.1**) explains how the recipes in the DNA are converted into proteins. It can be divided into three main steps. The first step is DNA replication, where a copy of the protein-coding gene is made. In the second step, transcription, a complementary messenger ribonucleic acid (mRNA) strand is made and transported out of the nucleus and into the cell's cytoplasm. In the last step, the mRNA is read by a ribosome and translated into a protein [8][9].

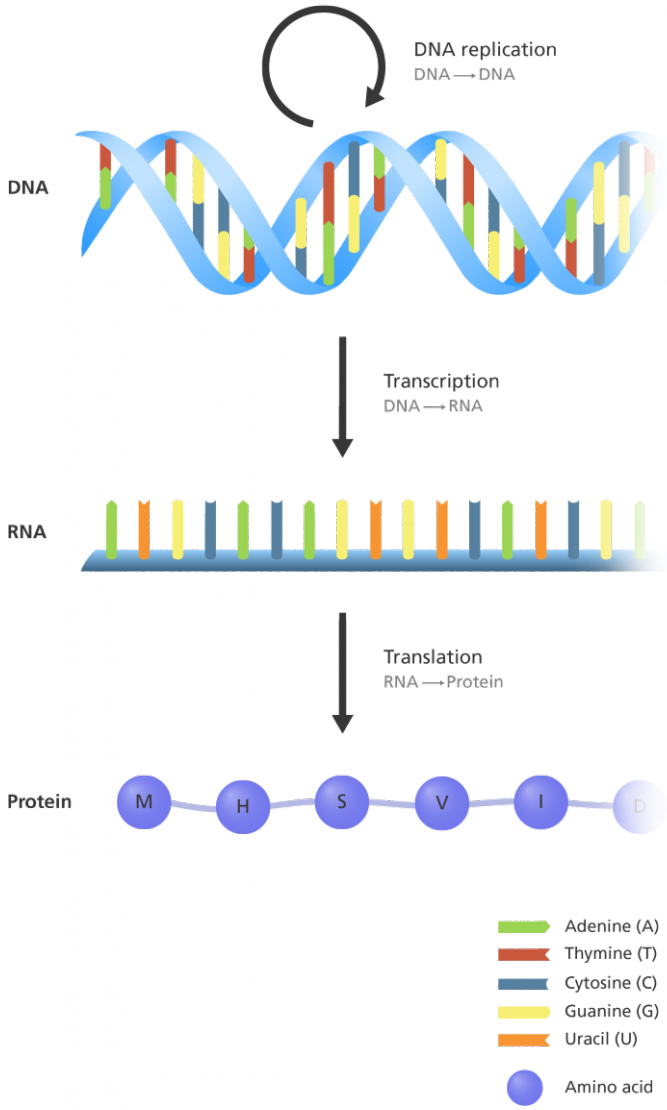

Figure 1.1: The central dogma of molecular biology [1].

Amino acids are organic molecules consisting of an amino compound and an acid compound, together with a side chain (R-group) specific to the amino acid. The side chain can vary in structure and mass, which gives the amino acid different properties. There are 20 standard amino acids [10], shown in **Table 1.1**. Two or more amino acids connected by peptide bonds (**Figure 1.2**) are called a peptide [11], and a chain of peptides make up a protein [12]. After the protein is created, it can go through post-translational modifications (PTMs). PTMs change the properties of a protein by adding or removing a modifying group to one or more amino acids or by breaking the peptide bonds between amino acids [13]. There has been identified around 1000 different types of PTMs [14].

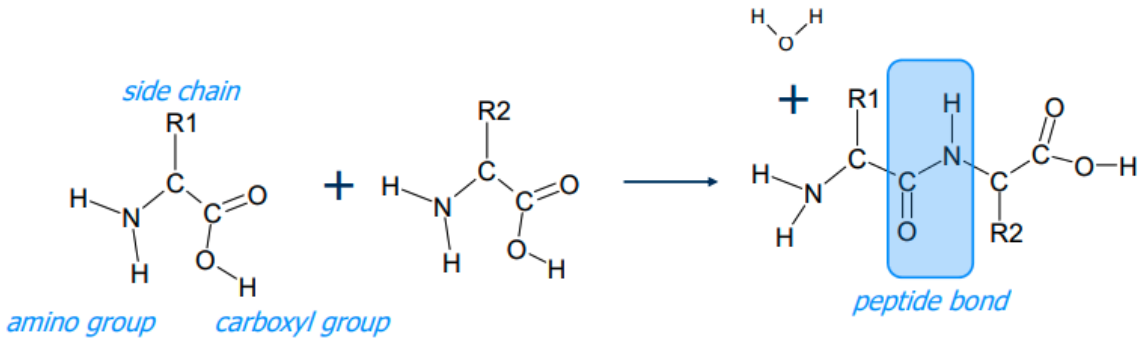

Figure 1.2: The peptide bond between two amino acids [2].

Table 1.1: Overview of the 20 standard amino acids [6]

Residue	3-letter	1-letter	Formula	Average mass
Glycine	Gly	G	C_2H_3NO	57.051
Alanine	Ala	A	C_3H_5NO	71.078
Serine	Ser	S	$C_3H_5NO_2$	87.077
Proline	Pro	P	C_5H_7NO	97.115
Valine	Val	V	C_5H_9NO	99.131
Threonine	Thr	T	$C_4H_7NO_2$	101.104
Cysteine	Cys	C	C_3H_5NOS	103.143
Isoleucine	Ile	I	$C_6H_{11}NO$	113.158
Leucine	Leu	L	$C_6H_{11}NO$	113.158
Asparagine	Asn	N	$C_4H_6N_2O_2$	114.103
Aspartic acid	Asp	D	$C_4H_5NO_3$	115.087
Glutamine	Gln	Q	$C_5H_8N_2O_2$	128.129
Lysine	Lys	K	$C_6H_{12}N_2O$	128.172
Glutamic acid	Glu	E	$C_5H_7NO_3$	129.114
Methionine	Met	M	C_5H_9NOS	131.196
Histidine	His	H	$C_6H_7N_3O$	137.139
Phenylalanine	Phe	F	C_9H_9NO	147.174
Arginine	Arg	R	$C_6H_{12}N_4O$	156.186
Tyrosine	Tyr	Y	$C_9H_9NO_2$	163.173
Tryptophan	Trp	W	$C_{11}H_10N_2O$	186.210

1.2 Omics

Omics is the field in biology that ends with -omics, such as genomics, transcriptomics, proteomics, or metabolomics. It is a high-throughput technology that is used to identify, characterize, and quantify all biological molecules that are involved in the structure, function, and dynamics of a cell, tissue, or organism [15].

Genetics is the study of an organism's genes and their role in inheritance, the way a particular trait and condition is passed down from one generation to another. Genomics

is the study of all the genes of an organism and the interaction between the genes and the organism's interaction with the environment [15][16].

The transcriptome is the complete set of ribonucleic acid (RNA) transcripts: the messenger RNA (mRNA), the non-coding RNA, the ribosomal RNA (rRNA), and the transfer RNA (tRNA) expressed by cellular or tissue [17]. Transcriptomics is the study of the transcriptome and everything related to the RNA, including its structure, function, and location [18].

Proteomics is the (large-scale) study of the function of all expressed proteins, also called the study of the proteome. The proteome is the complete set of proteins expressed and modified by a species or produced/present in a defined compartment within the species at a particular time. Proteomics is used to investigate when and where proteins are expressed, rates of protein production, degradation, and steady-state abundance, how proteins are modified (for example, with PTMs), the movement of proteins, and how they interact with each other [19]. Proteomics often uses mass spectrometry-based technologies to detect, identify, and quantify changes in protein isoforms and modifications, the interaction between them and their higher-order complexes [20][21].

1.3 Mass Spectrometry

Mass spectrometry (MS) has become the technique of choice when analyzing chemical samples in proteomics. It is a method that calculates the *mass to charge* (m/z) of molecules. Mass spectrometry consists of five components: a sample, the ion source, the mass analyzer, the detector, and the digitizer (**Figure 1.3**).

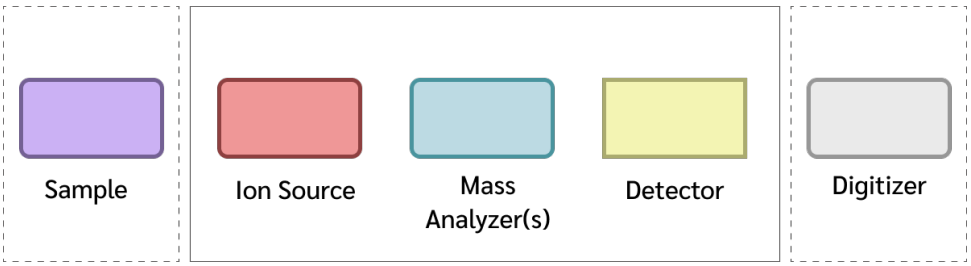

Figure 1.3: The five MS components: sample, ion source, mass analyzer, detector, and digitizer.

First, the sample has to be ionized to create electrical fields made by the ion source. The mass analyzer weighs the mass of the sample with electrical fields. When the ions

hit the detector they generate a current. The detector tracks the current change, where a high current means many ions, and a low current means few ions. This creates a continuous signal. Because it is harder to work with a continuous signal, the digitizer splits the analog detector signals into digital, discrete samples that show the current at a given timestamp [22].

1.4 Tandem Mass Spectrometry

Tandem mass spectrometry (**Figure 1.4**) uses two stages of mass spectrometry in sequence. First, the source creates ions that go through an ion selector, which only allows ions of a chosen mass/charge to pass through. The ions are then fragmented. It is possible to create a, b, or c and x, y, or z-ions (**Figure 1.5**), depending on where the peptide is fragmented. The ions are complementary, meaning that if a peptide is fragmented at the peptide bond, a b-ion is created with a complementary y-ion (**Figure 1.6**). The b-ion and y-ion are the most common ions because the peptide bond is the most vulnerable. After the peptides are fragmented, they are analyzed by a mass analyzer. The combination of the total mass and the mass of each fragment of that molecule help us to uniquely identify a particular analyte (a chemical substance that is the subject of chemical analysis) [22].

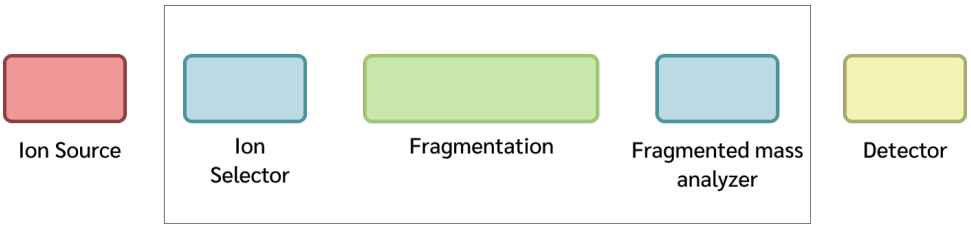

Figure 1.4: The four main components in tandem mass spectrometry: ion source, ion selector, fragmentation, fragment mass analyzer, and detector.

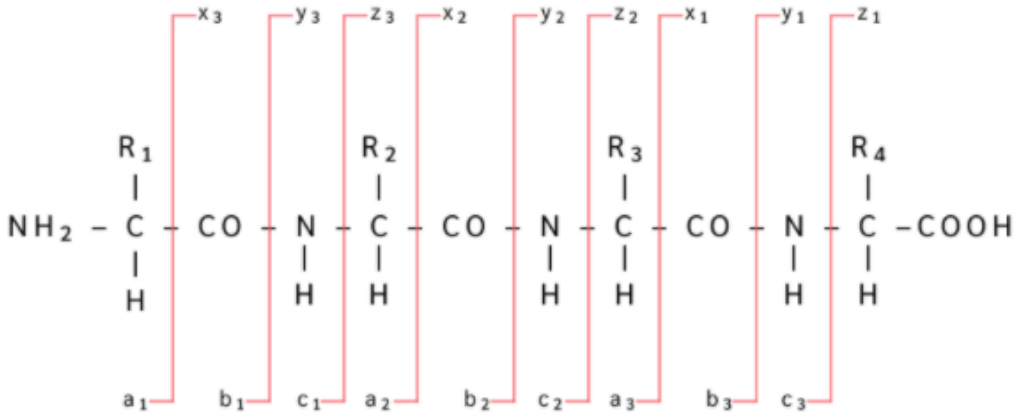

Figure 1.5: A peptide created by four amino acids. The red lines shows the possible peptide bonds which create the a, b, and c-ions with its complementary x, y, and z-ions.

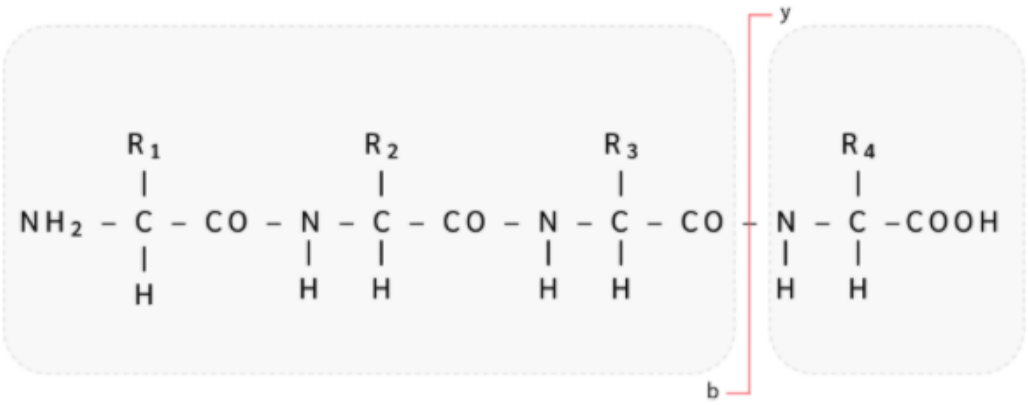

Figure 1.6: A peptide fragmented at the peptide bond, which creates a b-ion and/or a y-ion.

1.5 Peptide Identification

There are three main approaches for identifying peptide sequences from mass spectrometry:

- Searching against a database of known peptide or protein sequences.
- Searching against a spectral library of identified spectra.
- Using de novo sequencing.

The first approach relies on so-called (proteomics) search engines. Search engines are algorithms that identify peptides from tandem mass spectrometry using a reference protein sequence database [23]. The second approach is spectral library searching which uses

libraries of identified spectra to identify observed spectra [24]. Both of these approaches search for matches in a predefined database. Parts of protein sequences from many organisms are, however, still unknown, and even for those that are known, modifications such as post-translational events may prevent the identification of all or part of the sequence [25]. This means that a given peptide sequence may not be present in the database.

De novo peptide sequencing determines a sequence of amino acids from a mass spectrum without relying on prior knowledge of the possible amino acid sequences. The main principle is to use the mass difference between pairs of fragment ions to arrive at the mass of an amino acid residue on the peptide backbone.

Figure 1.7*A* shows a mass spectrum with a highlighted peak that has the mass to charge ratio of 300 Dalton. **Figure 1.7***B* shows the MS/MS spectrum of the highlighted peak in **Figure 1.7***A*. Identifying the MS/MS of a peak means that it is fragmented into several ions, in this case, y and b ions which can be used to identify the MS/MS spectrum and in turn derive the peptide sequence. The MS/MS spectrum is identified by finding the distance between the b-ions and the y-ions. **Figure 1.7***B* shows that the amino acids *glutamic acid* (E) is identified by finding the residue mass between the y_3 and y_2 ion which is 129,114 Dalton. Similarly, the distance between b_2 and b_3 can be determined as *aspartic acid* (D). This process continues until as many of the residues as possible are determined.

De novo sequencing has improved considerably with the latest instrument improvements. Muth and Renard tested and evaluated three de novo sequencing algorithms on accuracy and speed [26]. The best algorithm, Novor [27], had an accuracy of a minimum of 65.2% and a maximum of 83.6%, and a running time performance of 95 spectra per second on average, on a laptop computer. The study showed that the accuracy of the evaluated algorithms dropped significantly for a peptide length of 15 amino acids or more. An algorithm would probably give the same solution every time, since it follows a given approach. People, on the other hand, would vary more in their approach of solving a problem, and therefore find different solutions than an algorithm, and may also be better at finding solutions for a peptide with a length that the algorithms struggle with. This is something crowd sourcing can contribute with.

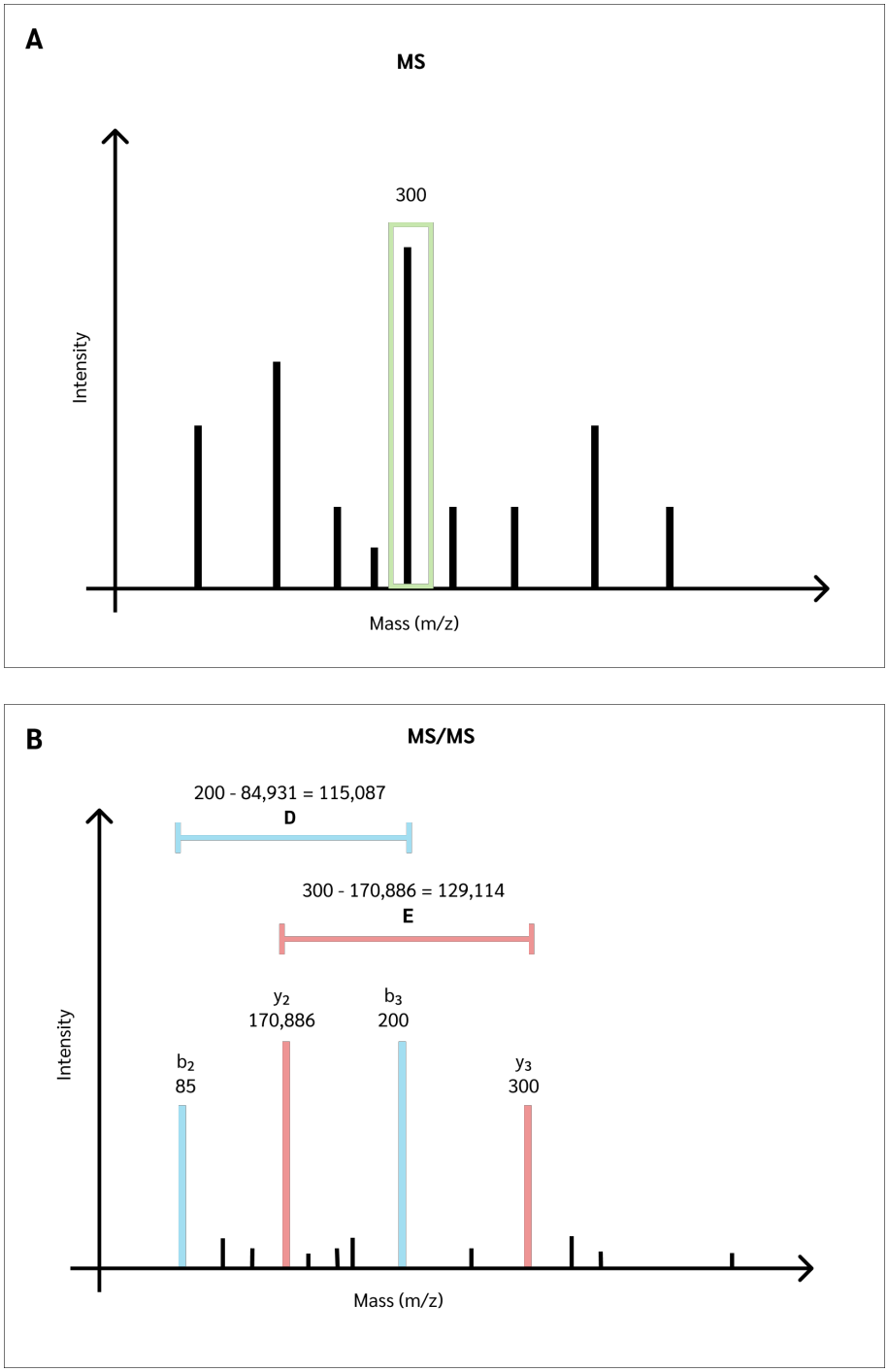

Figure 1.7: The spectra show the mass to charge ratio of the ions on the x-axis and the intensity on the y-axis. Figure A shows a mass spectrum, and figure B shows the tandem mass spectrum of the peak in the first figure that has an m/z of 300 Dalton. The numbers above the red and blue peaks are the mass to charge ratio for each peak.

1.6 Crowdsourcing

Crowdsourcing is a method for outsourcing tasks to a crowd of people. It is used in both academia and industry. Crowdsourcing can be cost-saving and provide the possibility to get help solving problems by obtaining input from a large crowd with a different background than the ones handing out the problem. It can also help process tasks that computers may find difficult to solve.

Crowdsourcing can be divided into four categories: (i) crowd creating, (ii) crowd rating, (iii) crowd processing, and (iv) crowd solving. First, we have crowd creating, which is any site that contains user-generated content, such as YouTube or Wikipedia.

The second category is crowd rating which uses the "wisdom of crowds" to perform assessments or predictions. For example, NASA has used crowd rating in their Clickworkers projects, where volunteers counted craters on celestial bodies, where statistical corrections aggregate the input into a formal scientific unity for researchers [28].

The third category is crowd processing, where a crowd performs many similar tasks, and the many identical answers help validate the quality of the work. This is used by, for example, Galaxy Zoo, one of the best-known online citizen science projects [29]. It started in 2007 with a data set of a million galaxies imaged by the Sloan Digital Sky Survey. The user's task is to classify the galaxies according to their shapes.

The final category is crowd solving. This is when a large crowd finds many different solutions to the same problem. The problems are often complex, and there is no existing solution to the given problem yet. For example, scientists are trying to map the brain, but a major bottleneck is the process of reconstructing the neurons, where it takes months to generate a data set and years to analyze it. Eyewire [30] has built a 3D puzzle game where humans and artificial intelligence (AI) map the neurons in the brain. In their first project, citizen scientists helped discover six new neurons in the retina, potentially helping to discover new cures for vision-related disorders [30].

An active crowd is crucial for crowdsourcing to work. The quality vs. quantity of the different crowdsourcing approaches differs. In crowd processing, the number of submissions is more important than the quality, and the users are generally rewarded for their participation. Whereas in crowd rating, the quality is more important, and the participants are rewarded accordingly. In crowd solving, both quality and quantity are essential [31].

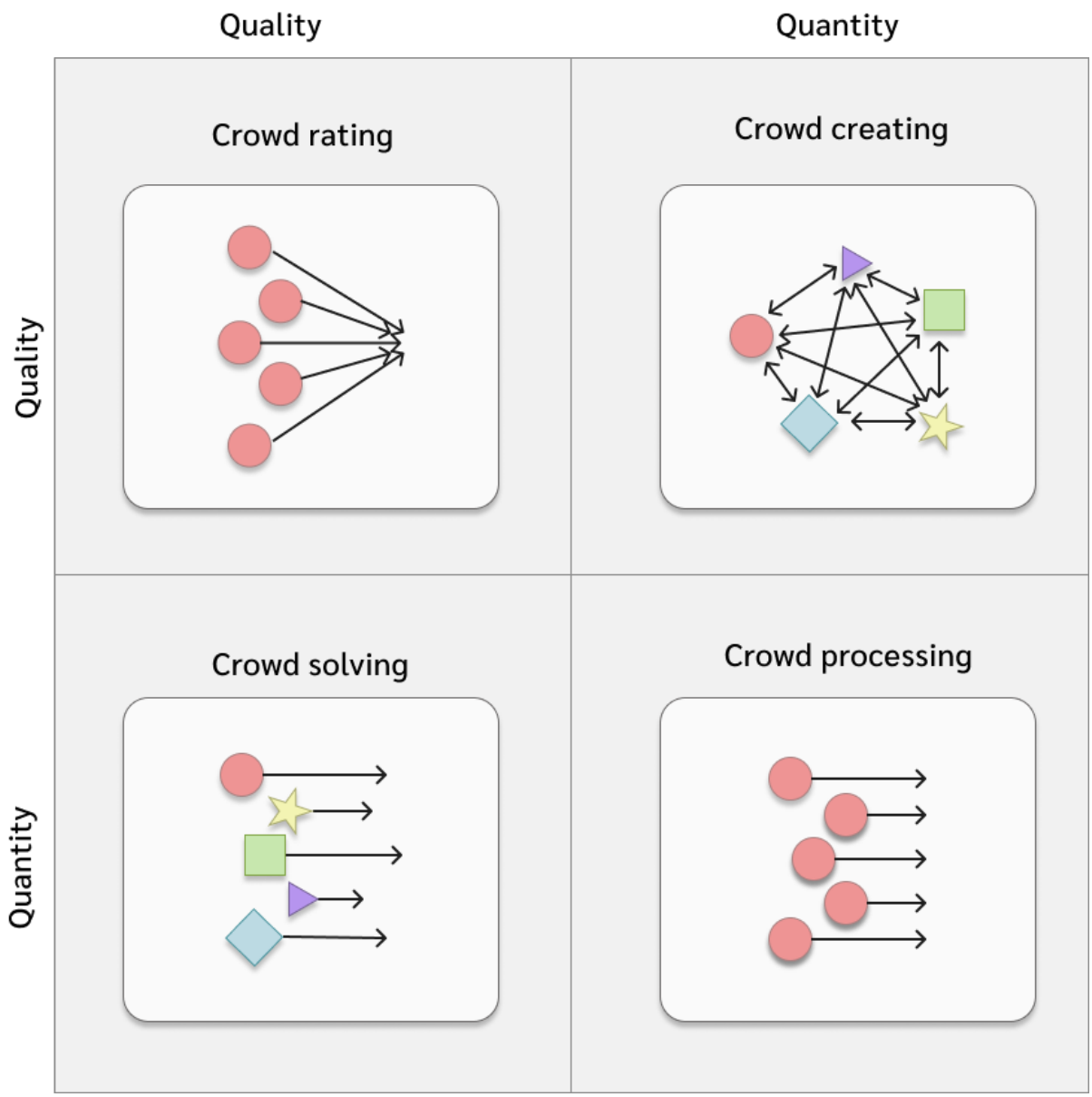

Figure 1.8: The four main crowdsourcing approaches.

1.7 Gamification

Gamification is the use of elements from game design in a non-game context to increase user activity and motivation. It is commonly used to improve user engagement, organizational productivity, learning, and physical exercise [32].

Game design elements often used in gamification are; points, leader boards, and badges. For example, Nike used gamification to motivate people to exercise by visualizing their progress, placing the users on different levels, and giving them badges for

accomplishing tasks, such as reaching a milestone of a specific total running distance [33]. Well-known examples of web pages that use game design elements for learning are; codeacadamy.com, and khanacadamy.org [34].

1.8 Related Work

In this section, examples of crowdsourcing games will be introduced. The focus will be on a subset of crowdsourcing called citizen science, also referred to as crowdsourced science. It involves both the public and amateur scientists in research.

Foldit

Foldit [35], released in May 2008, is one of the first online citizen science games to be developed. It was developed at the University of Washington by biochemists and game developers. The game has an established online community of participants, with over 240,000 registered users [35].

The goal of Foldit is to predict the structure of a protein, which was not until recently solvable by bioinformatic approaches [36]. Foldit takes advantage of humans' puzzle-solving intuitions where people compete to find the best way to fold a given protein. The players can play alone or in teams, where the protein structures that come closest to their "natural" configuration get awarded with the most points [37].

For specific complex problems, Foldit player solutions can, in some cases, outperform state-of-the-art computational methods. Khatib et al. made players write down their folding strategies as "recipes" to share with other players [38]. These recipes were shared and modified by other players, and in the end, there were two recipes that "stood out." After further studying the two recipes' algorithms, resemblances with another unpublished algorithm developed by scientists over the same period emerged. This shows that gamification is helpful in finding new solutions to protein folding and can also help discover and develop new strategies and algorithms.

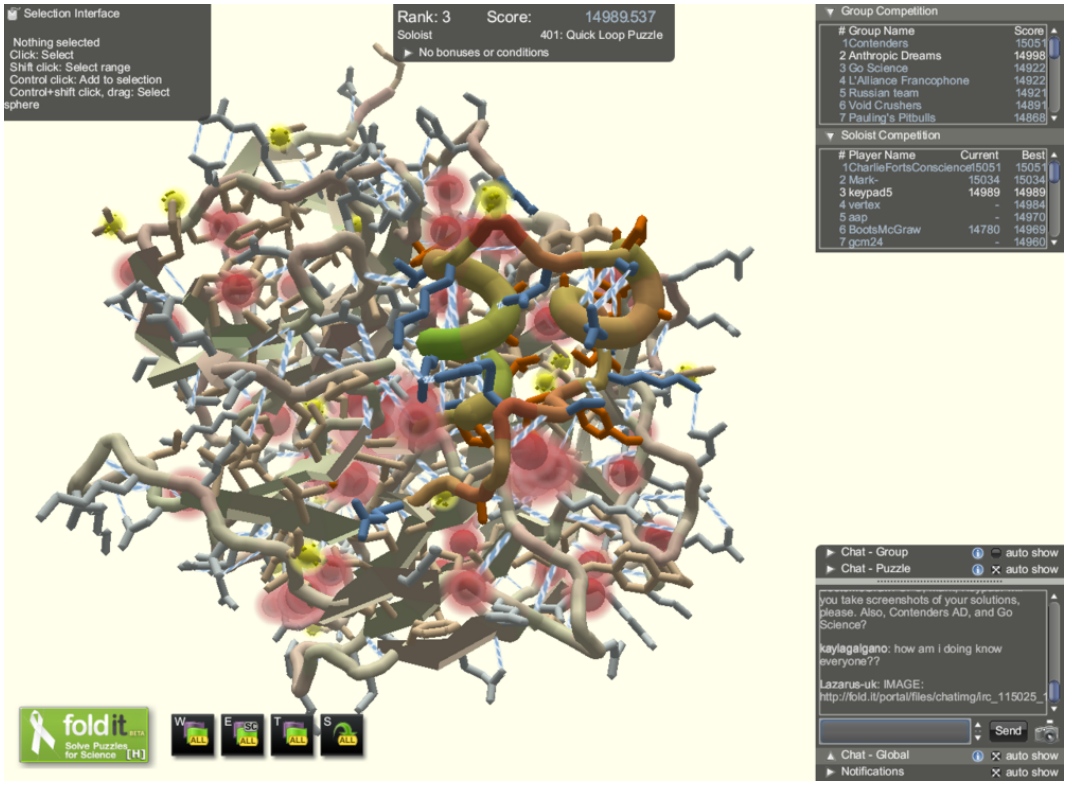

Figure 1.9: An example of protein folding in Foldit.

EteRNA

Ribonucleic acid (RNA) research has the potential to diagnose and treat diseases and develop new technologies of medicine. However, RNA is a complex molecule, and scientists do not fully understand its behavior. Creating a synthetic design to carry out a specific function and predicting how well it will perform when created is extremely difficult, even for supercomputers. EteRNA [39] is a citizen science game inspired by Foldit, supporting basic and applied research to predict and design synthetic RNA structures.

In EteRNA, players design RNA molecules by solving puzzles where each puzzle represents a targeted RNA shape or a set of design requirements. Designs voted on by the community are sent to Stanford University to be tested in the lab. Then the players receive feedback on how well the design worked such that they can improve their designs. The game has led to the development of several new RNA design algorithms [40].

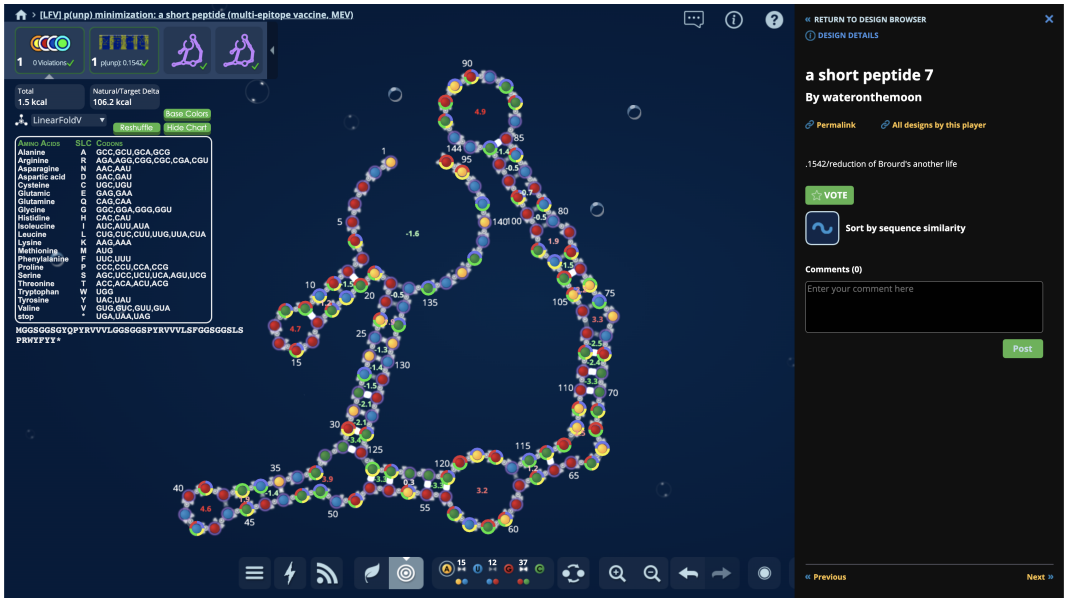

Figure 1.10: Exploration of RNA structures in EteRNA.

Phylo

Phylo [3] is a game about multiple sequence alignment optimization, developed by the McGill Center for Bioinformatics. A sequence alignment is a method of arranging DNA, RNA, or protein sequences in order to identify regions of similarity. The players solve pattern-matching puzzles that represent nucleotide sequences of different polygenetic taxa (characteristic trait in one or more populations of an organism, that is influenced by two or more genes [41]). Nucleotides are represented as different colored blocks, and the goal is to get the highest score for each sequence by matching as many colors as possible and minimizing gaps. The game uses data from the UCSC Genome Browser [42], which contains sections of human DNA that may be linked to various genetic disorders, like breast cancer [43].

The first year after Phylo was launched, they got more than 350,000 solutions, which improved the alignment accuracy by up to 70 percent. This shows that combining algorithms with crowdsourcing can help improve the solution of complex algorithms [44].

Figure 1.11: Multiple sequence alignment in Phylo [3].

Eve Online - Project Discovery

Eve Online is a futuristic, space-based multiplayer online role-playing game where the players can join several in-game activities together with hundreds of thousands of other players [45]. One of the projects in Eve Online is *Project Discovery*, an award-winning mini-game and citizen science project with different missions where the players get in-game credits when contributing [46].

Since June 2020 there is a collaboration between Dr. Andrea Cossarizza and the Project Discovery team to produce a new version of the project that aims to better understand the immune system's response to the coronavirus by analyzing data produced by flow cytometers. Each player is given flow cytometry plots over cell populations located in the blood, and the task is to draw polygons around clusters of cells. The project has gotten over 115 million submissions since 2020, which has helped produce data that would not have existed without the Eve Online community [47].

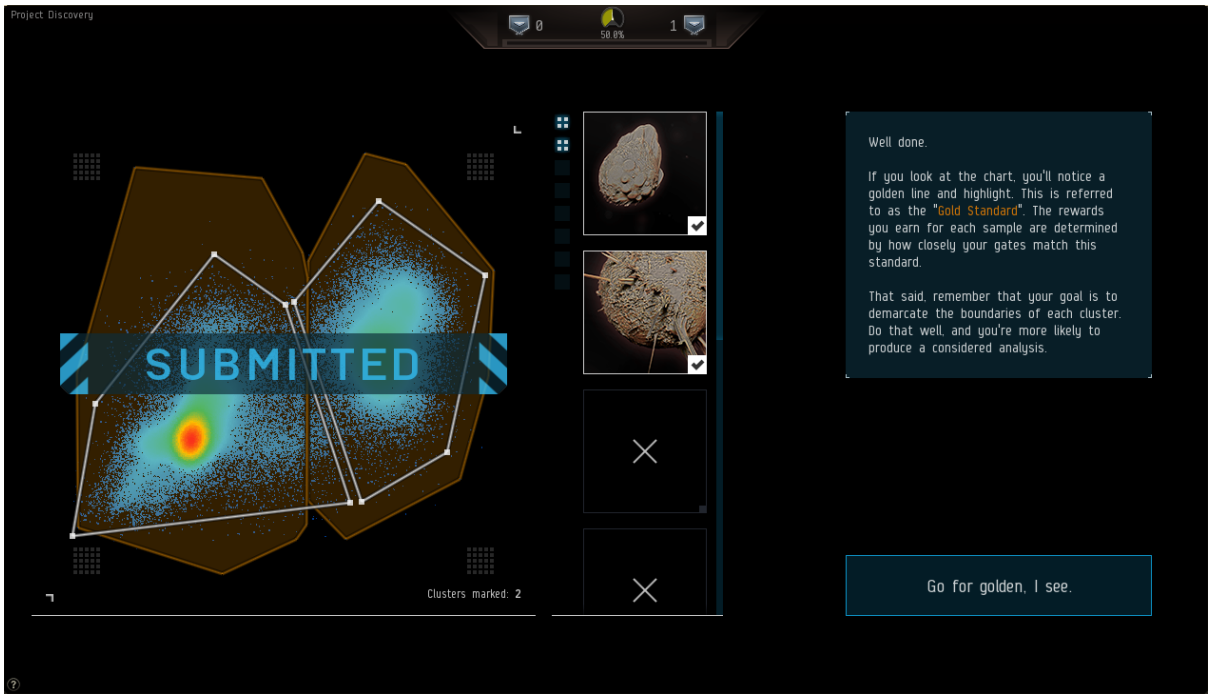

Figure 1.12: A flow cytometry sample in Project Discovery. Accuracy is determined by how close the players outline is to the "Gold Standard" that the researchers outlined [4].

Chapter 2

Methods

In this chapter, the development methods are introduced, together with the user testing approach and the technologies used in the project.

2.1 Development Methods

Using a development method is important to make progress more manageable when developing a product. We chose to follow an agile development method that allows flexibility and freedom to change elements in the project at every step of the development instead of a rigid step-by-step process [48]. An agile development approach consist of these five steps in each iteration:

- Meeting and planning.
- Design.
- Code and test.
- Release.
- Feedback.

Following an agile development method makes it possible to be flexible and regularly evaluate and adapt the project during the development phase. To manage the development, we used a project board with columns for work that needs to be done, work in progress, and the finished tasks. Each sprint started with a meeting where we planned which features we wanted to implement and decisions on how to designed them. This was added to the "needs to be done"-column. When one feature was implemented and released, it was discussed and evaluated and decided if any changes were required for the feature (**Figure 2.1**).

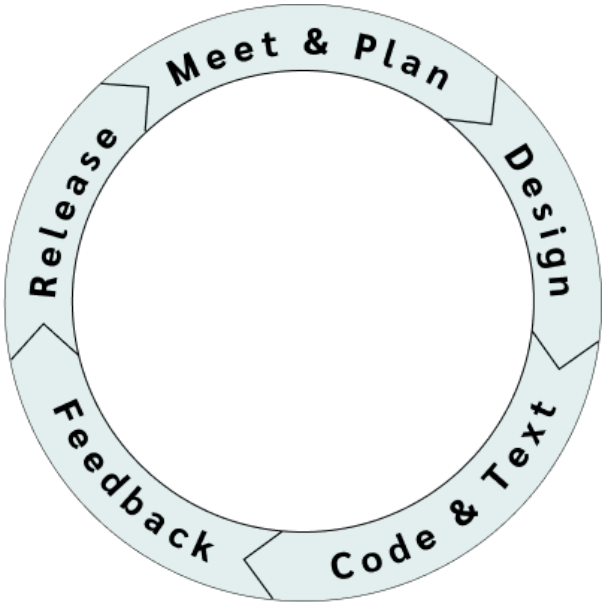

Figure 2.1: Agile development method [5]

.

2.2 User Testing

User testing is when end users test an application's function and/or interface in a realistic setting. The purpose of user testing is to evaluate the usability of the product and help understand where it has errors, missing requirements, or other issues [49].

Our aim for user testing was to see if the user understands how to play the game, score points and if any features should be added to make the game more intuitive.

We got several people to explore the game as they usually would with a new game. They did not get any information about the game before they tested it, except that the final game should be played on a smart phone or tablet, not a desktop.

While they were playing, we took notes of how they played the game. If they tried to do things that did not work, we took notes of what that was and considered features that may help future players understand how not to make similar mistakes. After the user was done with the testing, we asked if he or she had any additional feedback.

2.3 Technologies

2.3.1 Choice of Platform

There exist many gaming platforms, such as desktops, Playstation, Xbox, and mobile devices. Desktop gaming has long been the platform of choice because of its performance and graphics. Over the years, mobile technology has however improved, and most mobile phones now have significantly bigger screens and more storage capacity than they used to [50]. One of the main advantages of mobile games is that they can be played on the go. An accessible game would hopefully give us more players and more game entries, which is essential for a citizen science game.

We want a game that is easy to understand and accessible. A desktop application would therefore be too complicated for the intended game. We do not need detailed and complex graphics, and since we have limited time, it was decided to develop the game as a web or mobile application.

Since we also want the game to be easily available and played often, it made the most sense to go for a mobile application. Most people carry their phone at all times and often reach for it when they have spare time. Furthermore, it is easier to open an app on the phone instead of logging in to a web page each time someone wants to play the game (**Table 2.1**).

Table 2.1: The pros and cons for desktop, web and mobile application.

	Pros	Cons
Desktop app	- offline - more graphic/design opportunities	- can be limited by OS and hardware - may be more difficult to develop
Web app	- accessible - easy to update and deploy - can be accessed via multiple devices (phones, tablets, computers)	- online, must adapt to the user's device
Mobile app	- online/offline - accessible - mobile	- design must be simple/is limited by the space/ performance - rely on user to download updates

2.3.2 Choice of Framework

Before starting to implement the game, we had to decide which framework to implement it in. The two main options considered were React Native (a general development framework) and Unity (a game engine).

React Native is a framework created by Facebook in 2013. It uses JavaScript, rendered into native code for iOS, Android, and Windows. This makes it easy for developers to make apps for multiple platforms without adding much native code. Some popular apps made with React Native are Facebook, Instagram, and Discord [51].

React Native uses components that let the user split the user interface (UI) into independent functions. Components in React and React Native are JavaScript functions. The components can use props (i.e., properties) that are input for the functions. Components can also use states, similar to props but private and fully controlled by the components [52].

Unity is a game engine and IDE (Integrated Development Environment) with a publicly available free version. It is used for making applications in 2D, 3D, and simulations in augmented reality (AR) and virtual reality (VR). The applications can be deployed on over 20 platforms, including mobile platforms, desktop games, web applications, and consoles [53]. It comes with complete documentation for its application programming interface (API). Examples for most use cases are available in its documentation. Unity is prevalent and, therefore, has very active online forums that have accumulated the most questions and answers a new user is likely to encounter.

The Unity Editor is user-friendly with its intuitive system for adding elements to a scene. It also supports drag and drop functionality. It is a visual tool where the user can draw objects and place them on a canvas and then add functionality to these objects through scripts written in C#.

Popular games made in Unity include Valheim (desktop), Beat Saber (VR), Among Us (desktop, mobile), and Pokemon Go (mobile) [54]. Besides games, Unity has also been used in different industries such as film, automotive, architecture, engineering, construction, and research to, for example, simulate surgeries [55].

To select the most suitable framework we tested React Native and Unity by making basic prototypes. Even though React Native is popular for making mobile apps, it is not

Table 2.2: The pros and cons for the framework React Native and game engine Unity.

	Pros	Cons
React Native	- popular in the industry - lots of tutorials and documentation - reusable components	- not suitable for making advanced games
Unity	- mix between coding and a graphical user interface builder (GUI) - made for making games - lots of tutorials and documentation - developer friendly - easy to deploy to different platforms - easy to install and setup - has built in simulators, so it is possible to see how it looks on different devices without needing to download it	- less used in industry, except for game development and visualization

explicitly made for games. This was something we noticed when making the React Native prototype, as it was difficult to get references to game objects, partly as the programming language was unfamiliar. Unity is a game engine and thus felt much more suitable. It was overall easier to use and more flexible. We therefore decided to use Unity for developing the more advanced prototype (**Table 2.1**). The Unity version used is 2021.2.4.

2.3.3 Mascot Generic Format Files

A Mascot generic format file (MGF), is one of the standard formats for MS/MS data in proteomics. The MGF file contains one or more MS/MS peak lists. Each MS/MS peak list starts with a header containing basic information about the precursors, where the minimum information included is spectrum title, precursor mass, and precursor charge. After the header comes the data, consisting of the m/z and intensity of each peak [56] (**Listing 2.1**).

Listing 2.1: MGF File Example with one MS/MS peak list.

```
1
2 BEGIN IONS
3 TITLE=controllerType=0 controllerNumber=1 scan=7032
4 RTINSECONDS=1728.31038
5 PEPMASS=794.931640625
6 CHARGE=2
7 110.0716934 4153.9423828125
8 116.0707169 2535.9174804688
9 120.0811691 5449.2260742188
10 ...
11 1462.7722168 14497.068359375
12 1470.7617188 5701.6831054688
13 END IONS
```

2.3.4 JSON

JavaScript Object Notation (JSON), is a standard file format used to store and transmit data objects consisting of name-value pairs and arrays. JSON is language-independent, which makes JSON an ideal data-interchange language [57]. We used JSON to store the processed data to be able to easy access it and make game objects from it.

2.3.5 Version Control

In addition to the technologies mentioned above, we selected GitHub [58] as the version control system in the development process. We chose GitHub because of its availability and portability, and its user-friendliness. GitHub is generally used to manage the development process and to organize other users' contribution, such as editing and testing.

Chapter 3

Implementation

In order to explain the implementation choices for the prototype, this chapter starts by presenting the main idea and concept of the game. Next is an overview of the design challenges and solutions, followed by the implementation structure, containing the setup in Unity and the prototype.

The complete source code for the project can be found on `https://github.com/barsnes-group/DeNovoGame`. An executable version of the prototype for testing can be downloaded from the same page.

3.1 Main Idea

De novo sequencing of MS/MS spectra is a method for determining a sequence of amino acids from a mass spectrum (Section 1.5). PeptideShaker is a proteomics informatics software used to analyze and identify peptides and proteins [59]. **Figure 3.1** shows an example data set from PeptideShaker showing all of the proteins found in the data set, along with the peptides and peptide-to-spectrum matches for the selected protein and peptide.

PeptideShaker automatically annotates the spectrum with the most likely ions given the spectrum, and the peptide sequence is identified. PeptideShaker can use de novo sequencing to annotate the spectrum if the spectrum is not fully identified. There are two ways of doing this, either automatically or manually. In the automatic approach,

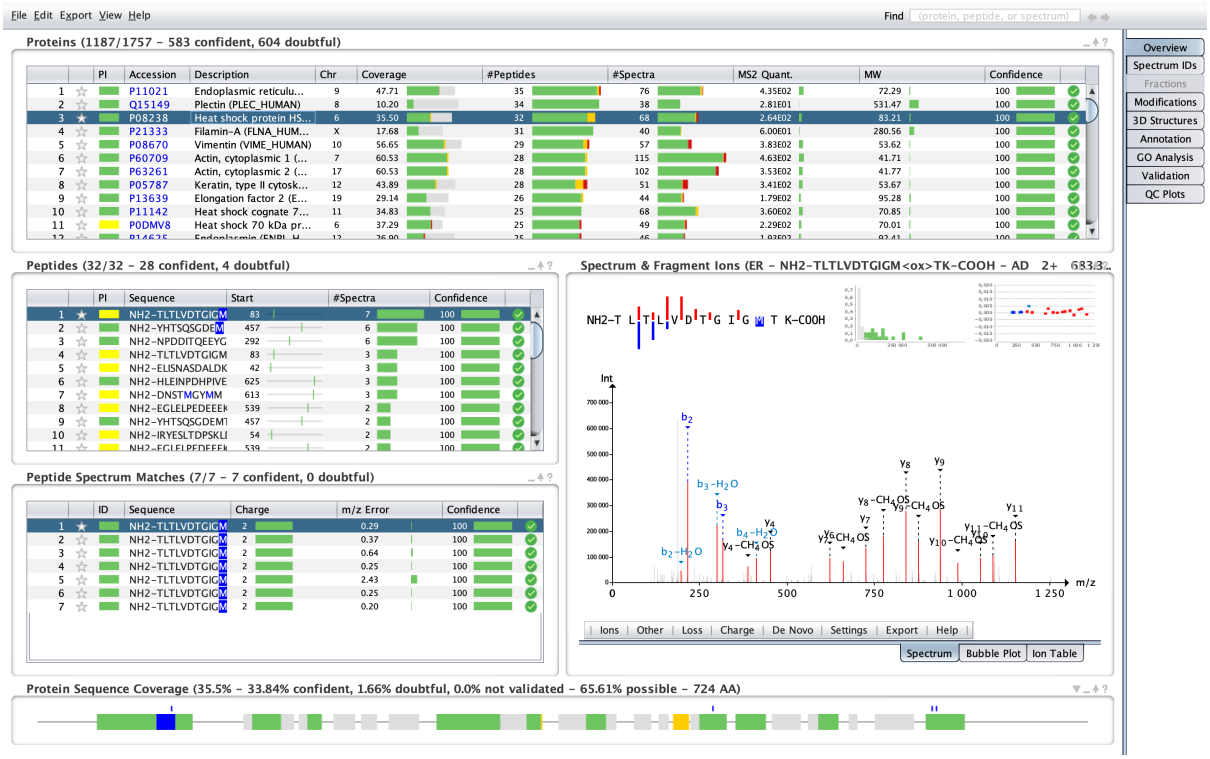

Figure 3.1: Example data set from PeptideShaker.

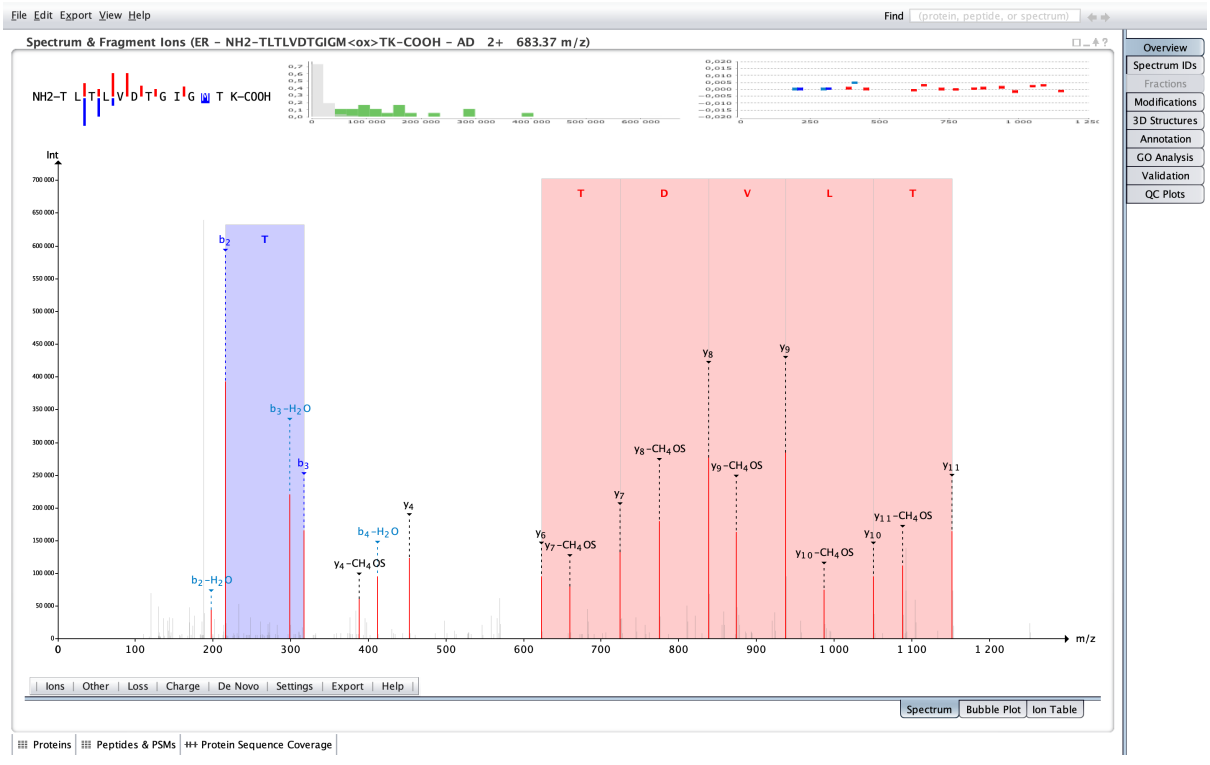

Figure 3.2: Using the automatic approach for de novo sequencing in PeptideShaker

PeptideShaker tries to find matches against the sequence identified by the peptide-to-spectrum matches (PSMs), as shown in **Figure 3.2**.

In the manual approach, the user can select one peak, and if the distance to a second peak matches an amino acid mass, the amino acid will be shown as in **Figure 3.3**. The figure shows that the amino acid "T" matches the distance between $b_2 - H_2O$ and $b_3 - H_2O$, and the amino acid "I" or "L" match the distance between $b_3 - H_2O$ and $b_4 - H_2O$. We want to explore if we can make a citizen science game that does something similar [59].

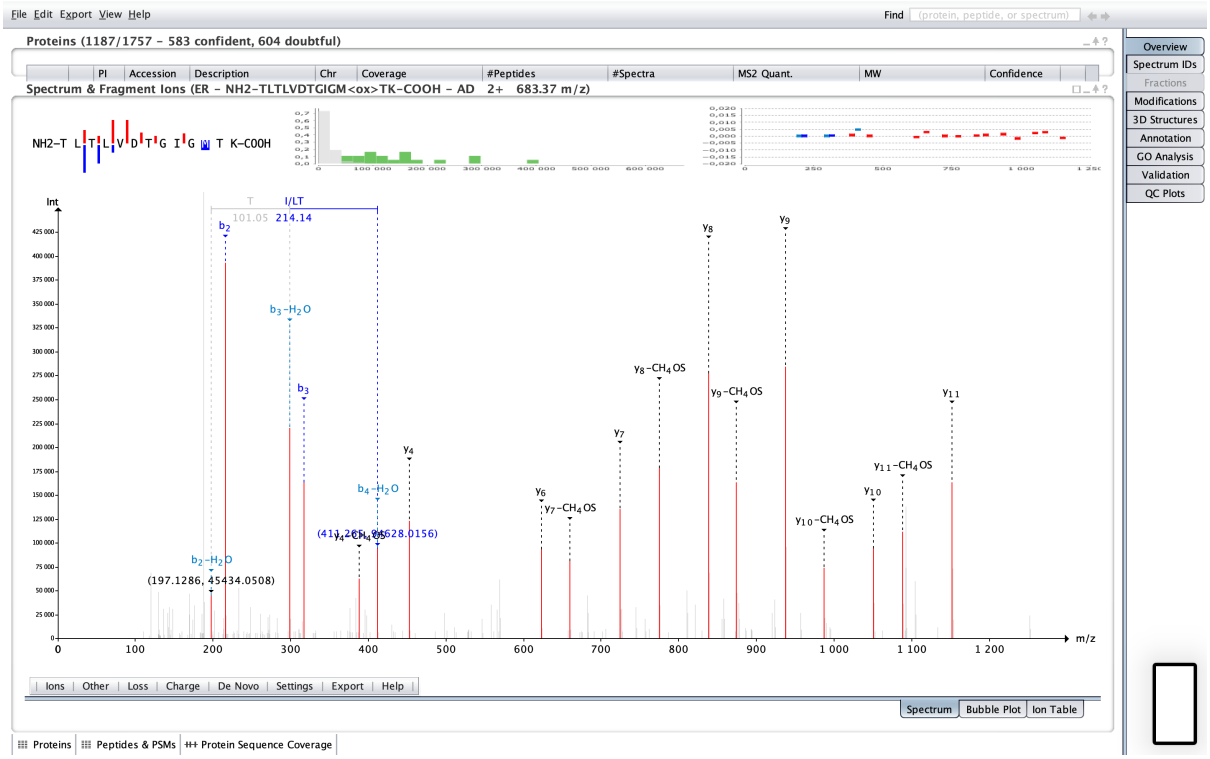

Figure 3.3: Using the manual approach for de novo sequencing in PeptideShaker.

The game's concept is to find the amino acid sequence that minimizes gaps and identifies as many amino acids as possible to get a high score. We want to hide the underlying proteomics in the game because it is unnecessary for the user to know the background to understand how to play. However, we want to have an About page that summarizes the background so the users understands that they are a part of a citizen science project. We also want to add a button such that the users have the option to see the amino acid sequence they have made. This makes it possible for the users to connect the game to the science.

3.1.1 Game Concept

Given that mass spectra contain lots of data and may have some noise, we first have to filter the data. This was done by only keeping the peaks with an intensity higher than a given threshold and the peaks where it is possible to place an amino acid in-between them.

Figure 3.4A shows a mass spectrum before filtering. There are too many peaks, and it is difficult to read the data. To make the spectrum easier to read, the noise and peaks with low intensity are removed. The default intensity threshold used for preprocessing is 85%. It removes the noise but still leaves enough peaks to be able to make an amino acid sequence. In addition to removing peaks below the threshold, the peaks where no amino acid can fit in-between are also removed. Two peaks where the distance is equal to the mass of an amino acid plus/minus the accuracy of the instrument will here be referred to as a *slot*. All the peaks that do not make up a slot are removed. **Figure 3.4B** shows a mass spectrum where all the grey peaks are the ones that are removed during the preprocessing. The intensity threshold is indicated by the dashed red line. The peaks over the threshold (in grey) are removed because the peaks does not make a valid slot.

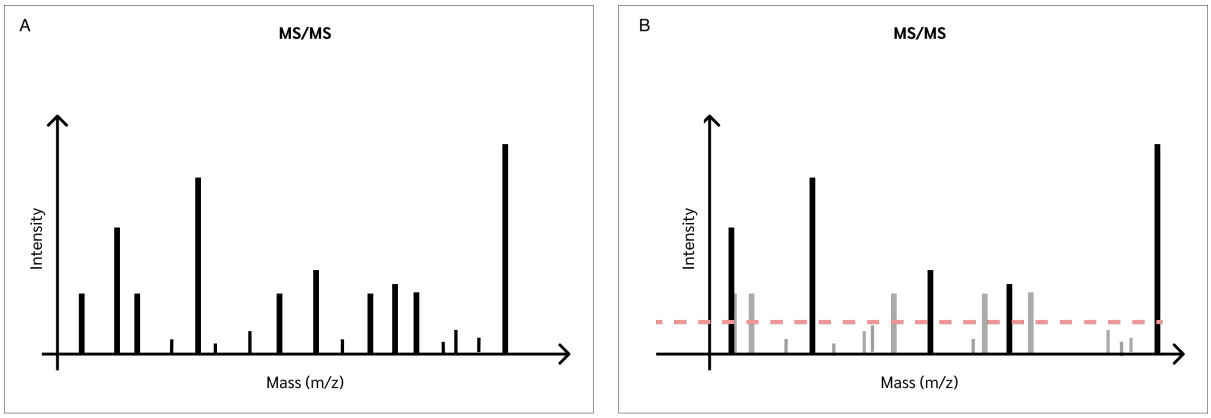

Figure 3.4: Mass spectrum before (A) and after filtering (B).

Figure 3.5 shows a filtered mass spectrum and indicates the five possible slots after filtering. The height of each slot is given by the average intensity of the peaks on each side. Note that slots can overlap.

Figure 3.6 shows four example boxes representing individual amino acids. Slots 1 and 3 are highlighted when the purple box is selected because the amino acid's mass is the same as the width of the slots in the spectra plus/minus a threshold determined by

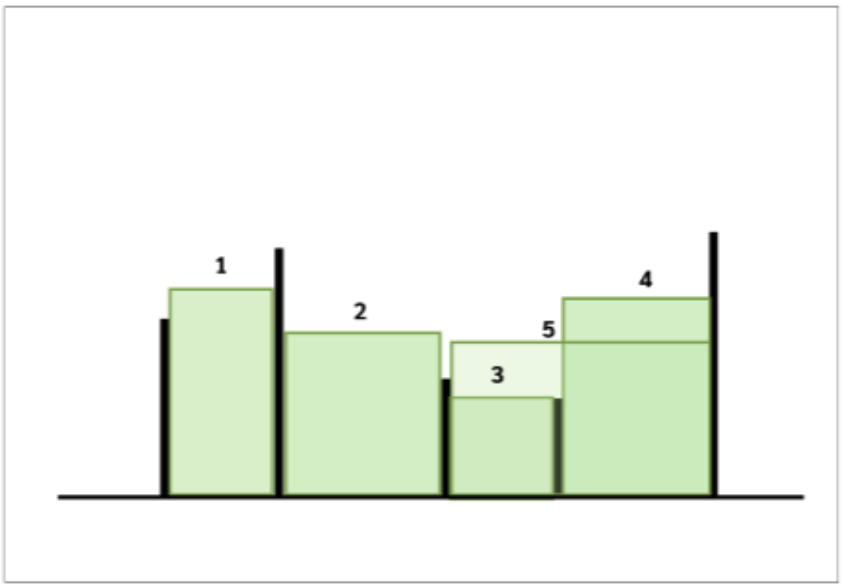

Figure 3.5: All valid slots for this playing board.

the instrument's accuracy, which is 0.02 in this case. The highlighted slots are the slots where the selected box can be placed. When a box is placed, the height and width of the box are changed to the same as the slot it is placed in.

The height of each slot determines the score of the game. Each slot is made up from two peaks, where each peak has an intensity. The score of a slot is calculated by taking the average intensity of the peaks. For example, if a slot is made from a peak of intensity 1 and a peak of intensity 3, the score of the slot will be 2.

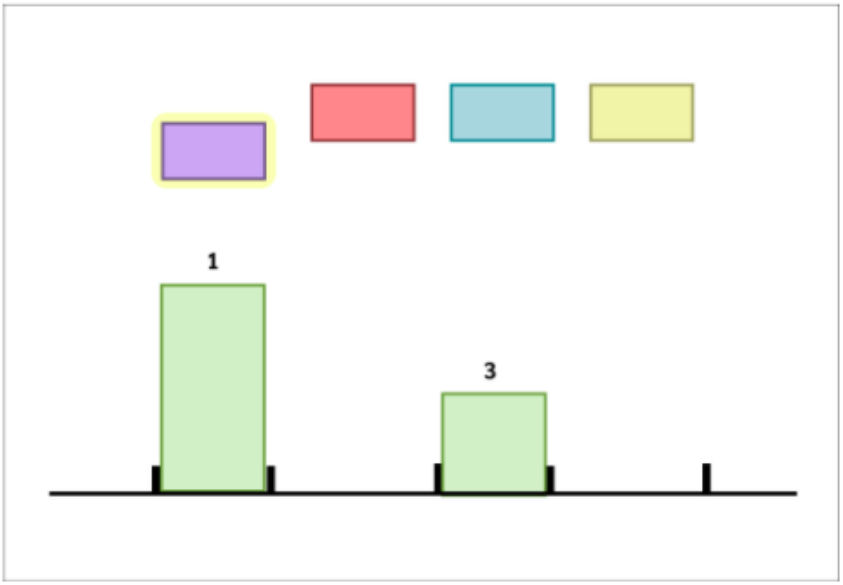

Figure 3.6: When a box is picked up, the slots where the box can be placed are highlighted.

In this example, Slot 1 gives more points than Slot 3 because it is higher. This also means that placing multiple and higher boxes gives more points than few and wide boxes. **Figure 3.7** shows the two solutions for the given puzzle. Either the user can place the yellow box, which makes peak number four unavailable (**Figure 3.7A**), or the user can place the purple and the blue box (**Figure 3.7B**), which use more of the possible slots resulting in a higher score.

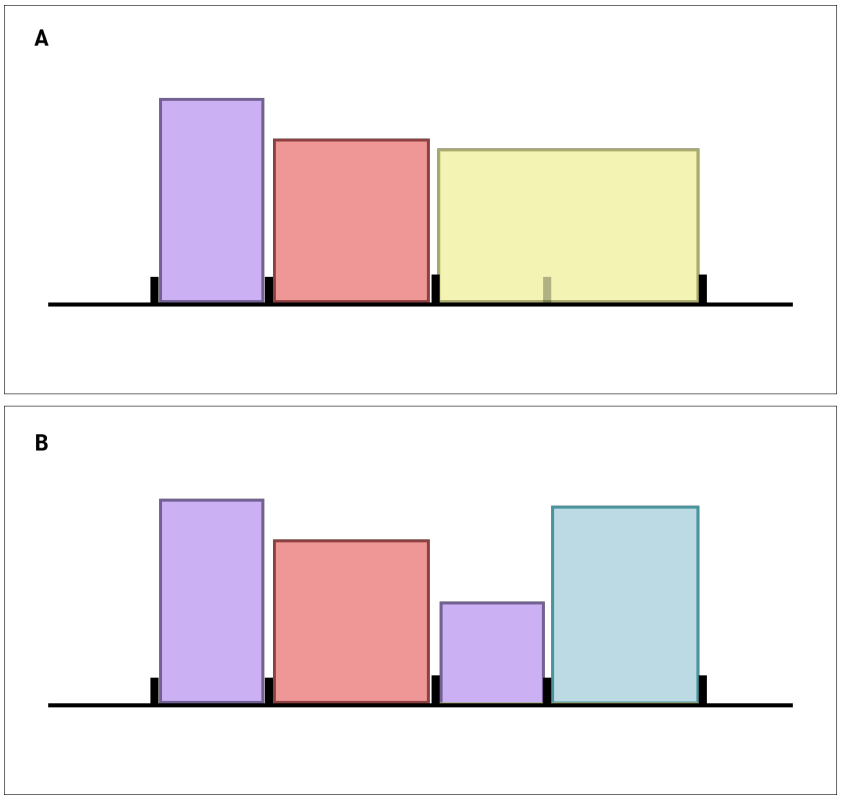

Figure 3.7: Two possible solutions for the given puzzle.

3.2 Design Challenges and Solutions

3.2.1 Game Design

As already mentioned, we have hidden most of the science from the players to make the game as simple as possible because we do not think the players are required to know the science to know how to play the game. Making it a science game may limit the audience because the science may be difficult to understand for people with no background or interest in the topic. This means that the game would need to use time teaching the player about the background before they can start playing and thus may loose players that are not interested in the background.

Box Design

First, we tried to have boxes with the same width and height as the amino acid they represent. Since the amino acids sizes are normalized to fit the screen better, there was almost no difference in the widths of the boxes, which made it impossible to tell the different boxes apart. Therefore there was no point in having the boxes be different sizes. A decision was made to instead have all the boxes be the same size and rather use different colors to distinguish them. The colors of the first boxes are from a list of chosen colors to make sure that the colors have enough contrast from each other to distinguish them. If there are more boxes than colors in the list, the next box is given a random color.

Peak and Slot Design

The peaks from the spectra are all set to the same height in the game. We think it would be too messy if the peaks are the same height as they are in the original spectra, and it would be difficult for the players to see which peaks make up a given slot.

First, we tried to highlight the peaks in a different color to signal where the player could place a box. However as slots can overlap, it was hard to see where one slot started and ended. The original plan was to color-code the peaks based on their intensity, but this would not work because several slots can start and end at the same peak.

We ended up making the slots as boxes where the height is set to the average height of the peaks. This made it easier to see each slot, even when there are slots that overlap. It is also easy to see which slots have the highest intensity, which also is the slot that gives the highest score.

The valid slots have a green color to indicate that the player can place a box there. We have also chosen to show the invalid slots in orange, so the player can see if they get more points for changing the position of the boxes on the playing board.

Numbers Above the Boxes

To provide players with more information on where and if it is possible to place a given box, we decided to add the number of valid slots the box can be placed in and the number of total slots for that box. The numbers above all the affected boxes will be updated when a box is placed. This may also help players that have difficulties distinguishing between green and orange slots to get information about the possibilities of where a box can be placed.

Resulting Amino Acid Sequence

Since the game, first and foremost, is a citizen science game, we want the players to have the possibility to see the connection between what they are doing and the science behind it. Therefore we added a button where they can see the amino acid sequence they have made from placing the boxes.

We also added the mass gaps to get the correct, complete sequence. A sequence often starts with a mass gap, which is the equivalent to the start coordinate of the first placed box. To calculate the mass gaps between the boxes, we took the second box's start coordinate minus the first box's end coordinate. The sequence usually ends with a gap. This is calculated by taking the precursor's mass multiplied by the charge minus the end coordinate of the last box.

The mass of the amino acids plus all of the mass gaps are equal to the precursor's mass multiplied by the charge of the precursor.

Hints Before Game Start

We predicted that most players would not go to the About page before starting to play. As a quick fix, we therefore added text at the start of the game with hints on how to get started. The text fades after a few seconds. When having these types of hints at the start of the game, it is essential that the text is clear and explains exactly what the user should do to get started in order to prevent the players from getting confused by the hints.

Mobile vs. Desktop App

As mentioned in Section 2.3.1, we decided that the game is best suited as a mobile application game. We want people to be able to test the game without having to download any external tools or programs.

To test the game as an Android application, we tried online Android emulators and BlueStacks [60], a program that makes it possible to run Android applications from the computer and the cloud. The user should be able to just download the program and then run the *.app* file or run it in the browser. This did not work due to trouble with the Android version BlueStacks accepted and the Android versions Unity can make.

We also tried Android Studios [61], which did work. But the program is quite large and needs several steps for setting it up before being able to run the app, which is too much to demand from the user.

To make it easier for others to test the game, we therefore changed the Unity Android build to a desktop build but still developed the game as a mobile application. Now the user only needs to download an executable file and run it to be able to test the game.

File Format

When we first implemented the script for preprocessing the data from the MGF file, we thought that we only needed a CSV file with the peak coordinates for developing the prototype in Unity. We soon discovered that it was not enough and that we also needed information about the amino acids and each associated slot.

We therefore decided to use JSON as our output file format, as it can contain more data and link it to an object. After altering the code to use JSON, it was straightforward to add more data that we later figured was needed in the game.

3.3 Preprocessing the Data

In the prototype, we downloaded spectra as MGF files (Section 2.3.3) from the PeptideShaker example data set as the data for developing the game. We used Python to filter the selected spectra. First, the data was filtered on the intensity of the peaks to

remove all the peaks under a given percentile threshold. We used a percentile threshold of 85%. Then, we removed all the peaks where an amino acid does not fit in-between. Since the number of peaks after filtering on intensity was relatively low, we used brute force to check if the amino acids could fit between two or more peaks plus/minus a given threshold of 0.02.

After the filtering, we normalized all of the peaks and amino acids to make it possible to use data from different data sets. We normalized the data by dividing all the peaks and the mass of the amino acid by the highest peak, the "MaxXValue" in the JSON file (**Listing 3.1**). Then we made slot objects from the normalized data that contains the slot's start peak index and coordinate, the end peak index and coordinate, and a list of intensities. This resulted in a JSON file that contains the amino acid letter, the mass, the maximum x value, which we later use to get the original numbers, the peptide mass from the MGF file, and a list of valid slots (**Listing 3.1**).

Listing 3.1: JSON file containing the preprocessed data


```
{
"AminoAcidName": "I/L",
"Mass": 7.736,
"MassOriginal": 113.0841,
"MaxXValue": 14.617685547,
"HighestPeakFromMGF": 1589.86328125,
"Slots": [
  {
    "start_peak_index": 7,
    "start_peak_coord": 78.372,
    "end_peak_index": 9,
    "end_peak_coord": 86.108,
    "intensity": [12.762, 32.643]
  },
  {
    "start_peak_index": 1,
    "start_peak_coord": 39.426,
    "end_peak_index": 2,
    "end_peak_coord": 47.162,
    "intensity": [17.142, 28.634]
  },
  {
    "start_peak_index": 3,
    "start_peak_coord": 53.939,
    "end_peak_index": 4,
    "end_peak_coord": 61.675,
    "intensity": [18.729, 35.171]
  }
]
```

3.3.1 Setting up Unity

Scenes

In Unity, Scenes are where we work with content. They are assets that contain all our parts of a game or application. One scene may be enough for a simple game, and more complex games may need one scene per level. In our prototype, we have four different Scenes: one scene for the start screen, one about scene, and one for each level. Each scene contains a hierarchy with game objects and components.

Hierarchy

The Hierarchy window in Unity displays every GameObject in a Scene, such as models, Cameras, and Prefabs [62]. The two levels in the prototype are built alike, so they have the same hierarchy. The hierarchy consists of the main camera, the game objects Score, GameController and Warnings, a ParentContainer, an EventSystem, background color, and a Canvas which contains UI elements such as buttons and text (**Figure 3.8**).

We have to make GameObjects to attach scripts that we want to use. The Score GameObject has a Score script attached to it, making it possible to display and update the score when the player makes changes in the game. Alike is the Warnings script attached to the Warnings GameObject.

The GameController is the main GameObject in the hierarchy (**Figure 3.9**). It contains the GameController script and the JSONReader script, where the JSON file is attached as a component to the JSONReader script. The JSON file is the input data for the game. By using different thresholds and MGF files when preprocessing the data, we get different JSON files to make more levels. It is also possible to customize the size of the peaks and boxes to each level in the GameController component. When using an input file with fewer peaks and boxes, it is possible to edit for example the size and space between the peaks in the GameController component by changing parameter such as the "Peaks Space" and "Scale Width" (**Figure 3.9**).

Continuing in the hierarchy is the ParentContainer with the PeakContainer, BoxContainer, and ValidSlotsContainer. Game objects such as the draggable boxes, peaks, and slots are created here at runtime. We have chosen to make containers for each game

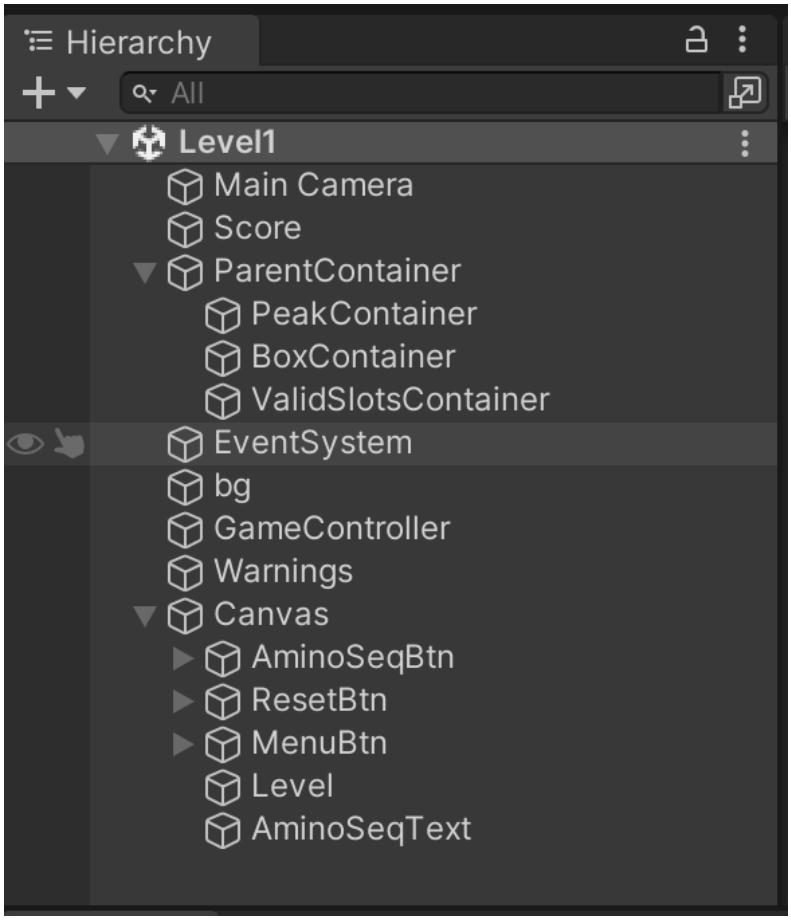

Figure 3.8: Hierarchy of Level 1.

object to make it easier to access them; for example, when we select a box, all the matching slots are created in ValidSlotsContainer. When we remove the pointer from the box, we want the slots to disappear. We do this by clearing all the game objects from the container.

The EventSystem is responsible for processing and handling events in a Unity scene [63]. It is automatically created when a user interface (UI) element such as a Canvas is added to the scene. The Canvas contains UI elements such as buttons and text.

Prefabs

A prefab allows us to create, configure, and store a game object complete with its components, property values, and child game objects as a reusable asset. The prefab asset works as a template from where we can create new prefab instances in the scene [64].

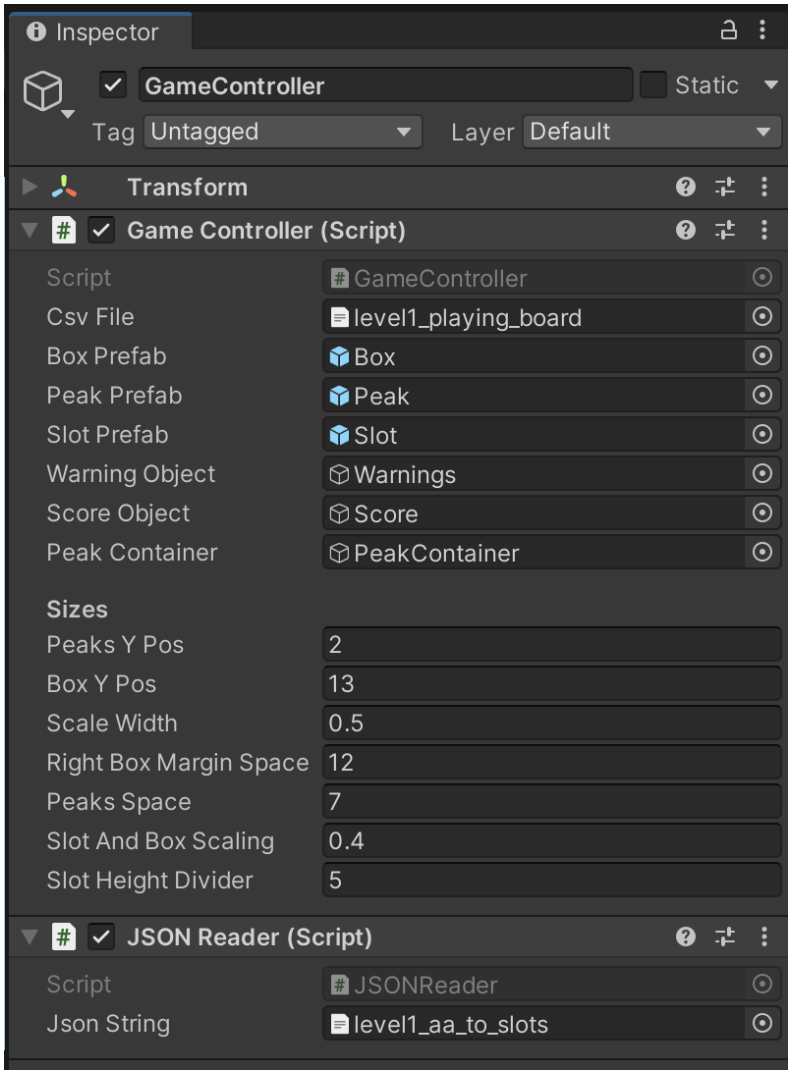

Figure 3.9: The GameController component.

Prefabs have been created for the boxes, the peaks, and the slots. The box prefab contains the DraggableBox script, a box collider, a text component, and a sprite (a 2D graphic object [65]). The box's prefab is initiated in the GameController script, where we set the scale and position for the boxes. Having prefabs makes it possible to change the sprite of the prefabs to, for example, pictures that look more "game-like."

Scripts

The game application consists of the classes: AminoAcidSequence, ButtonsScript, DraggableBox, GameController, JSONReader, Menu, Peak, Score, Slot, and Warnings, as seen in **Figure 3.10**.

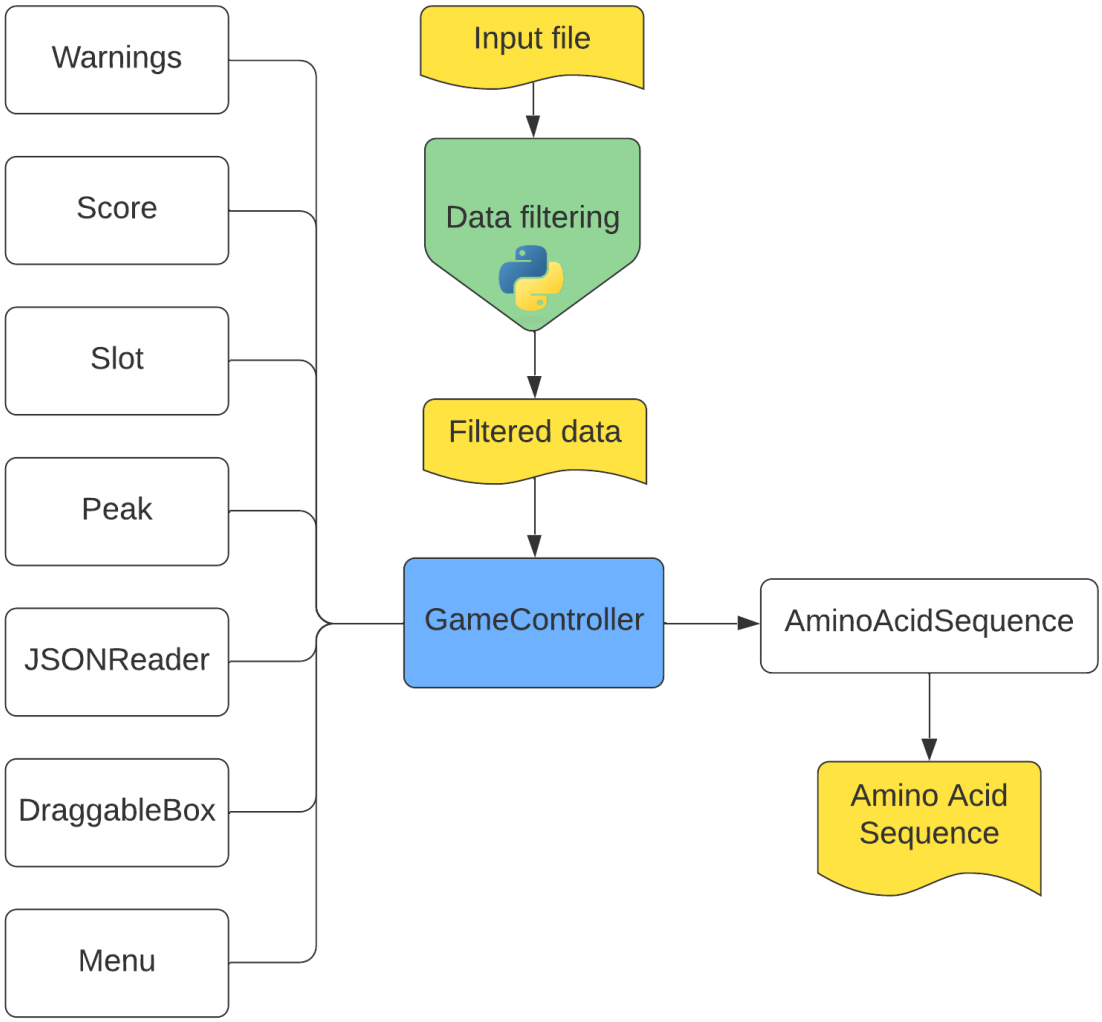

Figure 3.10: Overview of the project structure.

GameController is, as the name implies, the controller of the game or the main script. In GameController, most of the game logic is handled, and prefabs are initiated and drawn at run time. It also handles the actions for when a box is placed in a slot, including spawning a new box, changing the scale of the box to the scale of the slot it is placed on, and checking if the slot is valid and not occupied.

The AminoAcidSequence class gets the amino acids from the placed boxes and calculates the gaps in front, in-between, and behind them. It also has a method that writes the results to a file, which later can be used to process the results from the game.

DraggableBox is the class that handles the logic for the box and the actions when the mouse is pressed. This class makes it possible to move the boxes and snaps them into

the correct position if they are dropped in a valid slot; if not, the box is returned to its starting position.

The JSONReader class has a Serializable slot class and an amino acid class. A Serializable class makes it possible to store objects as text or make objects from text. Here, it is used to make C# objects containing the information we obtained in the Python script stored in the JSON text file.

The Score class calculates and updates the score by looping through all the boxes placed on valid slots and calculates the average intensity of the peaks where the box is placed. While the Menu, Peak, Slot, ButtonScript, and Warnings classes are minor classes that help control game objects:

- The Menu class loads the different scenes.
- The Peak and Slot class helps control and change, for example, the size and position of game objects or get information about the elements of the game object.
- The ButtonScript is used to handle the actions when a button is selected.
- The Warnings class makes it easier to handle invalid placements in the game. Warnings such as text or sounds can be added.

Unity Development Challenges

The order of execution for the event functions Awake and Start has to be considered in order to get the build of the application to look and work the same as in the editor. The Awake function is called for each object in the scene at the time when the scene loads, while the Start function is called before the first frame or physics update on an object. If we, for example, try to make the boxes in the Awake function and read the JSON file (with all the information needed to make the boxes) in a Start function, we will get an error. The Awake function is called before Start, and the information we need to make the boxes is not yet available. To ensure that everything implemented is in the build, it is therefore wise to have a working build from the start that is updated regularly, especially in a game where all the game components are made at runtime, like in the game we have made.

3.4 The DeNovoGame Prototype

When a player starts the application, the first thing they see is the home page, where they can access the levels, the About page, and exit the application (**Figure 3.11**).

Figure 3.11: The start screen of the DeNovoGame.

The About page has information on what the aim of the game is, how to play it, a summary of the background, and figures that illustrate the process of how the game is made (**Figure 3.12**).

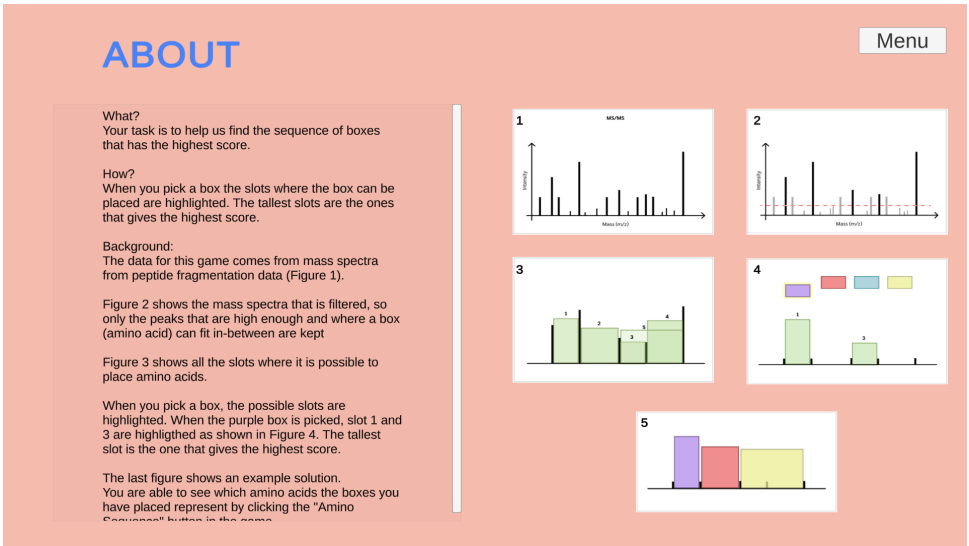

Figure 3.12: The About page.

Figure **Figure 3.13** shows the first level in the game. At the lower part of the screen is the line with the peaks, and above them are the boxes with the numbers of possible

slots and the number of total slots. The score is displayed in the upper left corner, which is updated when a box is placed or removed from the line with the peaks. In the upper right corner are the amino acid sequence, reset, and menu buttons.

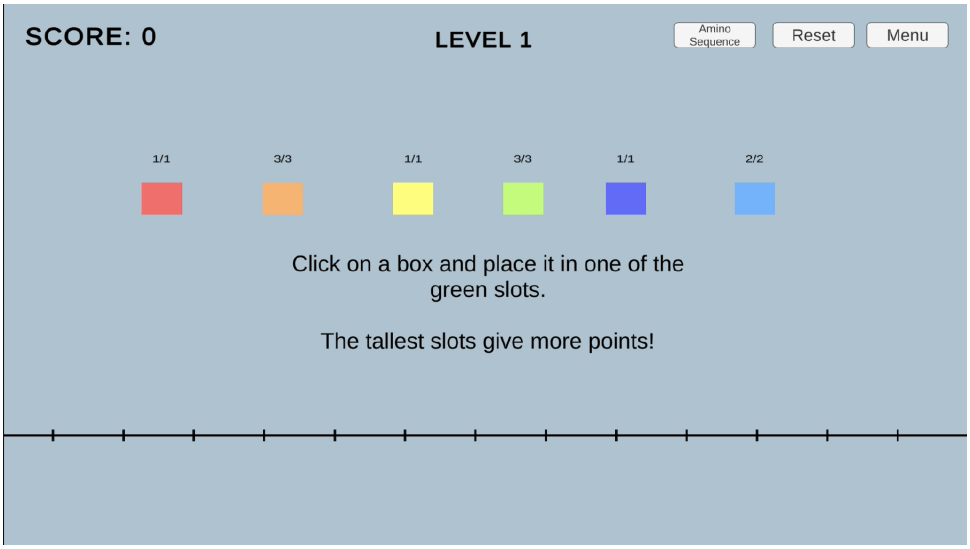

Figure 3.13: Start of Level 1.

When a box is selected, the valid and invalid slots are displayed. The valid slots are green, and the invalid orange. If the user tries to put a box on an invalid slot, it returns to its starting position (**Figure 3.14**). If a box is placed on a valid slot, the box changes size to the same as the slot's, and the score is updated. If it is possible to place the same box in several places, a new box of the same type is spawned at its starting position.

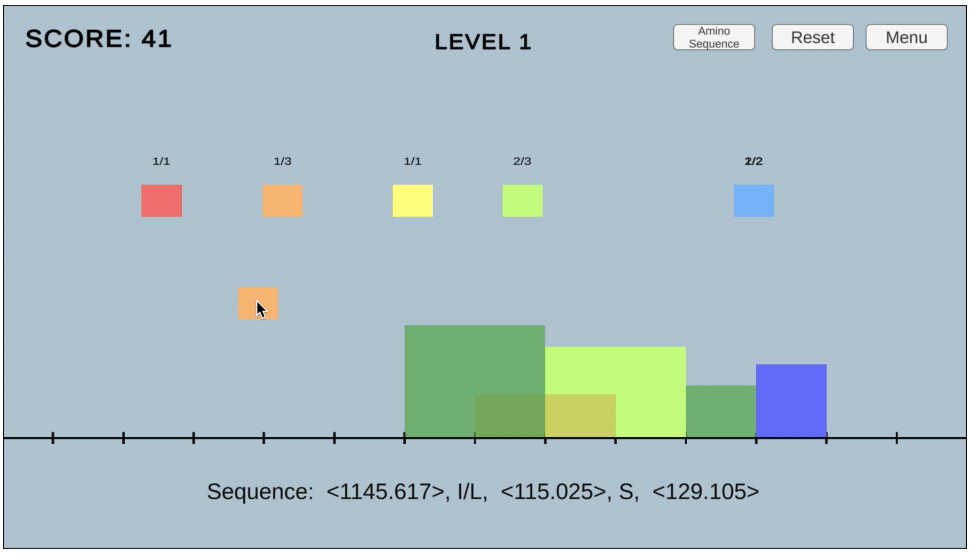

Figure 3.14: Valid (green) and invalid (orange) slots when a box is selected.

When the amino acid sequence button is clicked, the sequence representing the boxes

and the gaps is displayed (**Figure 3.15**). The amino acid sequence and score are updated when boxes are removed from the board as shown in **Figure 3.16)**.

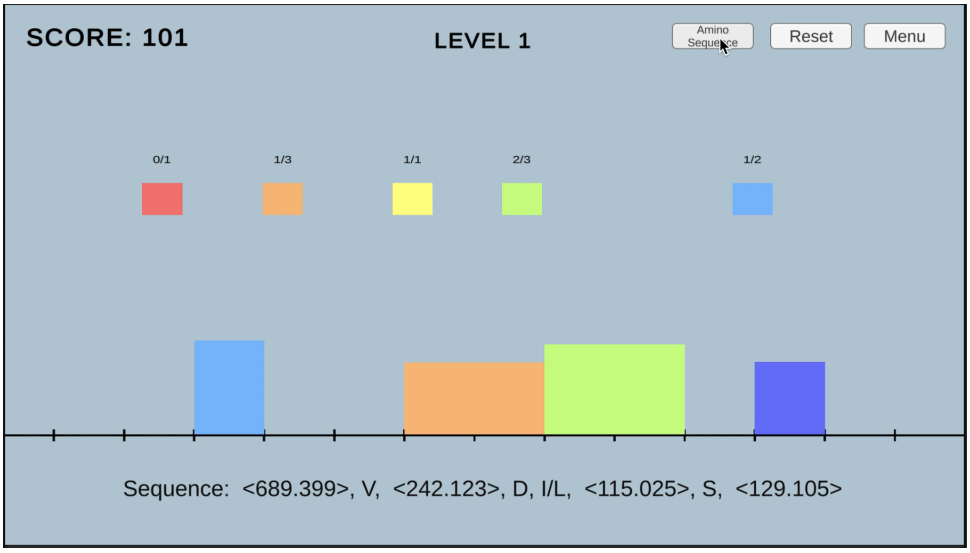

Figure 3.15: Shows the amino acids sequence of the placed boxes.

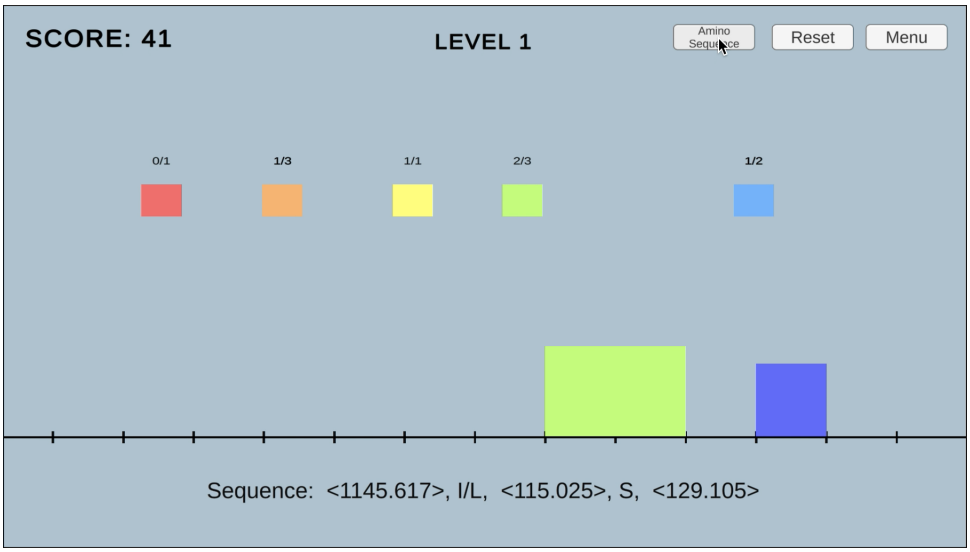

Figure 3.16: The sequence and score is updated when boxes are removed.

Figure 3.17 shows the Level 2 playing board. It is built like Level 1, but the data for making the game is changed to a different spectrum. This means that the boxes on this level represent other amino acids than the previous level, and the slots are also different. The prototype only has two levels to illustrate that it works with more than one spectrum. The plan for the real game is to extend the number of levels, which is easily done because of the way it is implemented.

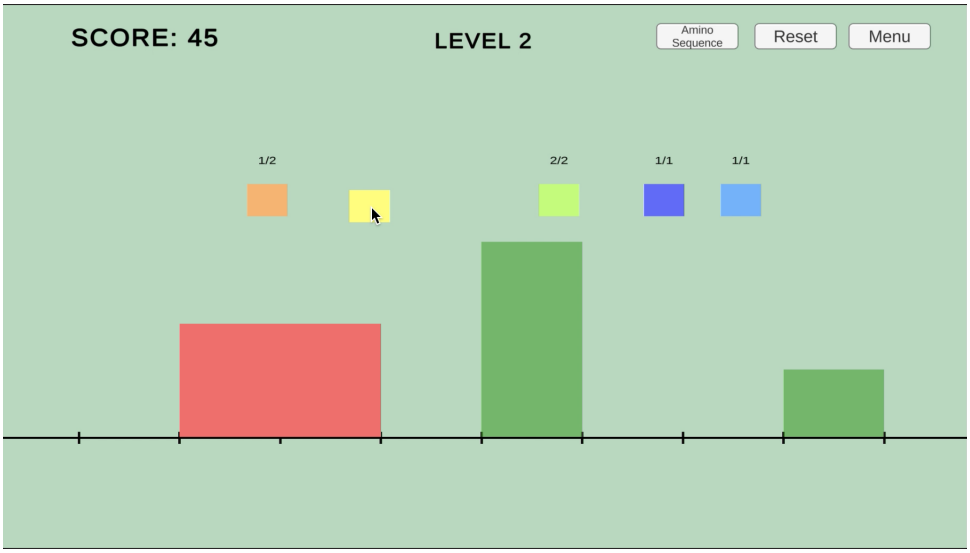

Figure 3.17: Level 2 with a different playing board than Level 1.

Chapter 4

Discussion

This chapter discusses the choices for implementing the game, including technology, crowd sourcing approach, development method, and the concept developed. The observations from the user testing are also mentioned, followed by our thoughts about solving de novo sequencing of MS/MS spectra with citizen science. In the end, we consider the addition of PTMs.

4.1 Choice of Game Technology

Before developing the project, we considered the frameworks React Native and Flutter and the game engine Unity. We chose not to look further at Flutter because it is very similar to React Native. We made basic prototypes in React Native and Unity before choosing which technology to use.

React Native is a framework for building native apps. Everything had to be coded with JavaScript, which was an unfamiliar programming language. It was difficult to get references to the game objects, making it hard to keep track of them. Since the game objects represent actual data from MS/MS spectra, is it essential to be able to refer back to the data the game objects represent.

The game engine Unity has a GUI builder that gives more control of the visual aspects. Unity uses the programming language C#, which was more familiar because of other languages I have used before. Since Unity uses GameObjects, where scripts and components can be attached, it was easier to have references to the game objects. This

made it easier to customize the different game objects, such as giving the different boxes different colors or changing their size when placed in a slot, while still knowing which amino acid the box represents. There are several tutorials on how to implement specific game features, which were a big advantage when using new technology.

The appearance and functionality are essential for the game to reach as many people as possible, and Unity makes it easy to change the game's appearance and functionality. Since we only had a few requirements before we started and expected to make changes along the way, it was also essential to use a development tool that allowed for flexibility. Based on our experience with making the basic prototypes in React Native and Unity, we are confident that Unity was the best choice of technology.

4.2 Crowdsourcing

In Section 1.6 we introduced the four categories of crowdsourcing: crowd creating, crowd rating, crowd processing, and crowd solving. The game we have can be categorized as a crowd solving game, i.e. where a crowd finds many different solutions to a given problem (Section 1.6). This fits the description of our game because each level does not have one correct solution but rather many possible solutions.

We think that crowd solving was the best crowdsourcing approach for making a game in the field of de novo sequencing of MS/MS spectra because we wanted to find new amino acid sequences. The other crowdsourcing approaches are mainly used to validate and classify data, this is however already covered by other software in proteomics, such as PeptideShaker (Section 3.1).

4.3 Concept Development

There are many different game genres. One distinction is whether a game is fast-paced or slow-paced. Tetris is an example of a fast-paced game where the players have limited time to place falling blocks to create a complete row. Foldit on the other hand is an example of a slow-paced game where the players work together or alone to find the best possible solution without a time limit. This allows the players to process the problem and to modify their solutions.

Both these options were considered when making the prototype. At the beginning of the development, we talked about the possibilities of having a Tetris-like game design where the amino acid boxes fall towards the ground, and the players have to place the boxes quickly on a matching slot. We concluded that this would not work if we later used the solutions from the game to compare to an algorithm because the players do not have time to find the best solution, only a quick solution.

The concept developed for the game was that the players have to minimize gaps and make a sequence based on the intensity of the peaks in the MS/MS spectrum. We wanted to motivate the players to find a good solution and give the user time to process before deciding where to place the boxes. The players should also be able to modify their solution to make it better. Therefore, we have chosen not to have a time limit in our game, as in Foldit.

4.4 Observations from User Testing

We used user testing to get opinions from the players about their thoughts and impressions of the application and to evaluate and find faults in the design of the prototype. The application was tested on a group of eight informatics students. The participants were both men and women in their twenties. The player performing the testing was given a laptop with the game, which they got to explore as we observed and took notes.

One thing we noticed was that most participants did not read the About page or the help texts before playing the game. This made some of the players confused about how to start playing. When they understood that they had to select a box to see the slots, they quickly understood that they had to place the box in a slot. The players placed boxes in slots until there were no more options and then seemed lost in what to do next. Some participants went back to the menu and tried Level 2, which they understood worked the same as the previous level. After playing around with the levels, some found the About page and read about the game.

The participants seemed to agree that the game works best as a mobile game and that it was clear how to place and remove boxes. Most of them want more hints before, during, and when there were no more moves. They thought it was interesting that they could read about the background of the game, and they saw the value of joining a citizen science project but agreed that it was not necessary to show the science in the game itself.

4.5 Algorithm vs. Citizen Science

When we started implementing the game, we discussed how the result from the game could be compared to the results from an algorithm. Based on the user testing we saw that the players had different approaches and results from each other.

We think that by solving de novo sequencing of MS/MS spectra with citizen science, the results would vary to a greater extent than solving it with one or more fixed algorithms. A fixed algorithm would always give the same answer because it has a fixed approach and always looks for the same solution. People can on the other hand use their creativity and will thus have different approaches to each other, and possibly different approaches to themselves, when playing the same level repeatedly or over time. Many game entries and enough people following the same approach when solving the game could potentially help make new or improve existing algorithms as they did in Foldit, as mentioned in Section 1.8.

4.6 Why Agile Development?

We followed an agile iterative development process when developing the game (Section 2.1). Each sprint started with a meeting where we planned which features we wanted to implement. The features were then planned and designed before being implemented. In our next meeting, we discussed if the feature needed additional changes. If so, it was developed further in the next sprint.

There are certain unique challenges developing software in science contra developing it for the industry sector. In the industry, requirements and specifications are usually defined at the start, before any implementation has been done. However, in scientific software development it is not always known what will work, which makes it difficult to have specific requirements before starting. We only had the main specifications of the game before we started implementing it, in addition to initial ideas for features of the game. For example, the original plan was that the box's width would be the same as the amino acid mass. After implementing this, we quickly noticed that the difference between each box was too small to distinguish them clearly, and thus had to find another solution to how to separate the boxes. Using an agile development method made it possible to be flexible when implementing new features and allowed us to find the design that worked best for the game.

Based on what we have learned, we would perhaps have used more time designing a low-fidelity prototype if we were doing this again. Low-fidelity prototyping is a quick and simple way of evolving an initial idea or concept, for example, on paper or a click-through prototype in an online designing tool [66], like Figma [67]. Using more time planning the design could have saved us development time and helped make more specific requirements early.

4.7 Considering PTMs When Doing De Novo Sequencing

In Section 1.1 we mentioned that amino acids with post-translational modifications (PTMs) exist in addition to the 20 standard amino acids. Since we developed a prototype, we only focused on the main functionality of the game and on figuring out which features worked and which that did not. We therefore decided to only use the standard amino acids and not including PTMs in the prototype.

PTMs can however be used for identifying specific peptides and proteins and, if implemented in the game, would give the players more options and possible result in sequences with higher scores. Since there exist over 1000 PTMs, the main challenge will be to decide which PTMs to add. In the current implementation of the prototype, brute force is used to decide which amino acids that can be used for a given game. This works because of the low number of amino acids. If PTMs were added to the game, an algorithm would have to be developed to decide which PTMs to add, as brute force would most likely be too time-consuming.

Another challenge with adding PTMs is the limited space on a mobile screen. Should there be a limit on how many PTMs that are added? If we were to use all the PTMs, most of the peaks would be left in the game and not filtered out. These would most likely give the player too many options and be rather confusing. But perhaps there are other ways of designing the game such that we can fit more boxes on the screen and still make it look nice? Additionally, would we maybe have to indicate whether a user places a modified amino acid or standard one? If so, how would the PTMs look different from the standard amino acids? These are just some of the many questions that have to be answered before PTMs can be added to the game.

Chapter 5

Conclusion

This thesis introduced the idea of using crowdsourcing to find new solutions to de novo sequencing of MS/MS spectra. Given that a group of people will have many individual and sometimes unique approaches at finding solutions we assumed that this could potentially provide different and hopefully novel solutions.

Through an agile development process where we discussed and evaluated each new feature, we developed a prototype of a game where the main concept is to find the amino acid sequence that uses as many amino acids as possible to get a high score.

Based on the implemented prototype we conclude that it is feasible to develop a citizen science game based on de novo sequencing of MS/MS spectra. The user testing also confirmed that people indeed have different approaches, and further large-scale testing has the potential to give new solutions to de novo sequencing of MS/MS spectra.

While the results of the prototype were promising, there are still unresolved challenges, such as how to include PTMs. To further develop the prototype, certain additional features can be implemented. For example, changing the design of the boxes and the background, and adding sound would automatically make it feel and look more game-like. More hints and tips on how to play the game and how to get a better score could also be added to better guide the players.

The next chapter goes more in-depth with regard to these and other ideas for future work.

Chapter 6

Future Work

During the development of the prototype, several ideas for additional features and better solutions for existing features were proposed. This chapter highlights a selection of these suggestions.

Better Use of the About Page

We saw that most players did not click the About page before playing the game in the user testing. The About page contains help on how to play the game and information about the background of the game. Naming the page "About" may be misleading, and it may be better to instead name the page "How to Play" or have two separate pages for About and How to Play.

A solution to this problem could be to make the About page into a pop-up window in front of the playing board, together with having a help button accessible during the game. Then the players would be "forced" to look at the page before playing the game. Another solution could be to have a "Level 0," made to show the players how to play the game and make the players follow a tutorial before being allowed to start playing the actual game by themselves.

Adding Additional Hints and Tips

Currently the players have to select a box and drop it before the information for that box is updated, i.e., the number of valid slots. In future versions, these numbers ought to be automatically updated.

The user testing also showed that the game should notify the players if they repeatedly make the same mistake. For example, one of the testers tried to put a box in an invalid slot several times without understanding why the box was not placed there. A notification in the form of text or sound could be added to prevent this from happening. This would be in addition to a Level 0.

Visualizing the Scores

We discovered that there should be more feedback on how the players score points and how to obtains the highest score. It should be clear that the players only get points for the height of the boxes and not the width, and that it is better to place more boxes than a few wide boxes. The easiest way to do this would be to show it in Level 0 and show the player that placing a box in a tall slot contra a wide slot results in a higher score.

To help visualize the scores for the different slots, the score when a box is dropped or removed could be displayed above each box. **Figure 6.1** shows an example where 20 points are added to the score when the purple box is placed and 15 points are subtracted from the score when the blue box is removed. Another option could be to show the scores of the slots when a box is selected and the matching slots are displayed (**Figure 6.2**).

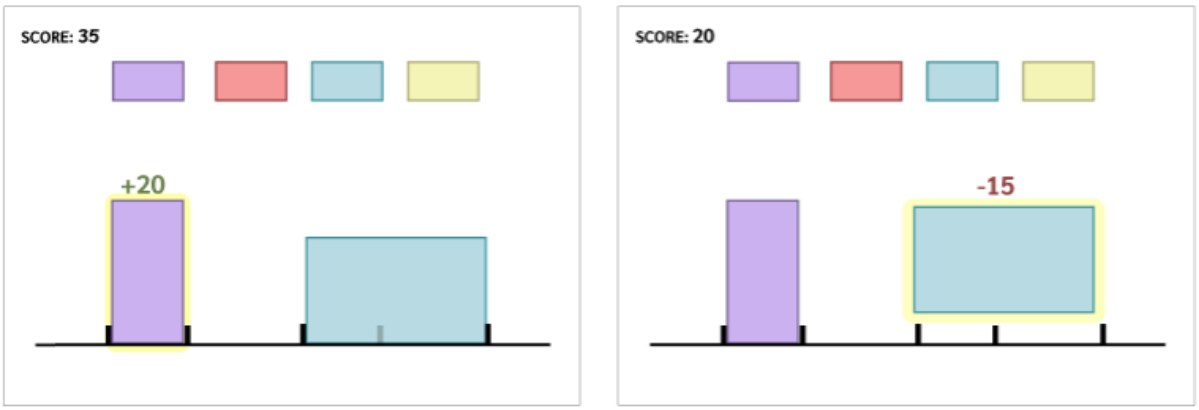

Figure 6.1: Shows how the points change when a box is added or removed.

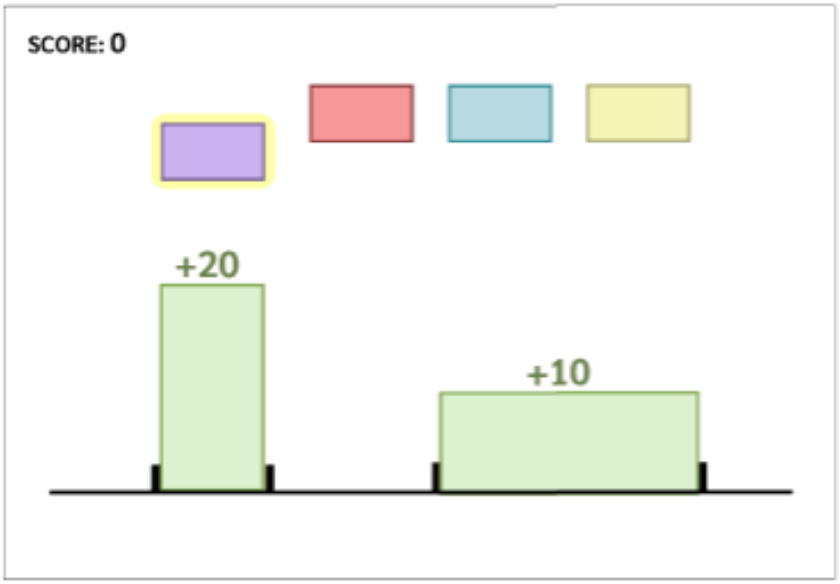

Figure 6.2: The points of the slots are highlighted when box is selected.

It is furthermore better to have longer continuous sequences without gaps. To motivate the players to do this, they could get extra points when placing boxes besides each other.

A message could also be displayed when there are no more possible moves, such that the players know that they can continue to the next level or remove boxes to change their sequence. In the user testing we got feedback that it could be an idea to show a player's high score for each level, thus motivating them to improve their score. Since the goal of the game is not necessary to get the highest score, a minimum score required to advance to the next level could also be considered.

Improvement of the Game Design

To test the concept of the idea, we did not focus on the graphical design of the prototype. However, to make the prototype look and feel even more like a game, and make the difference between the boxes even more significant, changing the texture, color, or shape of the boxes should be considered. This would also help elevate the game and make it look more exciting. This also applies for the other elements in the game, such as the peaks and the background. **Figure 6.3** shows an example of how the game design could be improved by giving the game a "city theme", where the boxes are changed to look like buildings.

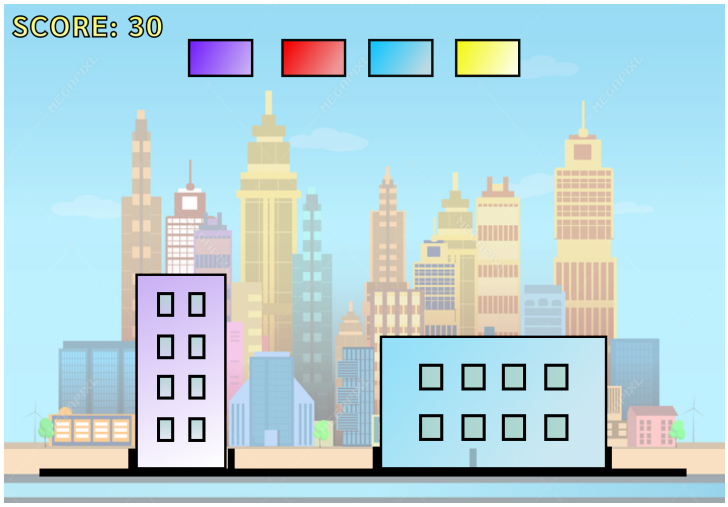

Figure 6.3: Example game design.

Large-scale Testing

The next step to test the game idea will be to test it on a larger scale. This includes adding more levels from MS/MS spectra, testing it on a bigger crowd and loading the spectrum files from online databases, such as PRIDE [68]. A high number of players and entries would produce data that could be compared to data from state-of-the-art software tools, such as PeptideShaker (Section 3.1). This would enable us to find out if the game really can contribute to identifying novel peptides and proteins, and if the results in turn can be used to improve algorithms for de novo sequencing of MS/MS spectra.

Bibliography

[1] "What is the Central Dogma?" [Online]. Available: https://www.yourgenome.org/facts/what-is-the-central-dogma

[2] L. Martens, "Lecture mass spectrometry basics - Part 2 of 7." [Online]. Available: https://www.youtube.com/watch?v=vXsotPtOdRY

[3] "Solve a puzzle and help genetics disease research," 2021, last accessed 8 December 2021. [Online]. Available: https://phylo.cs.mcgill.ca/play.php

[4] "File:project discovery p3 gold standard.png," June 2020, last accessed 2 December 2021. [Online]. Available: https://wiki.eveuniversity.org/File:Project_Discovery_P3_Gold_Standard.png

[5] "Why Choose Agile for Project Management," Dec. 2020. [Online]. Available: https://blog.ganttpro.com/en/why-agile/

[6] "The Amino Acid Masses." [Online]. Available: http://education.expasy.org/student_projects/isotopident/htdocs/aa-list.html

[7] "Molecular biology - Latest research and news | Nature." [Online]. Available: https://www.nature.com/subjects/molecular-biology

[8] E. Gråbøl-Undersrud, "proteinsyntese," Jun. 2021. [Online]. Available: http://snl.no/proteinsyntese

[9] "Central Dogma: Dna to RNA to protein," jul 19 2021, [Online; accessed 2021-12-09].

[10] P. Kierulf, "aminosyrer," Aug. 2021. [Online]. Available: http://snl.no/aminosyrer

[11] "peptider." [Online]. Available: http://sml.snl.no/peptider

[12] E. Gråbøl-Undersrud, "proteiner," http://snl.no/proteiner, Nov. 2021.

[13] M. Mann and O. N. Jensen, "Proteomic analysis of post-translational modifications," *Nature Biotechnology*, vol. 21, no. 3, pp. 255–261, Mar. 2003. [Online]. Available: https://www.nature.com/articles/nbt0303-255

[14] S. Degroeve, R. Gabriels, K. Velghe, R. Bouwmeester, N. Tichshenko, and L. Martens, "ionbot: a novel, innovative and sensitive machine learning approach to lc-ms/ms peptide identification," 2021.

[15] M. Vailati-Riboni, V. Palombo, and J. J. Loor, *What Are Omics Sciences?* Cham: Springer International Publishing, 2017, pp. 1–7. [Online]. Available: https://doi.org/10.1007/978-3-319-43033-1_1

[16] "Genetics vs. Genomics Fact Sheet." [Online]. Available: https://www.genome.gov/about-genomics/fact-sheets/Genetics-vs-Genomics

[17] "Transcriptomics - Latest research and news | Nature." [Online]. Available: https://www.nature.com/subjects/transcriptomics

[18] E. Milward, A. Shahandeh, M. Heidari, D. Johnstone, N. Daneshi, and H. Hondermarck, "Transcriptomics," in *Encyclopedia of Cell Biology*, R. A. Bradshaw and P. D. Stahl, Eds. Waltham: Academic Press, 2016, pp. 160–165. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123944474400295

[19] EMBL-EBI, "What is proteomics? | Proteomics." [Online]. Available: https://www.ebi.ac.uk/training/online/courses/proteomics-an-introduction/what-is-proteomics/

[20] M. Tyers and M. Mann, "From genomics to proteomics," *Nature*, vol. 422, no. 6928, pp. 193–197, 2003.

[21] J. Pevsner, *Bioinformatics and functional genomics*. John Wiley & Sons, 2015.

[22] I. Eidhammer, K. Flikka, L. Martens, and S.-O. Mikalsen, *Computational methods for mass spectrometry proteomics*. John Wiley & Sons, 2008.

[23] K. Verheggen, H. Ræder, F. S. Berven, L. Martens, H. Barsnes, and M. Vaudel, "Anatomy and evolution of database search engines—a central component of mass spectrometry based proteomic workflows," *Mass spectrometry reviews*, vol. 39, no. 3, pp. 292–306, 2020.

[24] J. Griss, "Spectral library searching in proteomics," *Proteomics*, vol. 16, no. 5, pp. 729–740, 2016.

[25] K. G. Standing, "Peptide and protein de novo sequencing by mass spectrometry," *Current Opinion in Structural Biology*, vol. 13, no. 5, pp. 595–601, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959440X03001386

[26] T. Muth and B. Y. Renard, "Evaluating de novo sequencing in proteomics: already an accurate alternative to database-driven peptide identification?" *Briefings in Bioinformatics*, vol. 19, no. 5, pp. 954–970, 03 2017. [Online]. Available: https://doi.org/10.1093/bib/bbx033

[27] B. Ma, "Novor: Real-time peptide de novo sequencing software," *Journal of the American Society for Mass Spectrometry*, vol. 26, no. 11, pp. 1885–1894, 2015, pMID: 26122521. [Online]. Available: https://doi.org/10.1007/s13361-015-1204-0

[28] "Nasa - flagship initiatives," last accessed 6 December 2021. [Online]. Available: https://www.nasa.gov/open/plan/peo.html

[29] "The story so far," last accessed 3 December 2021. [Online]. Available: https://www.zooniverse.org/projects/zookeeper/galaxy-zoo/about/results

[30] "Into The Brain." [Online]. Available: https://science.eyewire.org/

[31] B. Morschheuser, J. Hamari, and J. Koivisto, "Gamification in crowdsourcing: A review," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 2016, pp. 4375–4384.

[32] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: Defining "gamification"," in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, ser. MindTrek '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 9–15. [Online]. Available: https://doi.org/10.1145/2181037.2181040

[33] I. Blohm and J. M. Leimeister, "Gamification," *Business & information systems engineering*, vol. 5, no. 4, pp. 275–278, 2013.

[34] D. Dicheva, C. Dichev, G. Agre, and G. Angelova, "Gamification in education: A systematic mapping study," *Journal of Educational Technology & Society*, vol. 18, no. 3, pp. 75–88, 2015.

[35] "The science behind foldit," https://fold.it/portal/info/about, last accessed 27 February 2021.

[36] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, Aug. 2021. [Online]. Available: https://www.nature.com/articles/s41586-021-03819-2

[37] V. Curtis, "Motivation to participate in an online citizen science game: A study of foldit," *Science Communication*, vol. 37, no. 6, pp. 723–746, 2015.

[38] F. Khatib, S. Cooper, M. D. Tyka, K. Xu, I. Makedon, Z. Popović, and D. Baker, "Algorithm discovery by protein folding game players," *Proceedings of the National Academy of Sciences*, vol. 108, no. 47, pp. 18 949–18 953, 2011.

[39] "Eterna." [Online]. Available: https://eternagame.org/

[40] J. O. L. Andreasson, M. R. Gotrik, M. J. Wu, H. K. Wayment-Steele, W. Kladwang, F. Portela, R. Wellington-Oguri, null null, R. Das, and W. J. Greenleaf, "Crowdsourced rna design discovers diverse, reversible, efficient, self-contained molecular switches," *Proceedings of the National Academy of Sciences*, vol. 119, no. 18, p. e2112979119, 2022. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.2112979119

[41] "Polygenic Inheritance and Gene Mapping | Learn Science at Scitable." [Online]. Available: http://www.nature.com/scitable/topicpage/polygenic-inheritance-and-gene-mapping-915

[42] "UCSC Genome Browser Home." [Online]. Available: https://genome.ucsc.edu/

[43] A. Kawrykow, G. Roumanis, A. Kam, D. Kwak, C. Leung, C. Wu, E. Zarour, P. players, L. Sarmenta, M. Blanchette, and J. Waldispühl, "Phylo: A citizen science approach for improving multiple sequence alignment," *PLOS ONE*, vol. 7, no. 3, pp. 1–9, 03 2012. [Online]. Available: https://doi.org/10.1371/journal.pone.0031362

[44] "Phylo: A Citizen Science Approach for Improving Multiple Sequence Alignment," vol. 7.

[45] "What is eve online?" https://www.eveonline.com/, 2021, last accessed 30 November 2021.

[46] "Project discovery nominated for webby award," https://www.eveonline.com/news/view/project-discovery-nominated-for-webby-award, May 2021, last accessed 30 November 2021.

[47] "Project discovery: Flow cytometry," September 2021, last accessed 2 December 2021. [Online]. Available: https://wiki.eveuniversity.org/Project_Discovery:_Flow_Cytometry

[48] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: Review and analysis," *arXiv preprint arXiv:1709.08439*, 2017.

[49] "What is User testing? Definition." [Online]. Available: https://www.omniconvert.com/what-is/user-testing/

[50] "Mobile Vs Desktop: Which Platform Is Better For Gaming?" Jul. 2019. [Online]. Available: https://www.fortressofsolitude.co.za/mobile-vs-desktop-which-platform-is-better-for-gaming/

[51] "React Native · Learn once, write anywhere." [Online]. Available: https://reactnative.dev/

[52] "Components and Props – React." [Online]. Available: https://reactjs.org/docs/components-and-props.html

[53] U. Technologies, "Unity Real-Time Development Platform | 3D, 2D VR & AR Engine." [Online]. Available: https://unity.com/

[54] "List of Unity games," Feb. 2022, page Version ID: 1073950223. [Online]. Available: https://en.wikipedia.org/w/index.php?title=List_of_Unity_games&oldid=1073950223

[55] J. Cecil, P. Ramanathan, M. Pirela-Cruz, and M. B. R. Kumar, "A Virtual Reality Based Simulation Environment for Orthopedic Surgery," in *On the Move to Meaningful Internet Systems: OTM 2014 Workshops*, ser. Lecture Notes in Computer Science, R. Meersman, H. Panetto, A. Mishra, R. Valencia-García, A. L. Soares, I. Ciuciu, F. Ferri, G. Weichhart, T. Moser, M. Bezzi, and H. Chan, Eds. Berlin, Heidelberg: Springer, 2014, pp. 275–285.

[56] "Mascot database search | Data file format for mass spectrometry peak lists." [Online]. Available: http://www.matrixscience.com/help/data_file_help.html

[57] "JSON." [Online]. Available: https://www.json.org/json-en.html

[58] "Build software better, together." [Online]. Available: https://github.com

[59] M. Vaudel, J. M. Burkhart, R. P. Zahedi, E. Oveland, F. S. Berven, A. Sickmann, L. Martens, and H. Barsnes, "PeptideShaker enables reanalysis of MS-derived proteomics data sets," *Nature Biotechnology*, vol. 33, no. 1, pp. 22–24, Jan. 2015. [Online]. Available: https://www.nature.com/articles/nbt.3109

[60] "BlueStacks – Best Mobile Gaming Platform for PC & Mac | 100% Safe and FREE." [Online]. Available: https://www.bluestacks.com

[61] "Download Android Studio and SDK tools." [Online]. Available: https://developer.android.com/studio

[62] U. Technologies, "Unity - Manual: The Hierarchy window." [Online]. Available: https://docs.unity3d.com/Manual/Hierarchy.html

[63] "Unity - Scripting API: EventSystem." [Online]. Available: https://docs.unity3d.com/2018.2/Documentation/ScriptReference/EventSystems.EventSystem.html

[64] "Unity - Manual: Prefabs." [Online]. Available: https://docs.unity3d.com/Manual/Prefabs.html

[65] "Unity - Manual: Sprites." [Online]. Available: https://docs.unity3d.com/Manual/Sprites.html

[66] F. Tan, "The Differences Between Low Fidelity vs. High Fidelity Prototyping." [Online]. Available: https://www.protopie.io/blog/low-fidelity-vs-high-fidelity-prototyping

[67] "Figma: the collaborative interface design tool." [Online]. Available: https://www.figma.com/

[68] Y. Perez-Riverol, J. Bai, C. Bandla, D. García-Seisdedos, S. Hewapathirana, S. Kamatchinathan, D. J. Kundu, A. Prakash, A. Frericks-Zipper, M. Eisenacher *et al.*, "The pride database resources in 2022: a hub for mass spectrometry-based proteomics evidences," *Nucleic acids research*, vol. 50, no. D1, pp. D543–D552, 2022.