Aalto-yliopisto
Aalto-universitetet
Aalto University

Master's Programme in Chemical, Biochemical and Materials Engineering

Major in Biotechnology

InGENEous

Teaching Synthetic Biology through a Video Game

Aarni Aspi

Supervisor: Juho Rousu, Professor at Aalto-University

Advisor: Merja Penttilä, Adjunct Professor (Emerita) at Aalto-University,

Senior Advisor at VTT

Master's thesis
2025

Aalto-yliopisto
Aalto-universitetet
Aalto University

Author Aarni Aspi

Title of thesis InGENEous: Teaching Synthetic Biology through a Video Game

Programme Chemical, Biochemical and Materials Engineering

Major Biotechnology

Thesis supervisor Juho Rousu, Professor at Aalto-University

Thesis advisor Merja Penttilä, Adjunct Professor (Emerita) at Aalto-University, Senior Advisor at VTT

Collaborative partner VTT

Date 13.02.2025 **Number of pages** 89 + 16 **Language** English

Abstract

Synthetic biology is a rapidly evolving interdisciplinary field that combines principles of biology, engineering and computer science to address global challenges sustainably in various fields such as agriculture or biomedicine. This is accomplished by using engineered DNA sequences to genetically modify organisms with advanced functionalities. Synthetic biology is often demonstrated and taught using BioBricks - standardized DNA sequences that simplify the subject to a broader audience.

The purposeful use of video games in science, research and education has grown in popularity due to their accessible, engaging, and feedback-driven nature, which enables players to enter a highly focused and motivated flow state. While video games related to synthetic biology have so far provided in-depth knowledge of specific aspects of the field, none have yet provided the player with a comprehensive understanding of the field as whole.

InGENEous, an educational game on synthetic biology, aims to provide a holistic understanding of the field to its players, regardless of their background. The players must solve a variety of real-world problems by designing, building, testing, and learning with the help of BioBricks. In doing so, the game fosters critical thinking, creativity, and an understanding of the interdisciplinary nature of synthetic biology.

InGENEous also holds promise as a versatile software tool for research, enabling the players to export their designs and data from the game accurately. This feature facilitates a wide range of research applications and supports crowd-sourced citizen science projects.

Keywords Synthetic biology, BioBricks, iGEM, video game, citizen science, edutainment, biological logic gates, educational game, design tool

Aalto-yliopisto
Aalto-universitetet
Aalto University

Tekijä Aarni Aspi

Työn nimi InGENEous: Synteettisen biologian opettaminen pelin avulla

Koulutusohjelma Chemical, Biochemical and Materials Engineering

Pääaine Biotechnology

Vastuuopettaja/Valvoja Juho Rousu, Professor at Aalto-University

Työn ohjaaja Merja Penttilä, Adjunct Professor (Emerita) at Aalto-University, Senior Advisor at VTT

Yhteistyötaho VTT

Päivämäärä 13.02.2025 **Sivumäärä** 89 + 16 **Kieli** Englanti

Tiivistelmä

Synteettinen biologia on nopeasti kehittyvä poikkitieteellinen ala, joka yhdistää biologian, tekniikan ja tietojenkäsittelytieteen menetelmiä globaalien haasteiden kestävään ratkaisemiseen eri aloilla, kuten maataloudessa tai biolääketieteessä. Tähän pyritään käyttämällä erityisesti suunniteltuja DNA-sekvenssejä organismien geneettiseen muokkaukseen, mahdollistaen edistyneitä biologisia toiminnallisuuksia.

Videopelien käyttö tieteessä, tutkimuksessa ja koulutuksessa on kasvattanut suosiotaan niiden saavutettavan, mukaansatempaavan ja palautelähtöisen luonteen ansiosta, joka mahdollistaa erittäin keskittyneen ja motivoituneen flow-tilan pelaajissa. Vaikka synteettiseen biologiaan liittyvät videopelit ovat tähän mennessä syventäneet ymmärrystä tietyistä aihealueista, yksikään niistä ei ole vielä tarjonnut pelaajalle kokonaisvaltaista käsitystä synteettisestä biologiasta kokonaisuutena.

InGENEous on synteettiseen biologiaan keskittyvä opetuspeli, jonka tavoitteena on tarjota pelaajilleen kokonaisvaltainen käsitys synteettisestä biologiasta heidän taustastaan riippumatta. Pelissä pelaajien on ratkaistava erilaisia todellisia ongelmia suunnittelemalla, rakentamalla, testaamalla ja oppimalla BioBrick:ien avulla. Peli edistää kriittistä ajattelua, luovuutta, sekä synteettisen biologian poikkitieteellisen luonteen ymmärystä pelaajissaan.

InGENEous voisi myös mahdollisesti toimia tieteellisenä työkaluna, sillä pelaajien rakentamat biologiset laitteet ja niiden sisältämä tieteellinen data voidaan viedä ulos pelistä. Tämä ominaisuus tukee monenlaisia tutkimussovellutuksia, sekä mahdollistaa kansalaistiedeprojekteja.

Avainsanat synteettinen biologia, BioBricks, iGEM, videopeli, kansalaistiede, biologiset loogiset portit, pelillistäminen, opetuspeli, suunnittelutyökalu

Aalto-yliopisto
Aalto-universitetet
Aalto University

Table of contents

Preface and acknowledgements

"Humanity is a product of billions of years of deterministic evolution,
Yet we think ourselves or divinity to be the hero of this revolution,
Life and all its wonders are truly a humbling sight to behold,
From a vividly colorful parrot to the tiny, tiny water mold,
Across life's complex spectrum, secrets we must uncover,
For we are losing species at an alarming rate, what a bother,
To understand ourselves, with all life we must discuss,
For whether we like it or not, we are part of nature, and nature is within us."

SynBiotic Reflections
A poem by Aarni Aspi

When the work on this thesis began, Adjunct Professor Merja Penttilä acted as my supervisor from the School of Chemical Engineering in Aalto-University, and Professor Juho Rousu acted as my advisor from the School of Science in Aalto-University. Merja Penttilä retired from Aalto-University during the writing of this thesis, after which Juho Rousu took on the role of the supervisor, and Merja Penttilä took on the role of the advisor.

I want to thank my supervisor Professor Juho Rousu, and my advisor Adjunct Professor (Emerita) Merja Penttilä, for their patience, guidance and support. I would also like to thank The Technical Research Centre of Finland (VTT), Aalto-University, The Center for Young SynBio Scientists (CYSS) and International Genetically Engineered Machine foundation (iGEM) for providing information and resources. A special thanks to the Jenny and Antti Wihuri foundation for providing the funding for this thesis and the development of the test version of the game through CYSS.

I would also like to extend my gratitude to the people who consulted me and produced art, music and programming for InGENEous based on my designs and guidance. These people are credited below:

- Toni Saarela: Game Engine Programming
- Zsófia Kecskés: Character Art and Introduction Art
- Alli Loikkanen: Background Art
- Julia Rässa: Icons and UI Elements
- Tuomas Hellsten: UI Elements and UI Art
- Lukas Baldvinsson: Character Music
- Monika Tuominen: Introduction Art
- Jakub Dobi: DNA Sequence View Programming
- Juho Pirtinoja: Unity and JSON Consulting

It has been an absolute privilege to work with talented individuals to make the educational material of InGENEous presented with such high-quality assets.

I would also like to thank my partner Julia Rässa, my family and friends for their undying support, feedback, ideas and for keeping me sane while writing this thesis.

To everyone else who has been there with me along the way and feels like they didn't receive credit yet: you know who you are and thank you from the bottom of my heart.

Espoo, 13th of February 2025
Aarni Aspi

Abbreviations

3D	Three-Dimensional
AI	Artificial Intelligence
CDS	Coding Sequence
CSV-file	Comma-separated values text file
DBTL	Design-Build-Test-Learn cycle
DNA	Deoxyribonucleic Acid
GMO	Genetically modified organism
GWAPs	Games With a Purpose
HTML5	Hypertext Markup Language 5
IoT	Internet of Things
IRES	Internal ribosome entry site
JSON-file	JavaScript Object Notation file format
Met-tRNA	Methionine Transfer RNA
MMORPG	Massively Multiplayer Online Role-Playing Game
mRNA	Messenger Ribonucleic Acid
PCR	Polymerase chain reaction
Poly(A)	Polyadenylation
Pre-mRNA	Primary RNA Transcript
RBS	Ribosome Binding Site
RNA	Ribonucleic Acid
rRNA	Ribosomal RNA
SBOL	Synthetic Biology Open Language
SynBio	Synthetic Biology
tRNA	Transfer RNA
UI	User Interface
UX	User Experience
VG	Video Game

1 Introduction & Literature Review

This chapter contains a literature review for synthetic biology (SynBio), as well as a literature review on games and their use in scientific research and education. Covering these topics is essential to help the reader in understanding the terminology and mechanisms discussed in later chapters and the appendices, as well as to answer the first set of research questions:

- What is SynBio and what kind of approaches are used in the field?
- What are BioBricks and how are they used?
- What are games and how do they function?
- What is the flow state, and how is it achieved?
- How are games used in scientific research and education?

This chapter also includes a section that addresses the need for an educational game about SynBio, the aim of this thesis and a second set of research questions related with the development of the game itself (p. 27).

1.1 Synthetic Biology

SynBio is a rapidly evolving interdisciplinary subfield of genetic engineering, that integrates biology, engineering, and computer science. While traditional genetic engineering focuses on modifying the cellular functions of existing organisms, SynBio aims to create entirely new functionalities, introduce existing functionalities from one species to another, or redesign natural systems for specific purposes (Purnick and Weiss, 2009). At its core, SynBio revolves around the concept of engineering novel biological systems not yet found in nature via combining various genes (Hanczyc, 2020). By leveraging the principles of genetic engineering, systems biology and molecular biology, SynBio has provided further insights into how cells and biological systems function, leading to significant contributions to the broader field of biotechnology, and biosciences as a whole.

The following Figure 1. illustrates technologies and potential applications of SynBio using plants as chassis, however these examples are applicable to various other types of chassis as well. SynBio has been mostly focused on using microbial chassis for modification, since their well-researched and simple genetic structure enables rapid development cycles, and a higher degree of success in comparison to multicellular chassis, such as animals or plants.

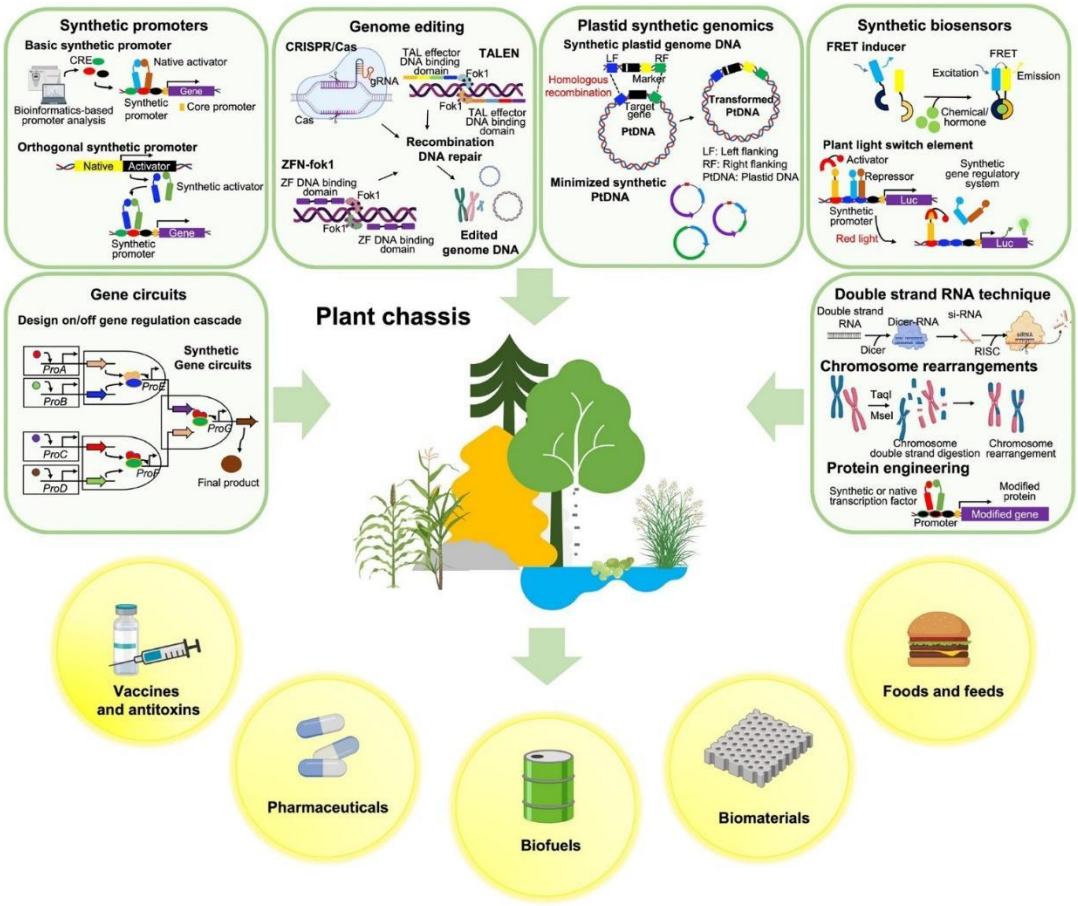

Figure 1. Technologies and potential applications of SynBio (Yang et al., 2022).

Various methods are applied to achieve the goals of SynBio, such as the design of libraries, databases and repositories for genetic information, such as genes, microbial strains and genetic components with specific functions, such as promoters, ribosome binding sites (RBSs), coding sequences (CDSs) and terminators. SynBio also involves the assembly of these genetic parts together to create genetic devices, circuits, logic gates, and systems, as well as optimizing the whole genomes of organisms. Computer models that predict the behavior of synthetic biological systems are also integral to SynBio and are designed using data acquired through laboratory testing. (Cameron et al., 2014)

The utilization of synthetic biology in biotechnology has also advanced the field of metabolic engineering, which can be viewed as a parallel field to SynBio or a subfield of SynBio (García-Granados et al., 2019). In metabolic engineering, entire metabolic pathways are scrutinized and reprogrammed to optimize the production of desired compounds, such as specialty chemicals,

pharmaceuticals, or biofuels. This ability to redesign metabolism at the genetic level is transforming industrial biotechnology with efficient and sustainable production processes that often involve waste streams, such as agricultural, textile and food waste as feedstock (Wang et al., 2024).

Living cells are essentially a collection of different biochemical reactions that provide the cell with a variety of useful functions that perpetuate the life of the cell and the conservation of its genetic information in DNA. Identifying and understanding the genes responsible for these functions allows the construction of a "virtual toolbox" of advanced genetic functions. Traditional approaches consider biological functions to be standard pathways of biology, and their purpose and interactions are studied extensively in the field of molecular biology. SynBio, however, investigates each cellular function through the lens of how they could be modified, replaced or removed for specific applications. This detailed exploration allows for the creation of new, synthetic parts tailored for unique functions beyond what is found in nature. At its core. SynBio focuses on developing tools by isolating proteins, ribonucleic acids (RNAs) or deoxyribonucleic acids (DNAs) from their original sources and engineering them into different cellular environments for innovative biotechnological solutions (Ausländer et al., 2017).

1.1.1 Understanding Gene Expression

Gene expression is the process in which the information in a coding DNA sequence, a gene, is processed, allowing a cell to synthesize various biomolecules. By manipulating gene expression, SynBio is used to engineer cells with advanced behaviors. (Wang et al., 2013)

Gene expression is an integral part of the central dogma of molecular biology (Schneider-Poetsch and Yoshida, 2018). It revolves around two consecutive biological processes: transcription, in which a sequence of messenger RNA (mRNA) is synthesized based on a DNA sequence, and translation, in which a protein is built based on the genetic information stored in the transcribed RNA sequence (Guo, 2014).

RNA has a similar linear structure to DNA, consisting of four different types of nucleotides linked with phosphodiester bonds. Unlike double-stranded DNA however, RNA is a single-stranded molecule containing a sugar ribose, and the bases adenine (A), guanine (G), cytosine (C), and uracil (U) instead of the thymine (T) found in DNA (Alberts et al., 2017). RNA can be divided into four general types, with mRNA being the specific type of RNA produced during transcription.

Transcription is catalyzed by RNA polymerase, an enzyme which consists of several subunits each responsible for different tasks: one identifies the correct site for the transcription complex, another synthesizes the single-stranded RNA molecule, and yet another one for detaches the mRNA from the DNA template upon synthesis. Transcription begins at the gene's promoter sequence, where transcription factors attach to the upstream (5´) end of the gene. By recruiting RNA polymerase, these transcription factors initiate the synthesis of the RNA from the gene's coding sequence, reading the DNA template from 5´ end to 3´ end directions until they reach a termination sequence for the DNA template, which signals the end of the transcription. The resulting RNA sequence is complementary to the template strand, except for uracil bases replacing the thymine bases from DNA.

Once the primary RNA transcript (pre-mRNA) is synthesized, it undergoes modifications for stability and efficiency, which are required for optimal translation. In eukaryotes, pre-mRNA contains both coding (exon) and non-coding (intron) regions, and specific enzymes remove introns and splice exon regions, forming the mature mRNA. Another modification is the addition of a 7-methylguanosine cap to the 5´ end and a poly(A) tail, consisting of 80 to 250 adenine nucleotides are applied to the 3´ end. The attachment of the poly(A) tail, known as polyadenylation, is found in almost all species, though exceptions exist (Slomovic et al., 2008). These modifications stabilize the mRNA before the mature mRNA is fully modified and ready for translation in the cytoplasm. In prokaryotes however, these additional modifications are not required, as prokaryotic DNA does not contain non-coding regions.

Translation occurs in ribosomes, functional protein complexes found in all species. Ribosomes are composed of ribosomal RNA (rRNA) and proteins. when mRNA binds to a ribosome, the mRNA sequence is read in units of three bases called codons. Each codon represents a specific amino acid or serves a regulatory function by initiating or terminating protein synthesis. Translation begins when the small ribosomal subunit binds to the mRNA at the 5' cap site and locates the start codon, AUG. The large ribosomal subunit then binds with the mRNA, and a charged transfer RNA (tRNA) carrying methionine, methionine transfer RNA (Met-tRNA), initiates protein synthesis. As the ribosome moves along the mRNA in the 5′–3′ direction, it facilitates the binding of tRNAs to their corresponding codons, linking amino acids together with peptide bonds. This process of translating genetic information from mRNA into a specific chain of amino acids, forming a protein, is integral to the central dogma of molecular biology, which is the flow of genetic information from DNA to RNA to protein (Koellner et al., 2018). Transcription and translation are illustrated in Figure 2.

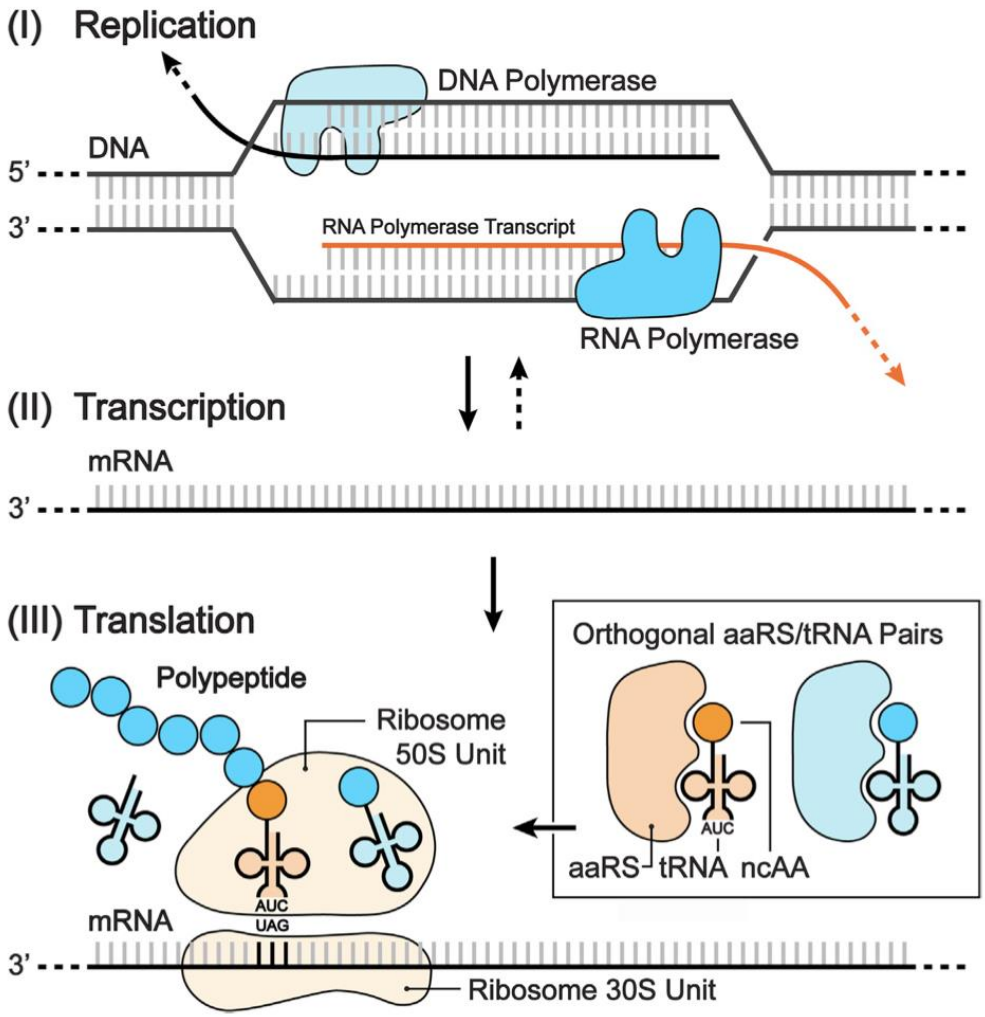

Figure 2. The pathway of cellular expression (Diercks et al., 2021).

This foundational knowledge on gene expression, along with an understanding of the principles of genetic engineering and molecular biology, enables the design of new genetic constructs in the field of SynBio.

1.1.2 BioBricks

To simplify modifying and building functionalities into organisms using DNA sequences, Tom Knight and his research team at MIT introduced the concept of BioBricks — standardized DNA sequences that adhere to specific restriction-enzyme based assembly standards (Knight, 2003). BioBricks have defined functions and combining them correctly allows target cells to be

reprogrammed to perform specific functions, demonstrating their practical application in engineering biology (Røkke et al., 2014).

BioBricks are especially useful in demonstrating and teaching SynBio, since BioBricks are categorized into various types by function, and each type has a visual glyph associated with them. This abstraction of complex mechanisms into visual iconography allows learning the subject matter in a simpler manner. This simplification also helps in the design phase of development, since the glyphs are quite useful for creating visual design diagrams. An example of a diagram with BioBricks and genetic devices is shown in Figure 3.

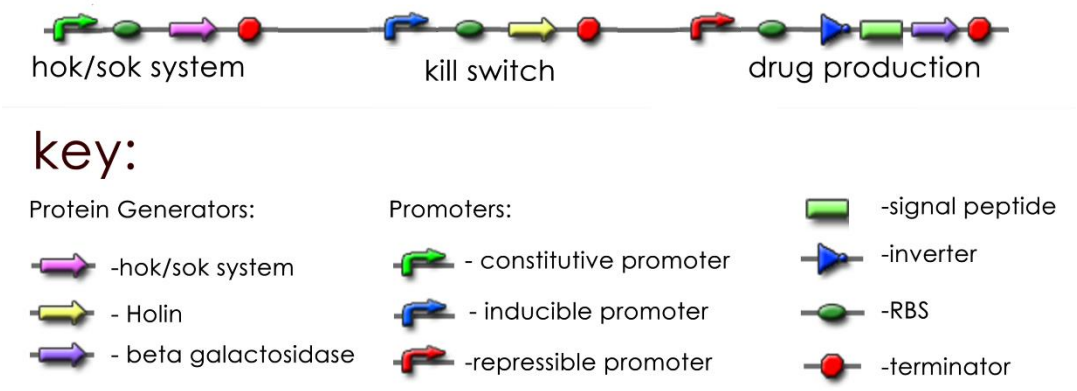

Figure 3. A diagram of biological devices built with BioBricks (iGEM - Team VIT Vellore, 2011).

Promoters are sequences responsible for controlling gene expression by enabling RNA polymerase to bind to DNA. Several types of promoters exist, such as constitutive promoters that continuously express the downstream sequence, or inducible and repressible promoters that are either activated or deactivated by specific transcription factors. (Cazier and Blazeck, 2021)

For example, the functionality of the pBAD promoter is repressed when the sugar glucose is present in the extracellular environment. When the cell has access to arabinose in the absence of glucose, the CDS downstream from the promoter is transcribed into mRNA and subsequently translated into a protein. (Khlebnikov et al., 2000)

Regulatory promoters such as the pBAD promoter allow the creation of biological “on-off switches”, leading to the possibility of creating biological logic gates, circuits and systems that operate similarly to their counterparts in

electrical engineering. Therefore, promoters can be considered as the sequences responsible for controlling the input of a genetic device. (Singh, 2014)

RBSs are DNA sequences which allow the mRNA transcript to bind with ribosomes and are located upstream of the coding sequence. The strength of an RBS determines the efficiency of this binding, and different RBSs have varying levels of strength. Therefore, the choice of which RBS to use enables control over the expression rate of the CDS. RBSs can be used to regulate the level of expression as needed, either to maximize production or to decrease production. (Salis et al., 2009)

CDSs are simply the regions of coding DNA that are transcribed during transcription. These parts essentially serve as blueprints for transcription, and the DNA sequence of CDSs therefore determines the amino acid sequence of the proteins assembled during translation. (Ausländer et al., 2017)

Finally, terminators are sequences that signal RNA polymerase to detach from the DNA, ending transcription. Like RBSs, different terminators have varying strengths: strong terminators cause RNA polymerase to unbind efficiently, while weak terminators are less efficient. The choice of terminator thus provides another avenue for direct control over gene expression. (Zhai et al., 2022)

A plethora of other types of BioBricks have also been characterized. These include parts that assist in cloning and replication, such as primer binding sites, origin of replication sites (ORI), origin of transfer sites (ORIT), and selection marker sites. Some parts are designed to optimize the DNA structure itself, either to assist expression or to hold specific molecules, such as aptamers, DNA origami, or insulators (Dong et al., 2020).

Other useful DNA structures that are often denoted with glyphs, include sequences that assist the assembly and ligation of DNA, such as sticky and blunt restriction sites, and overhangs. These are typically part of the BioBricks themselves though and could be considered as features of a BioBrick, instead of individual BioBricks themselves.

Once a design is ready, BioBricks are introduced to a target organism using a variety of methods. For microbes, this is commonly done via circular DNA structures known as plasmids. Plasmids can either be custom-built or acquired as standardized plasmid backbones, and they usually contain a selection marker site, typically an antibiotic-resistant gene, which allows for the separation of successfully transformed cells from unsuccessful ones. Another common feature of plasmids is the origin of replication (ORI) site, which

enables the plasmid to replicate during cell division, thus ensuring the function of the plasmid is perpetuated in subsequent generations of the cell. (Dolan and Matilla, 2022)

The target organism, known as the chassis, must also be chosen carefully, as some BioBricks work only in eukaryotes or prokaryotes. The genetic construct built with BioBricks will also impose a burden on the metabolism of the chassis, as new functions require energy, which may weaken existing metabolic pathways of the cell. This burden can be adjusted by either optimizing the genome of the chassis itself or by choosing a more suitable chassis. (Liu et al., 2019)

The standardization of BioBricks is driven by an international community of synthetic biologists, supported by open-source tools and data. This collaborative effort is largely catalyzed by both researchers and initiatives such as iGEM, an international SynBio competition, which promotes the development and sharing of standardized BioBricks. This enables the creation of complex biological systems in a modular and predictable manner (Goujet, 2018).

1.1.3 iGEM

The International Genetically Engineered Machine (iGEM) competition is an event where high school, undergraduate and overgraduate teams develop practical synthetic biology projects using various BioBricks. One of the key points of focus in iGEM is the development and standardization of BioBricks, and the competition encourages researchers and teams to submit their BioBricks to make them available to the global SynBio community through the iGEM Parts repository. As iGEM has expanded its global recognition, the number of BioBricks submitted into the repository has also increased steadily, along with the development of advanced DNA assembly techniques, such as Golden Gate Cloning and the Nobel-prize winning CRISPR/Cas9-system. (iGEM, 2024a)

These advances, along with the more recent advances in generative artificial intelligence (AI) technologies, have significantly enhanced the capabilities of iGEM teams in creating a diverse variety of genetic systems. The increased use of these techniques has not only improved the quality and scope of projects submitted to iGEM but has also enabled more and more iGEM alumni to secure funding based on their projects and form SynBio related start-ups. (iGEM, 2024b)

The integration of these sophisticated methods into iGEM projects exemplifies the competition's role in pushing the boundaries of scientific research and technology. The ever-growing repository of BioBricks in the iGEM Parts repository reflects the collaborative nature of the competition, as teams contribute their unique DNA parts to a shared database. This collaborative effort ensures that valuable genetic tools are freely accessible to the global synthetic biology community, fostering further research and development. (Kelwick et al., 2015)

An overview of some award-winning examples utilizing BioBricks from iGEM 2024 are detailed below, to illustrate both the importance of the iGEM competition, and to provide an understanding on the variety of applications enabled using BioBricks.

Tarakate - From Rainforest Deforestation to Sustainable Rubber: Using Plant SynBio to Facilitate Dandelion Engineering

The project of the iGEM team Marburg was to engineer 37 novel regulatory promoter BioBricks to enable the use of dandelions as a SynBio chassis. The aim of this project was to allow future researchers to make use of these promoters in a wide variety of plant-based applications, ranging from sustainable rubber production to helping with deforestation. The team received multiple awards, including a gold medal in the Overgraduate category, as well as the awards for best agriculture project and best plant-chassis project. (iGEM - Team Marburg, 2024)

Metalytic - Easy Automated Metal Biosensor

The project of the iGEM team HongKong-JSS aimed to build a variety of biological biosensors using BioBricks to detect heavy-metal contamination in various of settings. The team used metal-sensitive promoters to enable microbes to produce chromoproteins in the presence of metals, such as zinc, lead, cadmium and mercury. The team successfully developed a lead-sensitive and cadmium-sensitive detection system, with the lead-sensitive system being concentration sensitive. These systems are controlled by an automated hardware system using Internet of Things (IoT) and AI-technologies. The team won a gold medal in the high school category, as well as the award for the best hardware. (iGEM - Team HongKong-JSS, 2024)

ReneWool: Utilizing enzymatic degradation to upcycle mixed fabric waste into spider silk in *E. coli*.

The project of the iGEM team UAlberta involved engineering *E. coli* with BioBricks to produce enzymes that break down the components of cellulose-keratin-based mixed fibers into their monomeric protein and sugar subunits. These subunits can then be used as feedstock to produce spider-silk proteins in *E. coli*. The aim of this project is to combat the high accumulation of non-repurposed textile waste and to produce spider silk, which has applications ranging from aerospace and biomedicine to textiles and materials. The team won a gold medal in the Undergraduate category. (iGEM - Team UAlberta, 2024)

1.2 Science & Video Games

Video games are gradually establishing a role as tools for both formal education and lifelong learning, highlighting how they can support teachers and promote autonomous learning, which is a key component of 21st-century skills (Dede, 2009). Additionally, video games can be used for science popularization, making complex topics like geopolitics and societal challenges more accessible to a broader audience (Jacobs et al., 2017).

1.2.1 Games & The Act of Play

Various authors have proposed different, sometimes contradictory, definitions of what constitutes a game. Despite these differences, most definitions agree on the central importance of rules in delineating a game. One of the most influential definitions comes from Roger Caillois, who describes a game as a voluntary, rule-bound activity that is confined in time and space, uncertain in outcome due to player decisions or chance. Games are unproductive, and fictitious, requiring a "willing suspension of disbelief" (Caillois, 1961). Caillois also differentiates between two types of play: **ludus**, which is structured and rule-bound play; and **paidia**, free-form play (Caillois, 1955).

Johan Huizinga offers a similar definition, characterizing play as a free activity outside of ordinary life that, while not serious, deeply absorbs the player. He emphasizes that play is not connected to material interests or profit and occurs within its own boundaries of time and space, governed by rules.

Huizinga also introduces the concept of **immersion**, which hints at later ideas of engagement and flow in gameplay. (Huizinga, 1972)

These traditional definitions, however, face challenges from modern games, particularly video games. For example, in Massively Multiplayer Online Role-Playing Games (MMORPGs), virtual items and characters can be exchanged for real currency, making games productive in ways Caillois did not anticipate. Additionally, sandbox games, which often lack clear objectives or victory conditions, challenge the idea that games must have a defined structure or endpoint. In these games, rules are minimal, usually restricted to basic physics, and the player has significant freedom to create their own experience. This blurs the line between **ludus** and **paidia**, especially in games like Eve Online, where players can pursue personal goals without a predefined storyline or clear win/loss conditions. (Carter and Gibbs, 2013)

Contemplative games, like the poetic walking simulator **Dear Esther**, further complicate traditional definitions by focusing more on narrative exploration than achieving specific goals. These evolving forms of games have prompted scholars to seek new definitions that better capture the diversity of modern gaming experiences (Juul, 2010).

While these examples primarily focus on video games, the discussion also applies to other types of games, such as board games, role-playing games, and escape rooms. The primary distinction between these games lies in the medium through which rules and interactions are managed. In role-playing games, a game master typically oversees the rules, while in board games, this role might be shared between the game master and physical items like cards and pieces. In escape games, the physical environment often dictates interactions, with a game master monitoring from behind the scenes. Even when digital tools are used, such as computers in escape rooms, they are usually integrated into the game's narrative rather than serving as external aids.

1.2.2 Gameplay, Loops & Flow

The concept of "gameplay," is a more ambiguous term compared to well-defined elements like interfaces in digital platforms (Juul and Norton, 2009). Jesper Juul defines gameplay as the player's interaction with a game's rules and the overall experience of facing challenges and making choices within the game. It is focused on how the player controls and interacts with the underlying state of the game, distinct from elements like graphics, story, and audio, though these also influence the overall experience (Ryan et al., 2014).

Gameplay loops are a vital part of this concept, and they are defined differently depending on the context (Guardiola, 2016). In game studies, they often refer to the basic cycle of interaction and display. In the video game industry, they are categorized into micro, intermediary, and macro loops. For example,

in **Super Mario**, the micro loop involves actions like moving right and avoiding dangers, while the macro loop involves completing all levels of the game. In **Sim City**, micro loops involve building and handling emergencies, intermediary loops involve managing the yearly budget, and macro loops involve achieving long-term objectives like reaching a certain population level.

Gameplay, gameplay loops, and game genres are integral to the role of the game designer, alongside other elements like narration, story, and character interactions. Game designers focus on creating an engaging experience through level design and, the arrangement of game elements like traps, bonuses, and platforms. While aesthetics and graphics are central to a game's identity, they are not typically within the game designer's responsibilities.

Engagement and motivation are key concepts related to gameplay, often used interchangeably to describe a player's emotional involvement and investment in a game. Engagement is defined as the player's desire to direct emotions, thoughts, and affection toward the game, leading to emotional outcomes such as joy, pride, or frustration (Bouvier et al., 2013). An engaged player not only remains emotionally invested during gameplay but also continues to think about the game outside of play and wishes to keep playing (Bouvier et al., 2014). Motivation is seen as a willingness to participate, while engagement is more about ongoing behavior and emotional involvement.

The theory of flow (Chen, 2007; Sweetser and Wyeth, 2005) also ties into engagement, emphasizing the need to maintain a balance in challenge to sustain player immersion. Games that are too difficult can lead to frustration, while games that are too easy may cause boredom, which is why adjustable difficulty levels are common in video games. A graph illustrating the flow state can be seen in Figure 4.

Figure 4. A representation of the flow state. Adapted from (Chen, 2007).

The flow state is especially important in educational games, since learner engagement is integral to the effectiveness of education. When players are immersed in the flow state, they become more intrinsically motivated and learn more effectively. Clear goals, player agency, minimal distractions and immediate feedback are crucial to keeping players informed, motivated and

entertained. By carefully considering these design principles, developers can create educational games that are both fun and highly effective learning tools, fostering a more engaging and rewarding learning experience for players. (Kiili et al., 2012)

1.2.3 The Need for Interactive Digital Education

Education has been a state priority since industrialization (Carl, 2009), leading to significant efforts to democratize education, evidenced by the rise in global literacy from 12.05% in 1820 to 87.01% in 2022 (van Zanden et al., 2014).

The demand for affordable, large-scale education has spurred the development of new pedagogies and technologies. Modern pedagogies emphasize the importance of interaction between teachers and students, with students playing an active role in the learning process. This concept is rooted in Piaget's constructivism, which focuses on knowledge creation through interactions between new experiences and prior knowledge (Piaget, 1970).

Technological innovations have long been applied to education, from Edison's phonograph (Symes, 2004) to cinema, radio, television, video games, online courses, and smartphone-based educational apps. Despite these innovations, none have yet revolutionized formal education in a way that significantly reduces costs or simplifies the educational process, and the education system is often seen as being in turmoil (Mayes, 1995). This is attributed to various factors, including the challenges of engaging "Digital Natives" (Prensky, 2012; van Eck, 2006), children born into the Internet age who are perceived as challenging to reach through traditional educational methods.

The use of gamification in education began as early as the 1960s with projects such as teaching programming to children using the Logo programming language of MIT, and the PLATO program at the University of Illinois (Bitzer et al., 1961; Solomon et al., 2020). These early programs incorporated game-like features, such as reward systems and visual elements, to increase student engagement. They often used graphic assets set in familiar fantasy universes, like cartoons, space, or pirates, to create a more engaging learning environment.

1.2.4 Different Types of Interactive Digital Education

Video games (VGs) can be classified based on their context of creation and use; different VGs can target different audiences and prioritize either realism or fun. Since the concept of video games in education is so recent, the

terminology in the field has not been fully established yet. The following list provides an overview of different approaches and terms:

Digital Game-Based Learning: The use of VGs in education.

Simulators: Programs simulating real-world processes.

Educational Games: Games designed primarily for teaching, often used in academic settings.

Serious Games: Games developed for professional training, often including scientific or therapeutic purposes, sometimes categorized as "Games With a Purpose" (GWAPs).

Serious Gaming: Using VGs for training or education, whether they're serious games, educational games, or modified commercial games (Djaouti et al., 2011; Dunlap and Pecore, 2009).

The relationship between these terms is further illustrated on a Venn diagram in Figure 5.

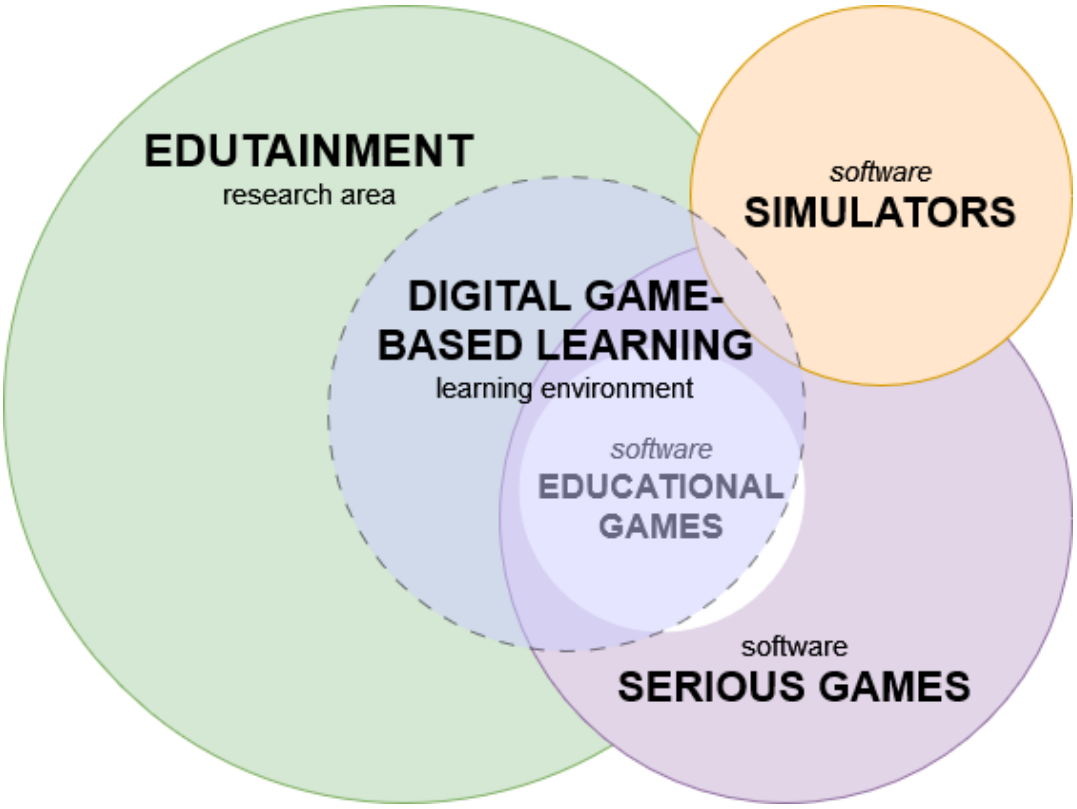

Figure 5. An adapted Venn diagram of relations between different types of interactive digital education (Tang et al., 2007).

There is no consensus on the types of science-based VGs, but these are among the most widespread internationally:

Educational Games: Developed by academic institutions to offer new or supplementary learning methods (Pashler et al., 2008). An example specific to biology include **CellCraft**, a game on cell biology that integrates higher education-level content into its gameplay (Dunlap and Pecore, 2009). A notable example specific to SynBio would be **Hero.Coli**, an educational adventure game where the player is tasked with modifying a bacterium with new abilities using BioBricks they find that help the player to progress further in the game (Goujet, 2018).

Science Popularization Games: Games like **Age of Empires** and **Kerbal Space Program** aim to popularize specific disciplines (e.g., history, space exploration) by balancing fun and realism, often with some inaccuracies. A specific example in genetics would be **Niche: A Genetics Survival Game**, a strategy game where the player evolves a species with specific genes to allow them to thrive in their environment with limited resources (Hossam, 2016).

Thematic Games: These serve as entry points to scientific topics but often contain inaccuracies**. Spore** is an example, where players guide a species through evolution, though the game has been criticized for its scientific inaccuracies, such as promoting an intelligent design view of evolution (Bohannon, 2008; Owens, 2012). Despite this, it can still be used for educational purposes with a critical approach (Schrader et al., 2016). Another notable example of games with biological themes would be **Subnautica** and its sequel **Subnautica: Below Zero**, where the player tries to survive in an extraterrestrial, alien underwater ecosystem, gathering both organic and non-organic resources. The ecosystem in these games is carefully crafted to invoke a sense of realism on how specific biomes could support a variety of life (*The Biology of Subnautica | Full Documentary*, 2022).

While educational and popularization games may have inaccuracies, they can still be valuable in engaging students and the public with complex topics.

1.2.5 "Games With a Purpose"

The interaction between research and VGs should be underlined by presenting additional examples. Fortunately, there are many games with a purpose (GWAPs), in subjects associated with synthetic biology.

One prominent example of these is **FoldIt** (Cooper et al., 2010), a game where players fold proteins to find their most stable 3D conformation. Protein folding is a complex problem due to the enormous number of possible configurations, and citizen scientists, who are good at recognizing patterns have helped advance protein folding research by playing **FoldIt**. The game has inspired other similar projects, such as **EteRNA** (Lee et al., 2014), **Nanocrafter** (Barone et al., 2015), and **Phylo** (Kawrykow et al., 2012), which focus on the folding and alignment of RNA and DNA.

Therapeutic games, which are used in mental health treatment and rehabilitation, allow patients to confront phobias or PTSD triggers in a controlled environment or guide physical exercise through exergaming (Wilkinson et al., 2008). Some therapeutic games also include data tracking or diagnosis features, helping medical professionals gather valuable information during treatment (Bamparopoulos et al., 2016). The introduction of Virtual and Augmented Reality has also been helpful for therapeutic purposes. One notable example of this is in reducing phantom limb pain for amputees, a topic on which multiple studies have been conducted (Ambron et al., 2021, 2018; Hali et al., 2024).

1.2.6 Related Interactive Media

There are various digital learning tools that share characteristics with digital game-based learning, particularly the examples of **gamified learning apps** and **simulators**.

Gamified Learning Programs and Apps:

- In the 1960s, programs such as the Logo programming language were used to teach children programming through interactive, game-like tasks (Solomon et al., 2020). These early efforts demonstrated that gamification could enhance engagement, although they also highlighted limitations in the transfer of knowledge from the gamified tasks to broader concepts.
- The rise of personal computers and later smartphones enabled the widespread adoption of gamified learning apps. These apps, such as **Duolingo** and **Memrise**, typically focus on language learning but also cover other subjects like geography, coding, and science. They use game-like elements, such as points, achievements and avatars, to increase user engagement while positioning themselves as serious educational tools. Additionally, these apps often incorporate the principle of spaced repetition (Ferguson et al., 2017; Kang, 2016), designed to reinforce learning by scheduling reviews at increasingly longer

intervals, based on the Forgetting Curve theory (Averell and Heathcote, 2011; Bailey, 1989).

Simulations and Simulators:

- Simulators are used for training by accurately recreating real-life events in a safe, controlled environment. They reduce stress and help learners become accustomed to situations they may encounter (Freitas and Maharg, 2011). Examples include flight simulators, which recreate realistic flying conditions, enabling pilots to practice procedures safely. Simulators are also used in professional settings for emergency protocols and in academic settings for scientific education.
- Some simulators blend digital and physical elements, such as professional flight simulators with realistic cockpits and digital displays. These tools have proven effective in education and training. The popularity of sandbox and crafting games in the video game industry shows that simulation-based learning tools also appeal to leisure users.

The classification of digital games related to learning or "serious" purposes is still up to debate; for example, **Kerbal Space Program** is a commercially available game based on realistic rocketry. While it has elements of both popularization games and simulators, it was not developed by professional rocket scientists.

1.3 Aim & Research Questions

As discussed in previous sections, the burgeoning field of SynBio holds immense potential in revolutionizing various fields of research and industry. Educating oneself on the complexities of the field requires a certain amount of background knowledge, however, and effectively conveying its intricacies in a simple form to the public has remained a challenge (Menard et al., 2024).

SynBio as a field would benefit substantially from bridging the informational gap between the general populace and experts of the field. For example, the understanding of genetically modified organisms (GMOs) across the global population is limited and divided, both due to misinformation and the complex, as well as controversial nature of the subject (Teferra, 2021). Easy-to-understand interactive education about SynBio could expedite policymaking related to the field, promote funding for SynBio projects, popularize the field,

combat misinformation and attract more individuals into the field, whether as researchers or entrepreneurs.

When it comes to existing citizen science games related to SynBio, such as **FoldIt** or **EteRNA**, the players do learn about protein folding, DNA structure and genetics through these games. However, the primary focus is on crowdsourcing design optimization rather than education. While players gain experience with scientific concepts, their experience tends to be more specialized in game mechanics and problem-solving processes rather than the scientific topics themselves.

With science popularization and thematic games such as **Niche** and **Spore**, players are given a set of mechanics that emulate genetics and evolution, encouraging them to think about how these mechanisms work in a more generalized sense. However, a deeper understanding of the underlying scientific principles is not the main purpose of these games. Instead, the primary goal is to provide players with enjoyable game mechanics that revolve around the themes of genetics and biology, making the science more accessible and engaging without necessarily aiming for in-depth education on the subject. Educational games, such as **CellCraft**, **Hero.Coli** and **Cell Lab**, teach the player about a small, very specific set of mechanisms. The scope of these games is quite limited, likely due to constrains such as time and budget. As a result, none of these games have yet provided the players a holistic understanding of all aspects of SynBio.

The concept of BioBricks presents a lot of potential in translating complex biochemical reaction chains into simple gameplay mechanics. This approach allows players to explore various aspects of SynBio in relation to the functionality of BioBricks. Initially the second aim of this thesis was to develop an educational game about SynBio, titled InGENEous, which would focus on designing genetic constructs with BioBricks in a realistic setting and linking these designs to real-world applications.

The scope of the game has expanded beyond just BioBricks during development. The current scope now includes not only designing biological devices, logic gates, circuits and systems, but all the steps of the Design-Build-Test-Learn cycle (DBTL), which are often applied in SynBio research (Gurdo et al., 2023; Kitano et al., 2023). This is achieved through the addition of a simulated laboratory where players can learn how laboratory procedures within the field facilitate the assembly of genetic constructs, as well as how to test and analyze the validity of these constructs. Greater emphasis is also placed on educating the player about how their knowledge relates to other fields, such as metabolic engineering, molecular biology and systems biology.

Additionally, the game covers the SynBio community and ethical considerations tied to SynBio.

A second set of research questions focus on the development of the game itself:

- Can an educational video game cover various aspects of SynBio thoroughly?
- How would an educational video game about SynBio facilitate the flow state?
- Can a game of this scope be developed rapidly and efficiently?
- Could an educational video game about SynBio be used for other scientific purposes?

2 Game Design of InGENEous

Before development of InGENEous could begin, a substantial amount of time was spent deciding on a reasonable scope for the game and establishing a set of core gameplay loops to support both the players in learning about the subject matter and to make the learning process as enjoyable as possible. Accessibility was also taken into consideration, and the game would be an open-source, non-commercial product with minimal system requirements. The target audience of the game would be anyone who can read English at an intermediate level. After these preliminary features were decided upon, a team was assembled to support the workload required for the game's development.

A game engine and software development kit, Unity, was chosen as the platform on which the game would be built. Unity was selected because it is free for non-commercial products, and all team members had some previous experience with it. The engine supports popular gaming consoles and operating systems (OS) such as WindowsOS, LinuxOS, MacOS X, AndroidOS, iOS, and browser-based solutions such as WebGL. Additionally, Unity offers extensive documentation and compatibility with a wide variety of third-party plugins (Unity, 2024).

A more thorough examination of hardware and software considerations can be found in Appendix A, which covers the hardware used for building the game, the hardware required for playing the game, and how Unity and the extensions used for the game's development function. If the terminology used in chapter 2.4.1 is difficult to follow, Appendix A will also provide the information necessary in understanding scenes, objects, components and scripts.

2.1 General Overview of The Game's Structure

The game begins with an introduction to both the game's story, and the subject matter. The story starts with the player travelling to a summer internship at a synthetic biology lab, Aikio labs. The player's future employer, professor Nuutti Aikio, has sent an interactive presentation, to help the player catch up on biology, evolution, DNA, genetic engineering, synthetic biology and the concept of BioBricks. After the introduction, the player arrives at Aikio labs and discovers that professor Aikio has gone missing.

While exploring the premises, the professor's lab assistant robot, LAB.RR, greets the player and explains that since the professor is missing, the player must start working immediately to fulfill the needs of the laboratory's clients.

LAB.RR installs the synthetic biology design software, EnGENEer, on the intern's laptop, where the player discovers a useful tutorial made by the professor for working with the software.

After finishing the tutorial, the player should have a good initial understanding of how different core BioBricks function together. Then they receive an email from their first client. The email explains what the client requires the player to design for them. The player is instructed to discuss the project further with the client. After the discussion, the player has a project to complete within EnGENEer, where they must produce a specific biomolecule using a given chassis. After completing the project, the player returns to discuss the design with the client, receiving a reward in the form of more information about the client, or cosmetic rewards. During this discussion the player may also ask the client questions about their area of expertise, how synthetic biology ties to their field, and personal questions to get to know the client better.

The player then receives another email from another client, and the game continues progressing in this manner. The projects increases in complexity, and after 3-5 projects the player will receive a new tutorial that teaches them how to navigate more complex game mechanics, such as chassis choices, producing multiple proteins, making logic gates and so on.

After all design-focused projects have been completed, the player must complete a laboratory and biosafety exam for LAB.RR to gain access to the laboratory, where the player is introduced to the rest of the DBTL-cycle. Under LAB.RR guidance, the players can use tools and devices to cut and ligate DNA, introduce their designs into plasmids, clone strands of DNA and more. This will allow the player to learn the "build" part of the DBTL cycle.

The laboratory also contains analytical devices, a bioreactor and other equipment, allowing the player to test whether their designs are functioning, and how they could further optimize the expression of their desired compounds. Using these devices, the player may also be tasked with fixing devices, circuits or systems that aren't functioning based on analytical data. This part of the laboratory experience will help the players understand the "test" and "learn" parts of the DBTL-cycle. Unlike design tasks, which produce theoretically functional genetic constructs, the laboratory will only serve as a simulation, since accurate metabolic models are far beyond the scope of the game. However, even if the numbers aren't accurate, players will still be educated on what kind of changes into their designs and laboratory protocols can increase or decrease their desired results.

The educational part of the game will end when the player has learned all the material taught by the game. As an additional challenge, players can use their new-found knowledge to solve the mystery of the missing professor, which is connected to SynBio and its ethical considerations.

2.2 Background Information

Background information about biology, gene expression and the methods and mechanics of SynBio is crucial in adding context to the players' actions within the game. However, background information isn't the main focus and should be taught using a multimedia approach instead of fully interactive gameplay to decrease the total development time of the game.

The introduction sequence of the game contains background information required to understand what BioBricks are and why they work, split into multiple chapters:

- Chapter 1. Prehistoric Life and Evolution
- Chapter 2. DNA, the Code of Life and How It Works
- Chapter 3. Using DNA and Genetic Engineering
- Chapter 4. Synthetic Biology and BioBricks

The player is prompted to choose a chapter they want to start with, allowing the player freedom based on their initial level of knowledge. The introduction is a collection of text, narration, music, animations and hand-drawn illustrations in the form of a presentation. Some of these illustrations can be seen on Figure 6.

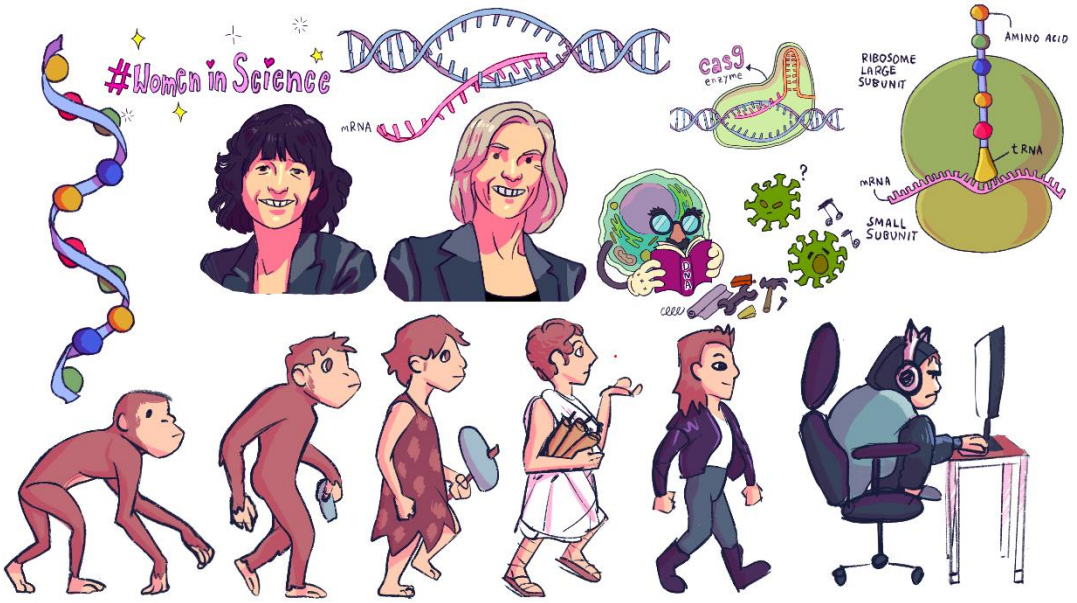

Figure 6. Hand-drawn illustrations from the introduction. Art by Zsófia Kecskés and Monika Tuominen.

The information in the introduction is presented by professor Nuutti Aikio, and the player can move at their own pace, as the presentation slides change only when the player interacts with the game. As an additional form of educational freedom, the player can interact with highlighted words in text to access a glossary entry, defining the meaning of the term. An example of this feature can be seen in Figure 7.

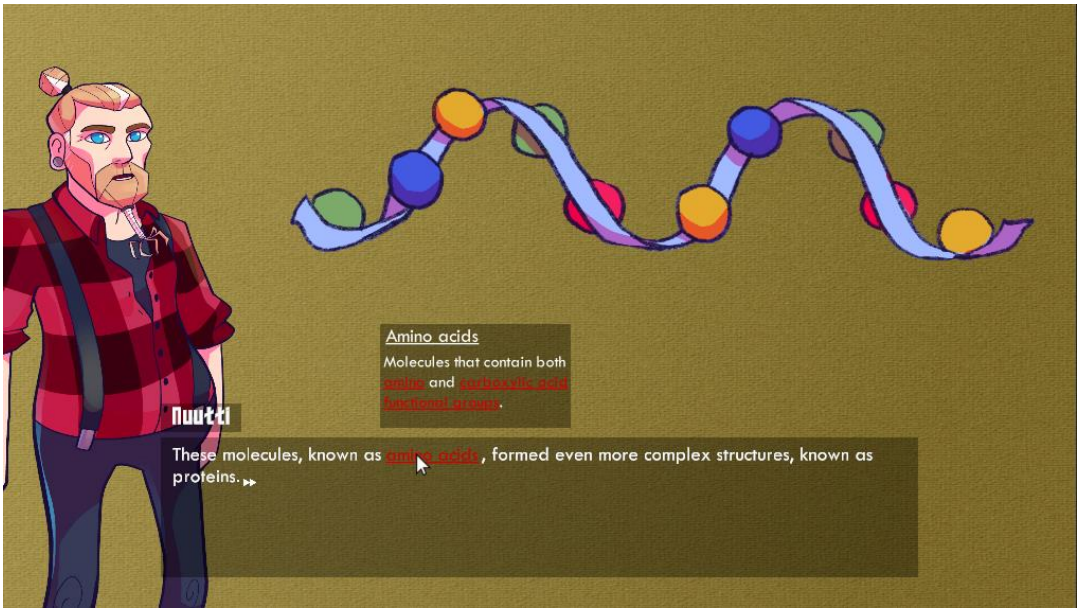

Figure 7. A screenshot from the introduction scene of the game, with a mock-up of the glossary feature. Art by Zsófia Kecskés and Aarni Aspi.

This feature allows the main text to both be shorter and tailored to an expert audience, without leaving less knowledgeable players behind. The glossary feature also extends into the tutorials and the game's database, allowing players to catch up on terminology wherever it is needed.

The game also includes documentation — an in-game book that compiles most of the educational in-game text and tutorials for easy reference. This feature allows players who might take a longer break from InGENEous to quickly regain their understanding on the subject matter and the game mechanics, without needing to replay earlier sections of the game. The glossary is included alphabetically at the end of the documentation, enabling players to review word definitions whenever they choose.

All these examples provide multiple avenues for the player to learn about the background information required for the game. In addition, some background information is also presented in discussions with the game's characters, as well as through interactable objects in the game's environment.

2.3 Community & Ethics

Another aspect of SynBio that is important to emphasize for players is that SynBio, BioBricks and associated online databases rely heavily on collective contributions. Therefore, the game should simulate various facets of the work performed by SynBio researchers, linking the player's actions and real world actions. This includes working on a computer and in a laboratory while collaborating with others and using BioBricks made by the community.

The design projects are both assigned to the player, and completed on an in-game computer, simulating how designing with BioBricks happens in the real world. The in-game computer has two main functions: the communication app "Forum Quorum", and the design app "EnGENEer". The following Figure 8. illustrates the main screen of the Forum Quorum app.

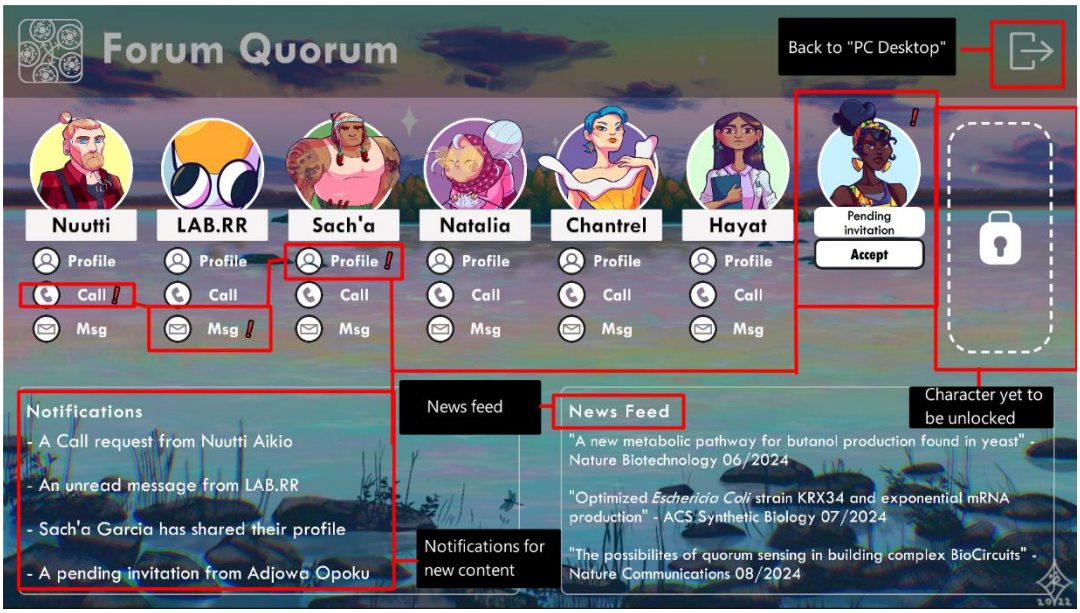

Figure 8. A screenshot of the Forum Quorum main screen. Art by Julia Rässa and Zsófia Kecskés. UI art by Tuomas Hellsten.

Forum Quorum is a fictional communication platform that enables the player to receive emails, view profiles of the game's fictional characters and engage video calls with them. Forum Quorum simulates the necessity of digital communication for global collaboration in real-word SynBio-research. Additionally, the platform features a news feed that pulls recent SynBio-related articles from the internet, allowing the player to be informed about the latest developments in the field while emphasizing the community-driven nature of SynBio. The player receives new design projects from clients via email in Forum Quorum, after which they may discuss project details with the clients further in video call. An example of these video calls can be seen in Figure 9.

Figure 9. A screenshot from the dialogue scene of InGENEous. Character art by Zsófia Kecskés, background art by Alli Loikkanen.

The clients the player collaborates with are fictional characters, each playing a critical role in conveying information about their field of specialization and its connection to SynBio. The projects they assign to the player are directly tied to their respective fields, enabling the player to grasp how their designs can address real world problems. A brief overview of these characters is provided in Figures 10. and 11.

Figure 10. The clients and their occupations. Character Art by Zsófia Kecskés.

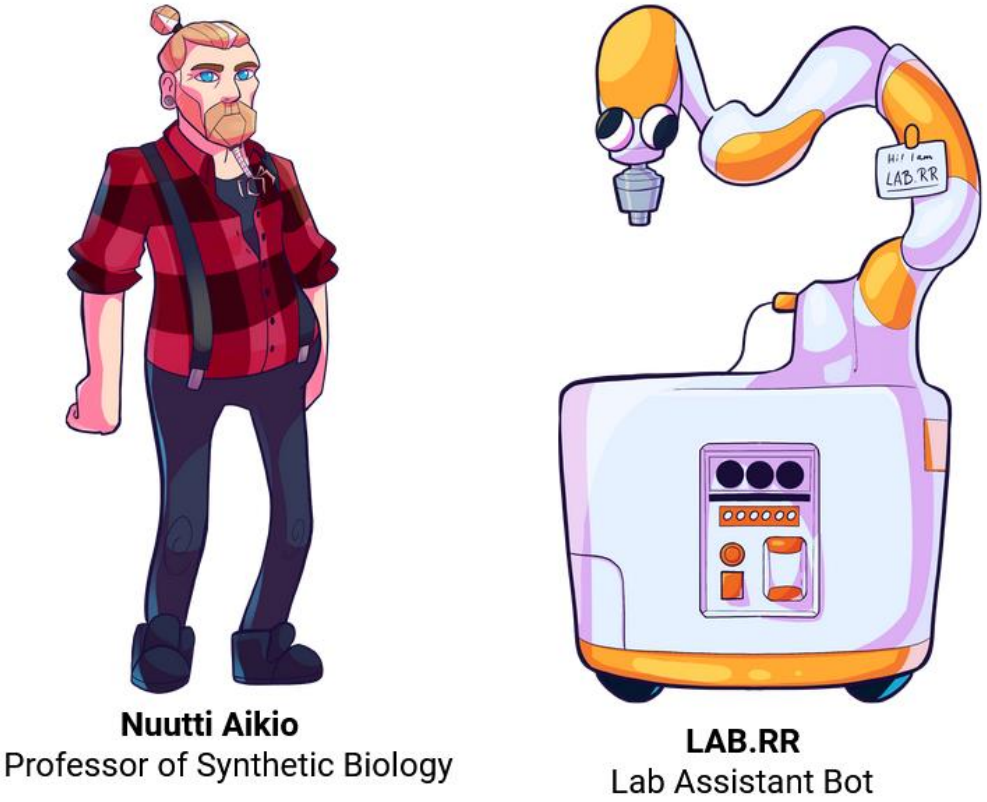

Figure 11. The players' coworkers. Character art by Zsófia Kecskés.

Apart from the clients, the game also has two other characters: professor Nuutti Aikio, the player's mentor, and LAB.RR, the lab assistant robot Professor Aikio introduces foundational concepts and sets the stage for the player's journey into SynBio. LAB.RR guides the player through laboratory safety, protocols, and equipment, ensuring they understand essential practices in a real-world lab setting.

The ethical considerations associated with SynBio, and the use of GMOs are also conveyed through naturalistic interactions with the characters. The player can engage in conversations with the characters on various topics related to SynBio, during which the characters present diverse perspectives on the ethics and safety of SynBio. This allows the player to understand the ethics of SynBio from multiple viewpoints, rather than being presented with a single, subjective perspective.

EnGENEer is the other app the players can access through their in-game computer. It is a fictional design software within the world of InGENEous, and the players use this software to solve the design projects presented to them by clients. The projects screen of EnGENEer can be seen on Figure 12.

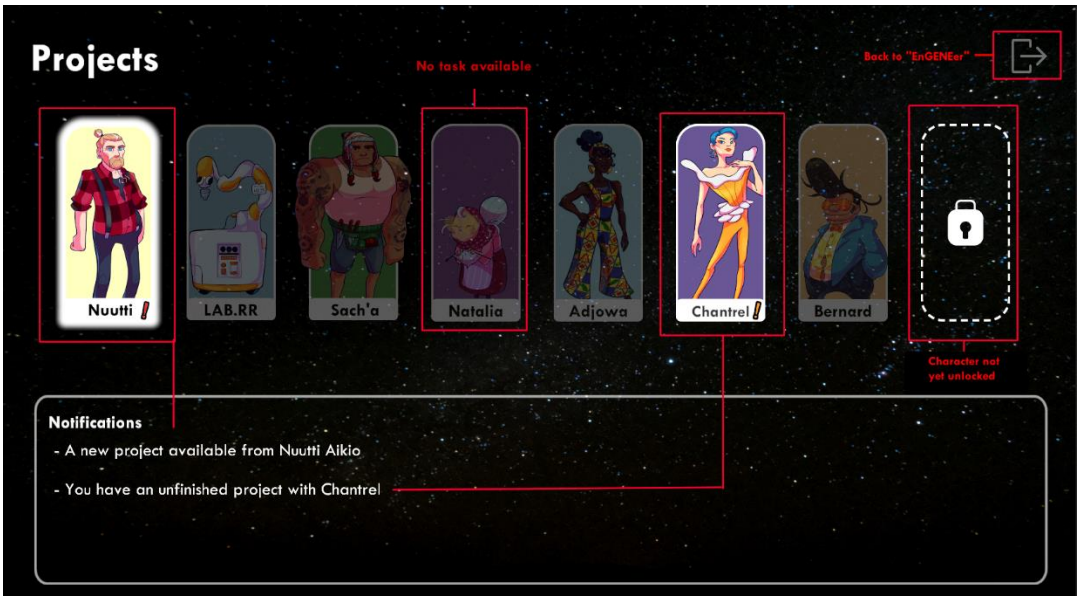

Figure 12. A screenshot from the projects overview screen. UI art by Tuomas Hellsten and Aarni Aspi.

The players may also access and browse the in-game database within EnGENEer. This feature allows players to familiarize themselves with different categories of BioBricks and other useful entries. The database and the various categories within are covered in more detail in chapter 2.4.

The features of the EnGENEer app highlight to the player how the work in SynBio is done through the use of software and online databases. The in-game database contains a curated selection of BioBricks from online databases, and these database entries contain hyperlinks to the actual parts, as well as the metadata and authors associated with each part. This emphasizes to the player how these databases are created as a collaborative effort.

This collaborative element is also present in the game's main menu, which has a button titled "More Games". When the player interacts with this button, they are presented with a list of hyperlinks to other games related to SynBio, such as **FoldIt** or **Hero.coli**. This provides the player an avenue to learn about SynBio and related fields through a different game, or to use their newfound knowledge to participate in citizen science projects.

2.4 The DBTL-cycle & BioBricks

From a gameplay standpoint, the DBTL-cycle is the most important facet of SynBio to teach the player. This includes all the steps of the DBTL-cycle, and what kind of procedures are associated with each step.

The design step includes core BioBricks and the correct order they should be assembled together in to create functional genetic devices in either prokaryotes or eukaryotes. It is also important for the player to understand how both transcription factors and external inputs function in relation to regulatory promoters, and how on/off switches can be used to create biological logic gates, circuits and systems. Additionally, the player should learn how to choose or modify their chassis, including optimizing the whole genome of the chassis to reduce burden and increase production of desired biomolecules.

The player should also understand how plasmids work, how DNA can be cut and cloned, and how to perform these tasks using PCR. The player should also be taught how to test their designs and how to analyze their results, and how these results can be used to optimize the production of their designs.

Teaching all this information to the player will take time, and it is better to approach these topics with a gradual increase in complexity. Whenever the player is introduced to new features of the design tool or laboratory via a tutorial, they receive a set of projects that are roughly equal in difficulty, but varied in the approach the player must take to complete them. This way, the tutorials can teach the players about larger concepts, such as devices, logic gates or systems, while individual projects can focus on smaller concepts and limitations using the game's modifier system, such as using phage promoters, modifying the chassis and so on. An example of the game's curriculum:

Chapter 1. Single device, chassis choice and prokaryotes

Tutorial 1. The device view

Project 1. A constitutive protein generator with a prokaryotic chassis.

Project 2. A constitutive protein generator that produces two proteins under one promoter. Players must choose a prokaryotic chassis themselves.

Project 3. An RNA generator using a phage promoter. Requires the player to modify a chassis to enable phage promoters or choose a chassis that supports them.

Chapter 2. Regulatory promoters, circuits and eukaryotes

Tutorial 2. The circuit view

Project 4. A regulatory measuring device, chassis choice up to player.

Project 5. A regulatory protein generator with a regulatory reporter system.

Project 6. A regulatory protein generator that produces 2 proteins in a eukaryotic chassis. Requires an IRES.

Project 7. Build a circuit that works as an inverter. Works as a lead up to logic gates.

Chapter 3. Logic gates

Tutorial 3. Gate view

Project 8. AND-gate with new modifiers and limitations

Project 9. OR-gate with new modifiers and limitations

…. And so on.

Once the player has access to the laboratory, they may then revisit some of these projects, prompted by the client. Some of these designs do not work initially, or they need to be assembled, tested or optimized. Thus, the first few projects that require laboratory work use the same designs the player already made, allowing them to continue with their own work. This approach helps the player maintain familiarity with the purpose of the design, enabling them to focus primarily on the laboratory procedures and optimization.

The main gameplay loop of InGENEous has the player solving various design projects using the in-game design tool, EnGENEer. The player is periodically taught how to use the design tool via in-game tutorials, which highlight how various parts of the design tool function. An example of an in-game tutorial can be seen on Figure 13.

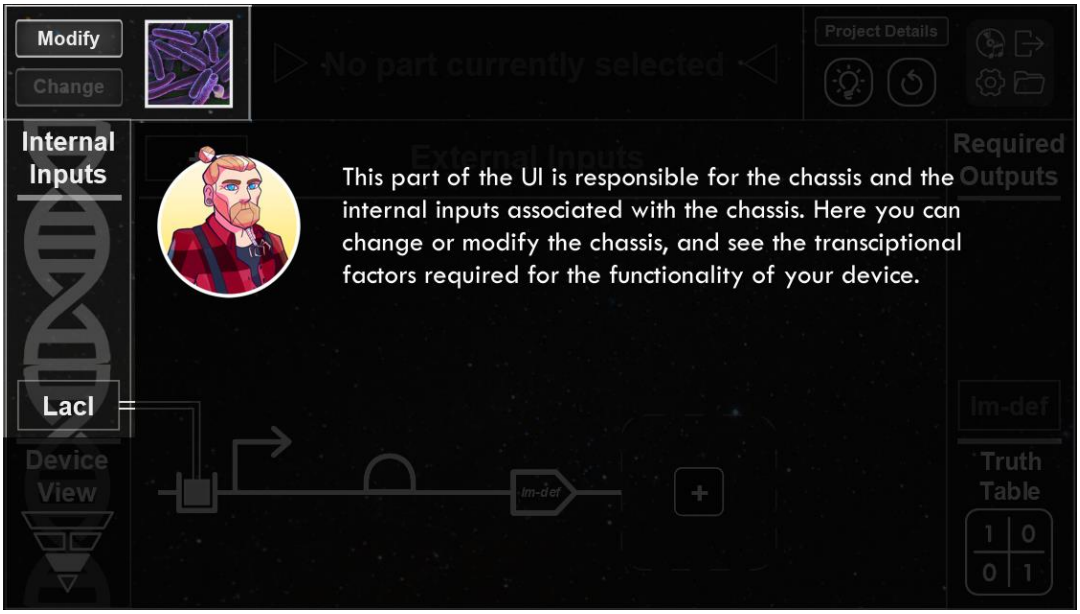

Figure 13. A mock-up of an in-game tutorial. UI and diagram by Aarni Aspi.

Once a player has completed a tutorial, a few projects are assigned to them by clients. A project might involve multiple different tasks, requiring the player to work both on the computer and in the laboratory. An example of an individual project screen can be seen in Figure 14.

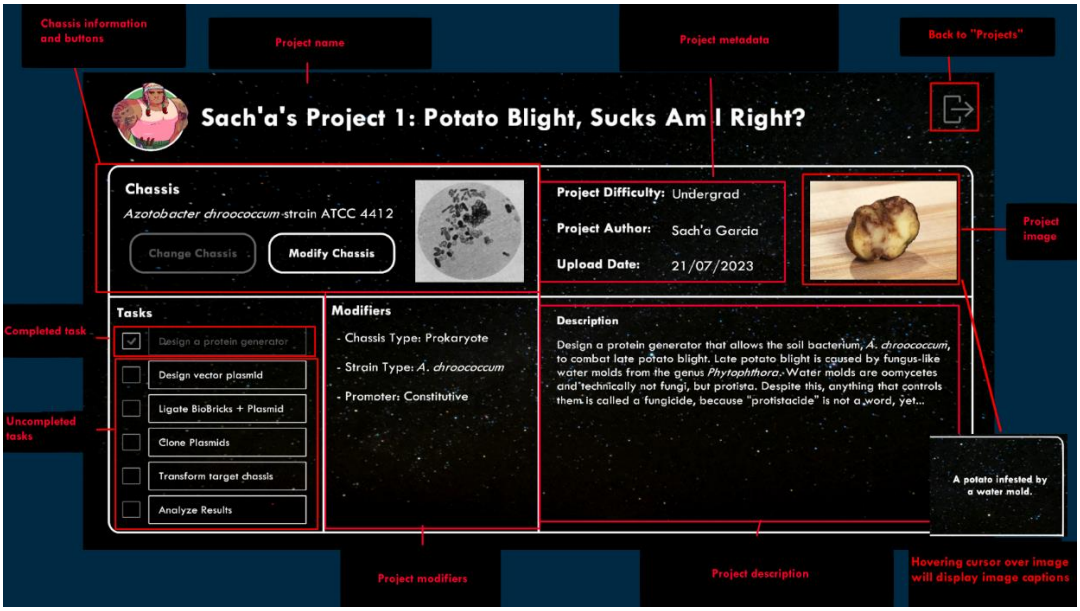

Figure 14. An edited screenshot of the project screen. Art by Zsófia Kecskés. UI art by Tuomas Hellsten and Aarni Aspi.

This screen shows the players all the information they need in relation to the current project. Once the player selects an individual task from the project screen, the game opens the design tool, and the work begins. The design tool is responsible for all tasks requiring the design of a device, circuit, logic gate or system using BioBricks. The design tool is split into separate “magnifications”; the sequence view, the device view, the circuit view, the gate view and the system view. The following Figure 15. shows the most recent mock-up for the device view, which is used for building singular devices.

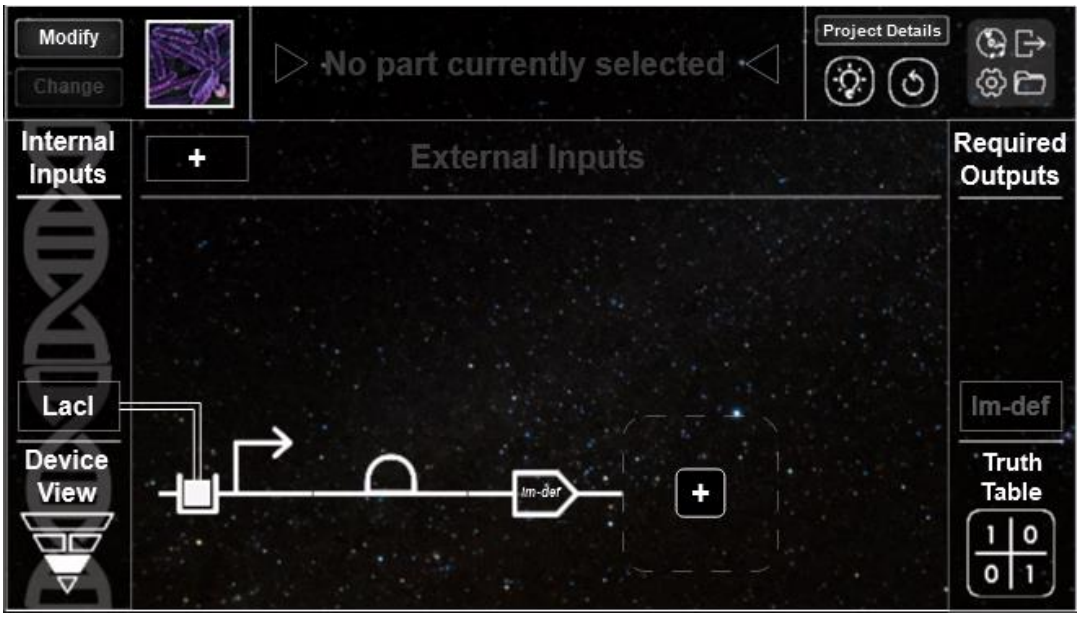

Figure 15. A mock-up of the device view. UI elements by Julia Rässa and Tuomas Hellsten. UI and diagram by Aarni Aspi.

The device view allows the players to perform a variety of actions required to design a functional device. These include changing or modifying the chassis and either adding or removing BioBricks in the design area. Players can also add, remove, activate or deactivate external inputs. Whenever players want to add or change something, they are taken into the in-game database, where they can choose the chassis, BioBrick, or external inputs they wish to use. After making their selection, the chosen component is placed in the design tool.

The device view automatically displays information relevant to the current task and project. This includes the required outputs on the right side of the screen, the internal inputs provided by the chassis on the left side of the screen, and the current magnification of the design tool. Other features include automated connecting lines for the inputs and outputs, and the selection window at the top of the screen, which allows players to view information about currently selected BioBricks, inputs, or outputs.

Additionally, the device view allows the player to review the project screen and contains a button for hints related to the current task, as well as a button for resetting the task. Other buttons allow the players to change the music, review documentation, modify the game's settings, and leave the design tool. Leaving the design tool will automatically save the progress of the current task, enabling the player wants to take a break from the game during a project.

Whenever the player creates a functioning device that produces the required output, they are prompted to save it into the in-game database. Most of the data associated with a device is sourced automatically by the game, but the player may write a short description and add their name as the author of the device. Once the player has completed all projects associated with the device view, they are introduced to the circuit view, seen in Figure 16.

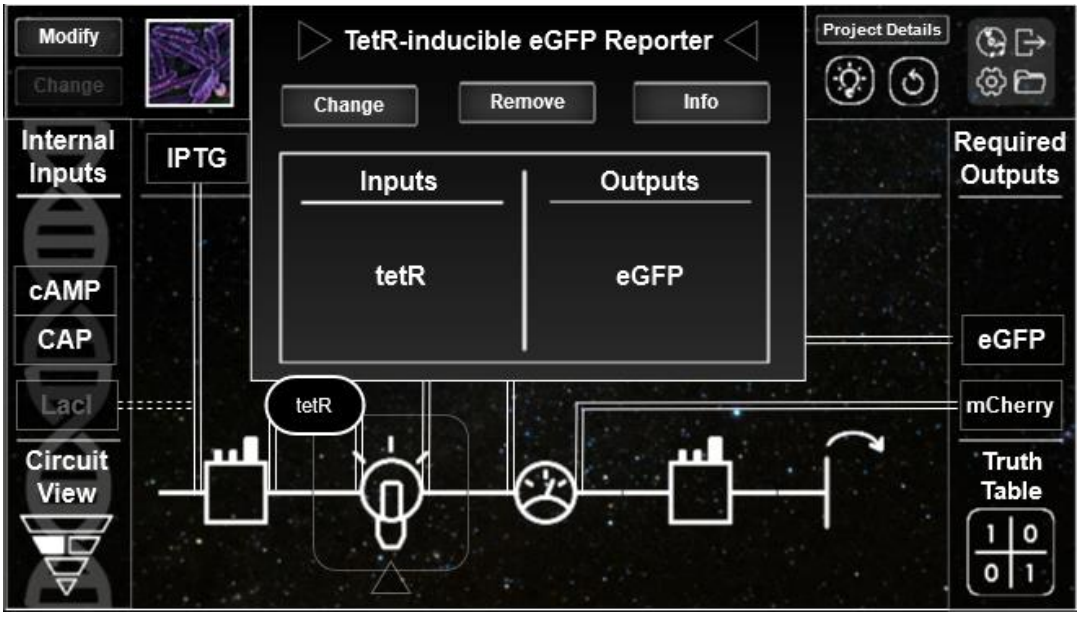

Figure 16. A mock-up of the device view, where a reporter device is currently selected, and its details are available in the selection window. UI elements by Julia Rässa and Tuomas Hellsten. UI and diagram by Aarni Aspi.

The circuit view works similarly to the device view, but in this magnification the players build circuits of multiple devices instead of singular devices. Players may use devices they have previously made or template devices from the game's database. If the players want to add a new device, the design tool enters into the device view, where they may either build a new device, or modify an existing device to their liking. Once the device is ready, its functionality is validated, and the device is saved into the game's database. The players are then returned to the circuit view, where their device and its associated inputs and outputs are now available in the design area.

The truth table, an important feature of the design tool, allows the players to review all the Boolean values associated with their chosen inputs and how their device, circuit, logic gate, or system changes the outputs based on different input-combinations. A mock-up of the truth table can be seen in Figure 17.

Figure 17. A mock-up of the truth table accessible in the design tool. UI by Aarni Aspi.

The values in the truth table are refreshed in real time, allowing the players to see data on where their design falls short in achieving the desired outcomes. The template truth table associated with a task can be toggled on or off, allowing players to compare their values with the required ones.

Players may also review each device separately to see their respective truth tables or view a summary that includes the entire circuit, logic gate or system along with the associated external inputs and outputs. This approach lets players either understand their design in a broader context or pinpoint issues in specific parts.

Within the truth table, players can also validate their design, prompting the game to verify that the BioBricks are in the correct order and compatible with each other and the chosen chassis. This is required if the players create or modify a device through the circuit, gate or system view, to prevent dysfunctional devices being included in the database.

The truth table also allows the players to enter the free draw mode if they wish to make an informative diagram of their design. Additionally, players can export the truth table as comma-separated values (CSV-file), enabling them to open their truth table as a table or a data sheet outside the game. The free draw mode, data exporting and their use is covered in more detail in chapter 3.3.

When the players create a functioning circuit that produces the required outputs, they are prompted to save it into the in-game database, similarly to the device view. All the data associated with the devices in the circuit is saved

into the database as a single entry and becomes available for use in future projects. Once the player has familiarized themselves with circuits, they are introduced to the gate view, seen in Figure 18.

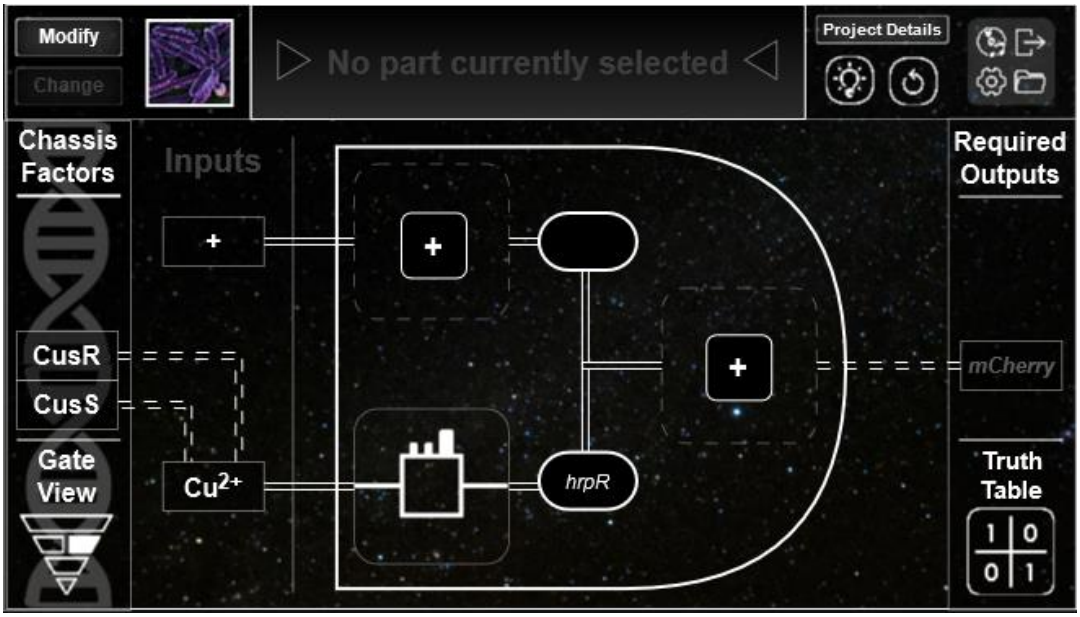

Figure 18. A mock-up of the gate view. UI elements by Julia Rässa and Tuomas Hellsten. UI and diagram by Aarni Aspi.

The gate view allows players to choose a template logic gate, after which they can build their functionality by using devices, similar to the circuit view. In this mockup, the chosen template is an AND-gate. This view works quite similarly to the circuit view, with the main difference being that the inputs are moved to the left side of the screen for better spacing.

Similarly to the device and circuit view, the players can also save their gates into the in-game database after meeting the task completion criteria. In this view, the truth table also becomes more important, as the template truth table of the current task is required to verify that the task has been completed correctly. Once players have completed all logic gate-related tasks, they can move on to working on systems of multiple gates and devices. The system view can be seen in Figure 19.

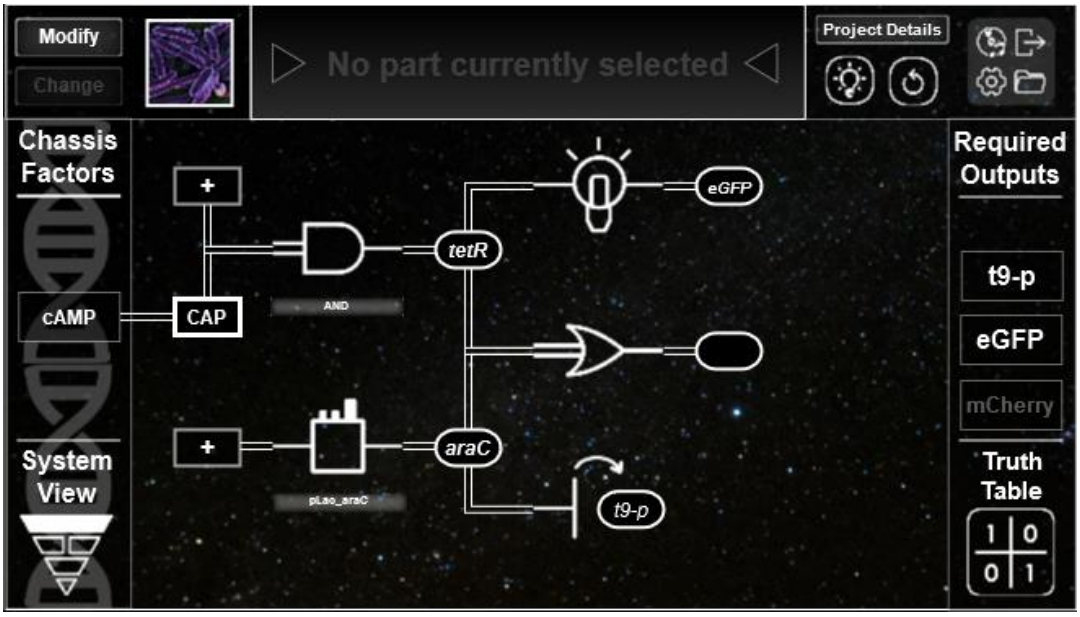

Figure 19. A mock-up of the system view. UI elements by Julia Rässa and Tuomas Hellsten. UI and diagram by Aarni Aspi.

The system view allows the players to build systems that include multiple logic gates and devices, enabling them to familiarize themselves with more complex tasks requiring advanced Boolean logic. In this view, the truth table becomes even more important, as task completion will require specific outputs in relation to various input combinations.

Whenever players add a device or gate into a slot in the design area, the game will automatically create buttons for the inclusion of both inputs and further slots in the system. However, this view is still a work in progress, and the game's programming must adjust slot creation to ensure the system itself only expands horizontally, and not vertically. This adjustment should be made to improve text visibility.

For systems that use multiple different chassis, the background should include highlighted areas that group all the devices and logic gates associated with a specific chassis. These issues are solvable through programming, and are currently in the prototype stage, as the focus has primarily been on the device, circuit and gate views. Nonetheless, players can still save their systems into the database once the criteria have been met.

The final view, the sequence view, is only accessible from within the game's database. However, the sequence view will become more relevant in the later stages of the DBTL-cycle and will primarily serve as an additional source of information during design projects. A screenshot of the sequence view can be seen in Figure 20.

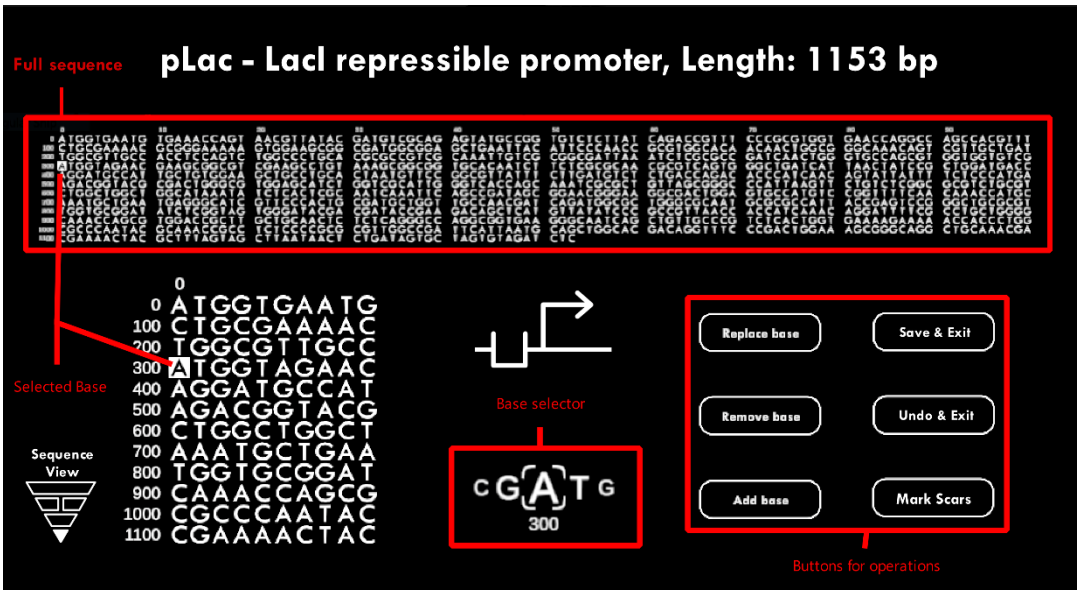

Figure 20. A screenshot from the sequence view screen. UI art by Julia Rässa and Jakub Dobi.

The sequence view allows the players to view the full sequence of a BioBrick, device, circuit, logic gate or a system. It can be accessed using the sequence view button on the currently selected part's database entry. The sequence view screen displays the name and length of the current sequence the player is viewing. Since the letters become quite small with longer enough sequences, two different magnifications were made for this screen.

The base selector displays the currently selected base and allows the players to move up or down across the whole sequence to select a single base. The sequence view enables players to replace, add, or remove a base pair at the currently indexed position in the sequence, which is highlighted in the base selector. Players can also mark scars by selecting an index range of bases the scar is located in.

This screen is very informative and provides players with a deeper understanding into DNA sequences. Sequence editing will also be useful for players who wish to export their designed sequence out of the game, either in an edited or unedited form.

Since the sequence view is accessed within the in-game database, and the database is integral to using the various views of the design tool, it is important to also highlight what the database contains and how it functions. A screenshot of the in-game database can be seen in Figure 21.

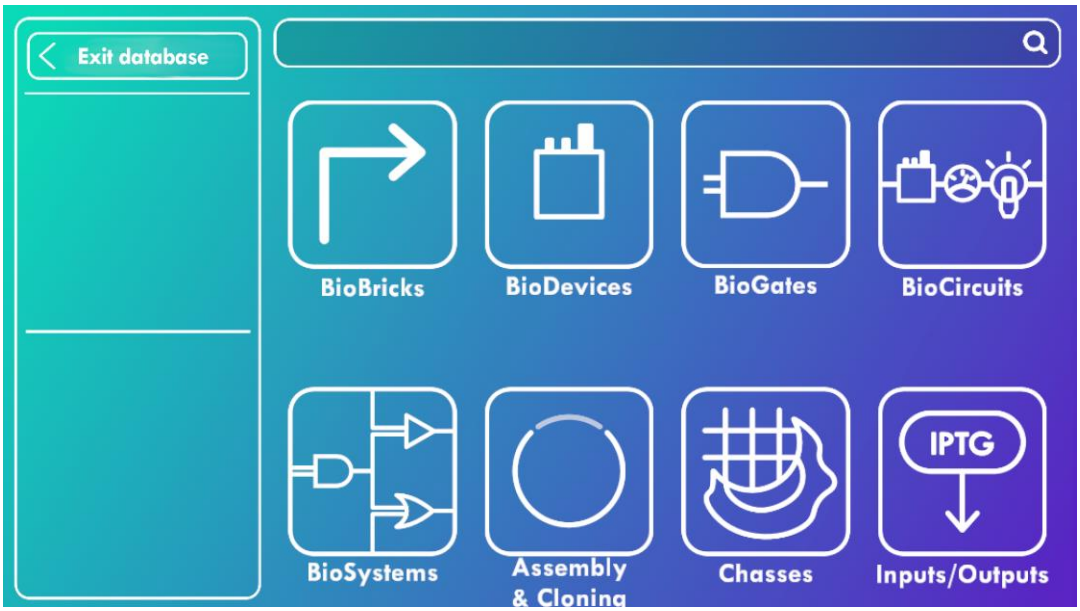

Figure 21. A screenshot of the home screen of the in-game database. UI art by Julia Rässa and Tuomas Hellsten.

The database is currently split into eight main categories: BioBricks, BioDevices, BioCircuits, BioGates, BioSystems, Assembly & Cloning and Inputs/Outputs. For consistency, the prefix "Bio" is added before each category to indicate these are built with BioBricks, even though this isn't the current consensus in the field. This nomenclature helps distinguish these constructs from the nomenclature used in electrical engineering and provides more information without needing additional context. The contents of these categories can be seen in Figures 22, 23 and 24.

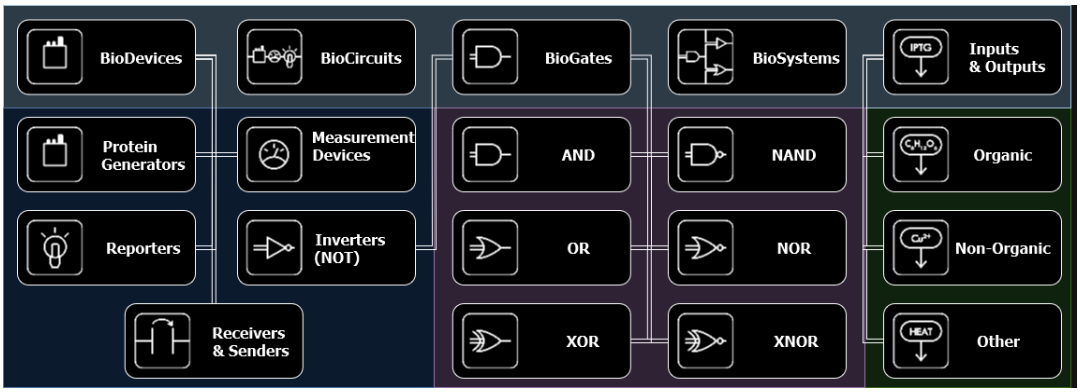

Figure 22. A diagram of several subcategories from the in-game database. Icons by Julia Rässa. Diagram by Aarni Aspi.

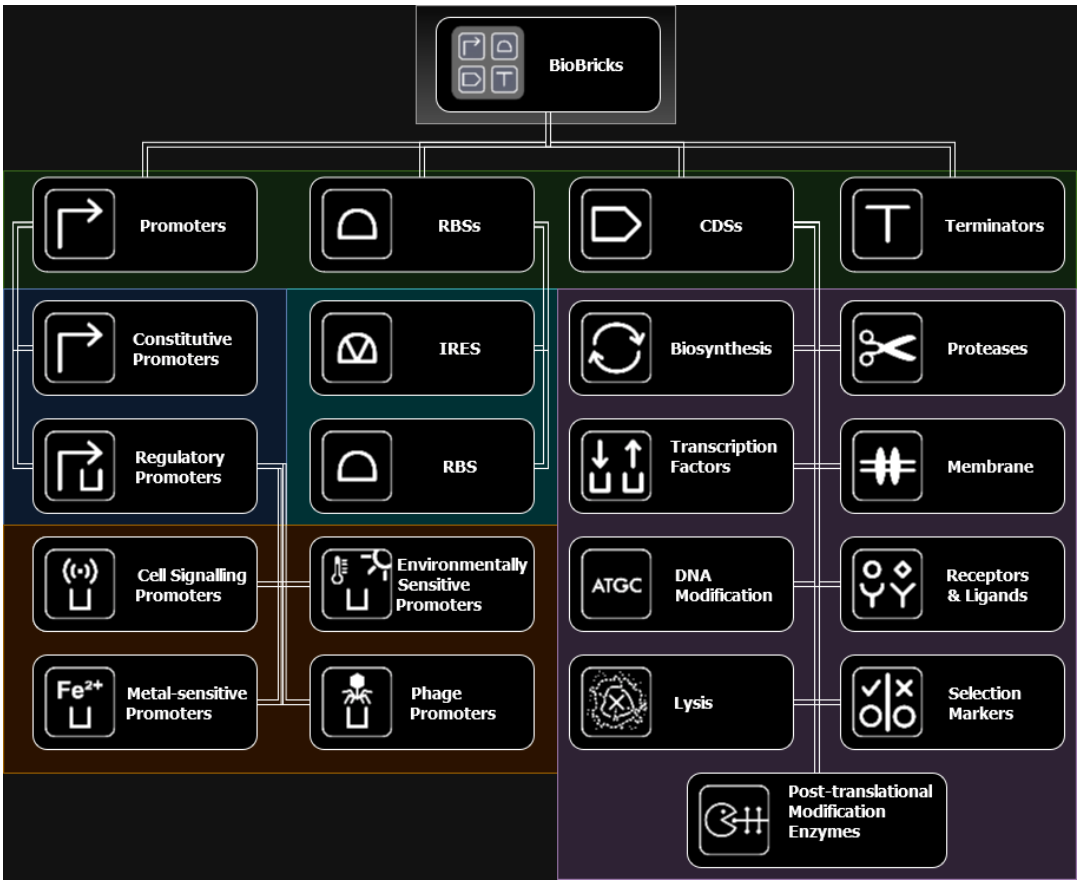

Figure 23. A diagram of the subcategories of BioBricks. Icons by Julia Rässa. Diagram by Aarni Aspi.

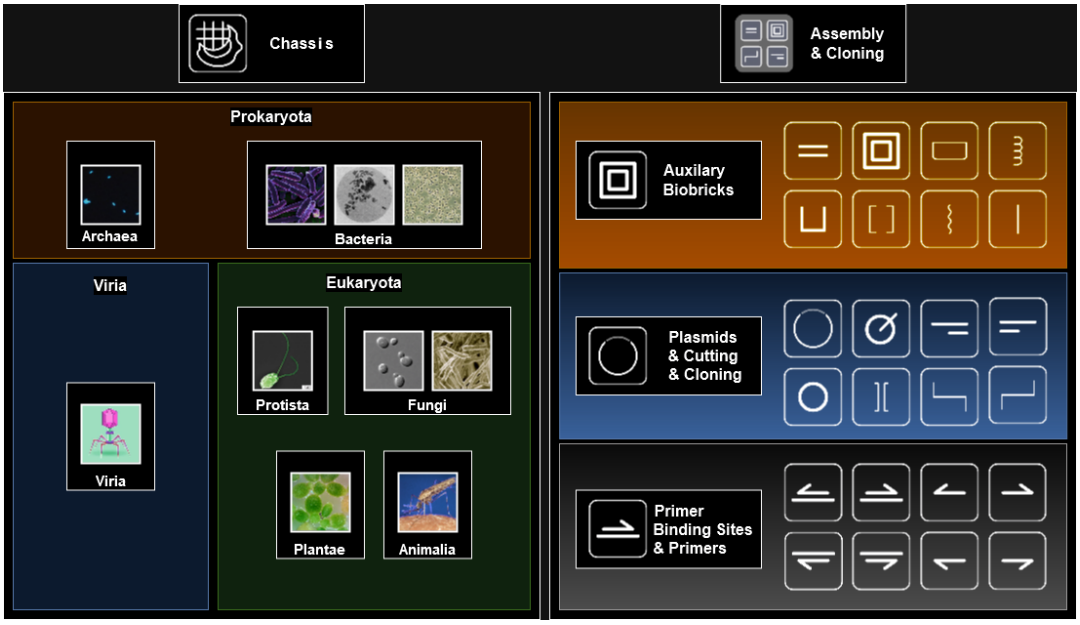

Figure 24. A diagram of two main categories, chassis, and assembly & cloning. Icons by Julia Rässa, Diagram by Aarni Aspi.

These diagrams were made to show the breadth of the game's database. The categorization of the various parts in the database is still under development, as there is no hard consensus on the categorization of these parts among the SynBio community. The current categories have been created to allow the player to familiarize themselves with the database slowly.

The actual database in-game only displays a few categories at a time, ensuring that entries and information within each category are fully visible on a single screen. This design allows players to easily choose between categories and entries without needing to scroll extensively within a category. Each subcategory contains a list of database entries, as seen in Figure 25.

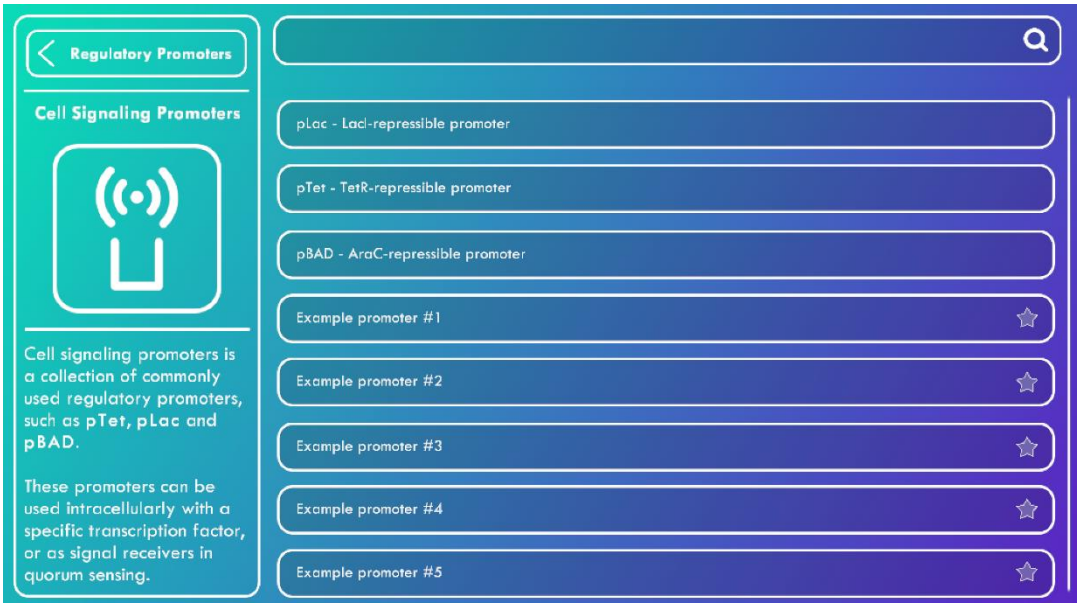

Figure 25. A screenshot from the in-game database, highlighting a list of individual database entries in a subcategory. UI art by Julia Rässa and Tuomas Hellsten.

The individual entries in the list consist of both original BioBricks from the game's database, and BioBricks modified or made by the players or educators. These custom entries are denoted with a star on the right side of each individual entry, and this same notation applies to player-made devices, circuits, logic gates and systems as well.

Each entry contains a curated set of information relevant to the players, enabling them to make informed decisions when selecting parts for their designs. This includes considerations such as chassis compatibility and the inputs and outputs of these parts. An example of an individual database entry can be seen in Figure 26.

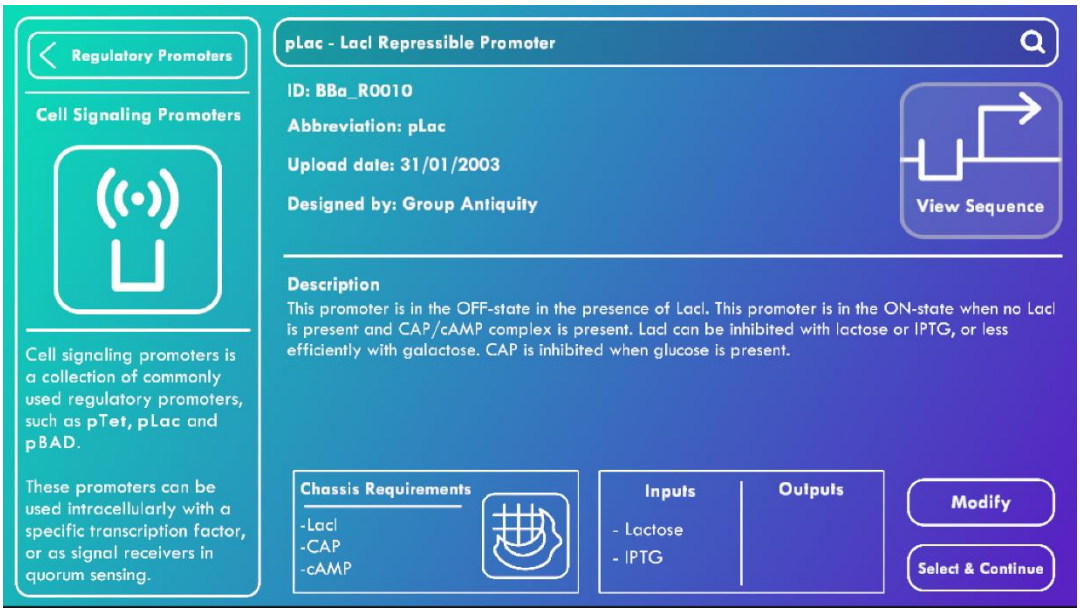

Figure 26. A screenshot of an individual entry from the in-game database. UI art by Julia Rässa and Tuomas Hellsten.

Each individual database entry varies depending on the category of the chosen part. For example, a promoter entry contains information about the inputs required for its on/off functionality, as well as the chassis requirements. CDS (coding sequence) entries, on the other hand, provide information about the output produced by these CDSs, while RBS (ribosome binding site) and terminator entries include details about their strength. Chassis entries list the transcriptional factors that these chassis provide, which are crucial for enabling specific functionalities. This system of detailed entries helps players make informed decisions based on their design needs.

All the entries in the database have a set of parameters associated with them, and each field in an individual entry is pulled as **key-value pairs** from a **JSON**-file that contains the entire database. This structure ensures that the database is organized and accessible for the game, allowing for seamless integration of parts and their attributes. Examples of these database entries are available in Appendix B.

Once the players have completed all projects using the design tool and database, they will gain access into the laboratory. However, before entering to the laboratory, LAB.RR requires the player complete a general laboratory and biosafety exam. This step is designed to emphasize the importance of safety precautions when working in a laboratory mirroring real-world practices and ensuring that the player understands the basic safety protocols required in biological research settings.

The players are provided with learning materials in the form of a comprehensive presentation to study, covering essential laboratory and biosafety concepts. Afterwards, they must answer a set of multiple-choice questions correctly in dialogue with LAB.RR. If the players fail to answer correctly, they are given a new randomized set of questions until they pass the exam. This process ensures that players have a thorough understanding of laboratory safety before proceeding to the actual laboratory work.

After passing the exam, the players gain access to the laboratory, entering the later stages of the DBTL-cycle. This will enable the players to tackle more complex projects, which may involve tasks such as cloning, using PCR, reading analytical data, operating a Bio Stat and so on. The players will also receive a tutorial on using each device, guided by LAB.RR. The laboratory is divided into three main sections, each with distinct functions. A mock-up of these sections, along with their subfunctions can be seen in Figure 27.

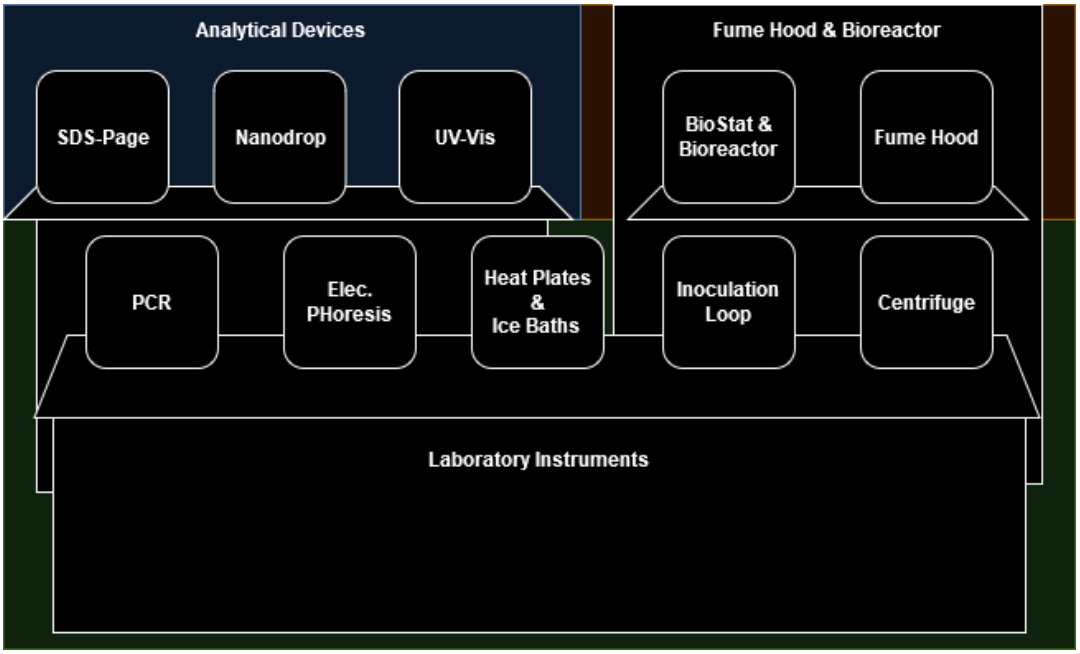

Figure 27. An initial concept for the laboratory. Diagram by Aarni Aspi.

In the laboratory, players will learn how to assemble their designs and introduce them into their chosen chassis. They can then grow their cultures and analyze their results using analytical devices to verify the functionality and efficiency of their designs. The projects can also include preset designs and analytical data, allowing players to adjust their production by modifying these designs based on the data, or even optimizing the genome of their chassis.

Since the BioBricks and chassis have a large amount of data associated with them, this data can be used to approximate how changes in the design of genetic constructs or genome of the chassis will affect the burden on the chassis or the output. Although accurate numerical values would require extensive metabolic modeling to be included in the game, players can still learn about the impact of orthogonality and other factors that affect the strength of expression, such as RBS and terminator choice. Various graphs in the laboratory will either increase or decrease based on the players' choices, teaching them how to optimize the functionality of their designs. This allows the players to understand how different design choices can influence the overall performance and efficiency of their synthetic biology projects.

This part of the game is only in the initial design phase however, because the test version of the game was intended to include only the design phase of the DBTL-cycle. Once the test-version of the game has been released and player feedback has been received, this feedback will then be used to guide the design and scope of the laboratory itself.

All these features of both the design tool and the laboratory should provide the players with a strong foundation of knowledge about the different phases of the DBTL-cycle. The functionality of these features requires extensive programming to both automate processes for the players and to validate that the players have designed functional devices.

2.4.1 Programming Structure

Some of the features of the game, such as Forum Quorum, the dialogue, the exploration, the design tool and the database, require substantial amounts of programming to function. These features are either related to the engine-side programming, or the science-related programming. The engine-side programming supports the game's structure itself and facilitates the science-related programming of the game.

The science-related programming is crucial, as it provides all functionality tied to the projects, the design tool and the database. This chapter provides insight into how these features are handled by the game's programming.

The projects of the game are currently created manually by filling **key-value pairs** associated with a project into a **ProjectData** JSON file. These values are then used to populate the project screen of a selected project with the associated information, as seen previously on Figure 14, and used to set the limitations and required outcomes of tasks related to a project. Table 1 contains a list of **keys** and **values** associated with a single project.

Table 1. Example data associated with a single project.

Key	Value	Function
LVLID	LVL1	Index of project.
LVLNAME	Potato Blight, Sucks Am I Right?!	The name of the project.
AUTHOR	Sach'a Garcia	The author details.
DATE	20/10/2024	Creation date.
DIFFICULTY	Novice	Difficulty.
TASKS	TASK1: Design a protein generator. TASK2: Find the right plasmid.	List of tasks associated with the project.
MODIFIERS	Chassis: *A. chroococcum*	Modifiers.
ERRORS	lm-def =! 1: Your device is not producing an antifungal agent. Con =! 1: Use a constitutive promoter. PlasmidX56 =! 1: Incompatible plasmid chosen.	Custom errors associated with the project. Prevent completion if errors occur.
COMPLETION	TASK1: lm-def = 1. TASK2: PlasmidX56 = 1	A list of variable values required for project completion.
DESC	Design *A. chroococcum* to combat late potato blight Late potato blight is caused by fungus-like water molds from the genus Phytophthora. Water molds are oomycetes and technically not fungi, but protista.	The description of the project.
IMG	LVLART1.jpeg	Name of project image.
IMGDESC	A potato infected by a water mold.	Image caption.
AVATAR	IMAGEFILE.jpeg	Name of author image.
CHASSIS	*A.chroococcum_ATCC_4412*	Name of the chassis.
INPUTS	INPUT1 = Glucose	Inputs (if given).
ANALYSIS	SDS = sdsdata.jpeg, NANO = nanodata.jpeg, SPCTR = spectradata.jpeg	Names of the image files associated with analytical data.
HINTS	Task1: In the previous tutorial you built a constitutive protein generator. How did you do it?, Task2: The plasmid needs to be compatible with your chassis and the cutting sites must adhere to your design.	Hints the player may access using the hints button if they get stuck on a task.

The examples provided in Table 1. highlight what kind of data is currently associated with a single project. Many of these **keys** may have multiple **values** tied to them. In these instances, the text string is split using dots (.), commas (,) or underscores (_), to load more variables into a temporary file that is responsible for verifying that the players have succeeded with the required outcomes of a project.

More **key-value pairs** can be created into this structure whenever new features or level-specific requirements are needed to expand the educational breadth of the game. These may include additional requirements for level completion, such as logical conditions that need to be true or a template truth table, which can be included with the project data as a data sheet. Using these Boolean truth values provides the game with a set of rules the players' design must adhere to, thus allowing the creation of more complex projects.

As seen previously on Figure 14, the chassis section specifies which chassis is used in this project. If the players are required to use a specific chassis, the chassis is "locked" and can be only modified, not changed. In some projects the chassis is not predetermined, in which case the player may choose a chassis they believe would work well. Sometimes the BioBricks required to complete a project are functional only with a specific chassis.

The metadata section of the project screen was designed with custom projects, such as those created by educators or citizen science projects in mind. This allows the project author (typically an educator) to include an image representing the project. Image captions are accessible by hovering over the image. Additionally, the author can include details such as difficulty level, their name, and the date of creation.

The modifiers section contains specific requirements for the current project, allowing players to quickly understand what the project entails and what limitations are enforced. These modifiers are used to force the player to solve the problem within a given framework, introducing new angles of approach to problems of similar difficulty.

The tasks section contains all the individual tasks the players are required to complete to finish the project. The projects are split into multiple tasks for several reasons. This structure allows the players to tackle the project in whichever order they prefer. It also allows the game to track the players' progress more easily. The final and most important reason is to enable the players to work either in the design tool or the laboratory, with the game automatically transporting them to the appropriate game scene based on the task at hand.

The main programming structure of a project works as follows: Once the players have gained access to a project in the **Projects Overview** screen, they may then open this project. This triggers the **ProjectDataHandler**

script to read project details from the **ProjectData** file. The project details are then displayed on the **Project Screen** for the player. A simplified flowchart illustrating the process, from opening a project task to completing a project task can be seen in Figure 28.

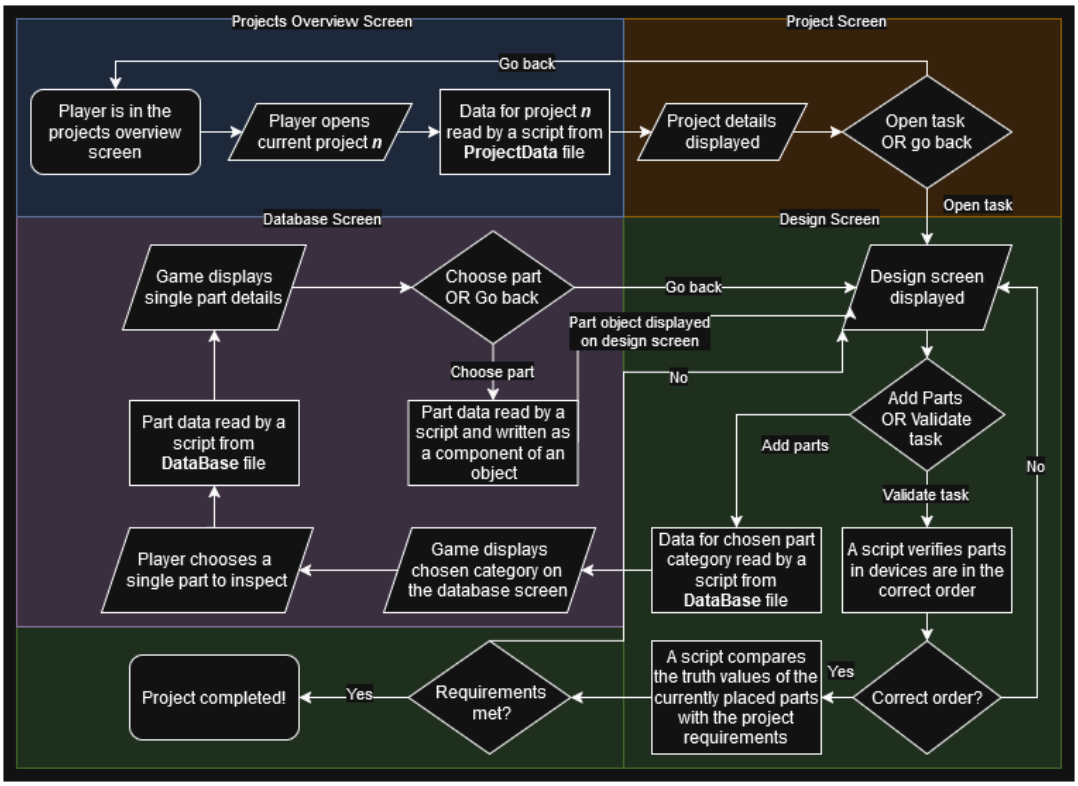

Figure 28. A simplified programming flowchart illustrating how a player completes a project. Diagram by Aarni Aspi.

When the player has selected a task, a temporary **TaskObject** class is created using a script to store the Boolean values from the **ProjectData** file. This data is stored as variables in an array, which is read periodically by the **ProjectDataHandler** script. All Boolean values associated with the parameters of the chosen chassis, as well as the parameters of BioBricks in the design area are also stored into the same **TaskObject** class.

The player may also change, remove or modify both the chassis and BioBricks, as well as turn inputs ON/OFF. All these actions, as well as the actions denoted in the flowchart, will modify the player's truth table, which is essentially the aforementioned array in the **TaskObject** class. The variables in the truth table are refreshed after every player input, allowing the truth table to update in real-time according to player actions.

The **Database** file is a nested JSON file. Main database categories are essentially a list of **key-value pairs**, just like the **ProjectData** entries. Each key has an array of data tied to it. Within an array, individual database entries are listed, with parameters in the form of **key-value pairs** tied to each entry. An example of these **key-value pairs** is shown in Figure 29. Additional examples of various individual database entries in JSON-format are available in Appendix B, that showcase further how these **key-value pairs** are stored.


```
{
  "part_name": "AraC_repressible_promoter",
  "part_id": "BBa_K206000",
  "nick": "pBAD",
  "subcat": "Inducible",
  "subcat2": "Cell_Signaling",
  "sequence": "acattgattatttgcacggcgtcacactttgctatgccatagcaagatagtccataagattagcgga
  "length": "130",
  "designers": "Amelia_Hardjasa",
  "_1": "Arabinose",
  "_0": "Glucose",
  "Chassis_comp": "AraC, CAP, cAMP",
  "description": "This promoter is in the OFF-state in the presence of just AraC.
},
```


Figure 29. A database entry for a promoter and parameters tied to it.

The most important lines in this JSON entry are the **_1** and **_0** keys. These keys serve different functions depending on the type of part they are associated with. In this example and other promoters, **_1** determines when the promoter is **on** and **_0** determines when it is off. In repressible promoters, such as pBAD, the on-off functionality is a bit more nuanced and production usually lies somewhere between 0-100%, but for the design projects it's better to simplify the functionality initially. Altering the ratio of glucose, arabinose, AraC, CAP and cAMP will become more relevant in the laboratory.

Another notable **key** is **Chassis_comp**. This key includes a list of text strings as values, which are checked individually by the game's programming to ensure that AraC, CAP and cAMP are available with the chosen chassis. This is done by simply comparing the list of strings associated with the **Chassis_comp** key of the chosen chassis and the parts being used in the current project to ensure compatibility.

The game verifies that BioBricks are compatible and in the correct order by using the **DeviceValidation** script. This script is used whenever the player creates a device to ensure its functionality, before it is added into the database. A flowchart illustrating the part order verification process can be seen in Figure 30.

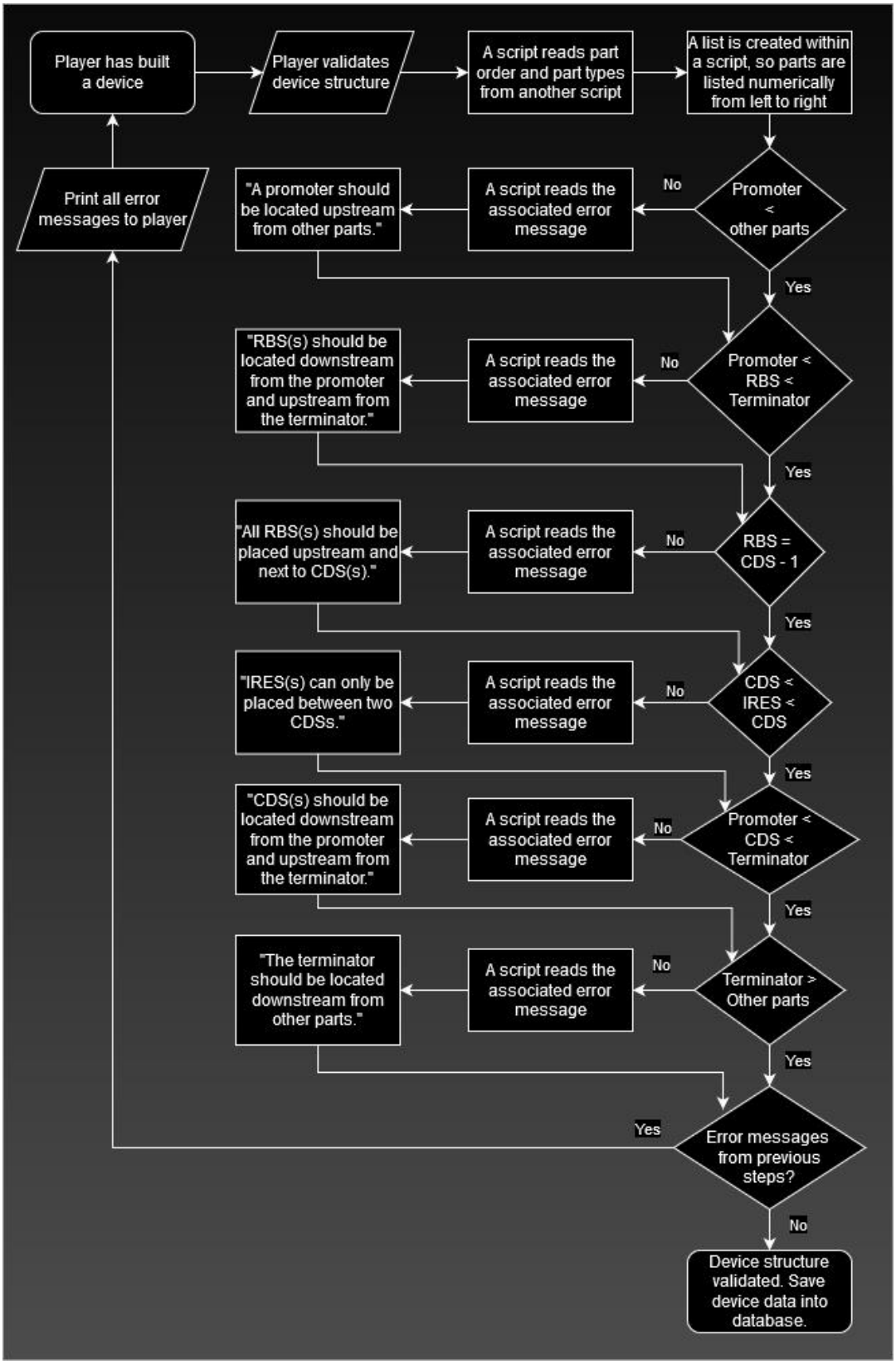

Figure 30. A programming flowchart illustrating the part order verification process. Diagram by Aarni Aspi.

As seen on Figure 30, the device verification process is quite simple. When parts are added in the design tool, the **DeviceValidation** script creates a list based on the part order the player has chosen, with a part type string (e.g. **RBS)** and its corresponding index in the list. Since each part now assigned a number, numerical **if** operations can now be computed based on the list. The flowchart above shows all the operations associated with part order verification, but a few additional operations are required to validate device functionality. These aren't in the flowchart due to space constraints.

These additional steps are tied to the number of part types in the chosen device. For example, if a device is missing a promoter, an RBS, a CDS or a terminator, an error message is created stating "**Part type** missing. A functional device requires a **part type** to **part function**". An additional check is performed to ensure that, for instance, the device does not contain multiple promoters or that the number of RBS + IRES matches the number of CDS. In reality, a device can have multiple promoters, but this is something that can be taught to the player in later stages of the game.

2.5 Facilitating Flow

As mentioned previously in chapter 1.2.2, the flow state enables the player to be highly focused and motivated by the activity they are engaging in. To facilitate flow in the game, various approaches have been implemented to increase player enjoyment, and to reduce frustration.

One important aspect of the flow state is immersion, a willing suspension of disbelief that allows the player to feel like they are within the game itself (Michailidis et al., 2018). The community aspect of SynBio, covered in chapter 2.3, was chosen as part of the game's scope not only for educational purposes but also to support immersion. The game's setting and story are the main elements of this, and they have been carefully designed to get the player invested in learning about SynBio.

The driving force of the story is the mystery of the missing professor. This not only gives the player an immediate sense of importance in helping the clients with their projects but also acts as a story hook that entices the player to keep going to find out what really happened. As the player progresses through the project, they will gradually uncover clues about the professor's whereabouts by investigating the premises with LAB.RR. Finding the professor will also require the player to demonstrate their understanding of SynBio, which serves as a motivating force for the player to not only to learn but also to understand SynBio in a more holistic sense.

The setting of the game grounds the player into the real-world, both simulating how SynBio is conducted in the real world and making them feel like they are a real intern working at an actual SynBio facility. The setting is enhanced not only by the players' actions within the game but also by the game's presentation. A screenshot of Aikio Labs, the location where the game takes place, can be seen on Figure 31.

Figure 31. A screenshot of the fictional Aikio Labs. Art by Alli Loikkanen.

Aikio Labs was designed very purposefully to support the themes of the game. Aikio labs is a self-sustaining solar, wind and bioenergy-powered cottage, containing a laboratory and a greenhouse full of synthetic biology experiments, an office and secret locations. Allowing the player to traverse between multiple spaces within the premises enhances immersion, making the player feel like they are in a space that could exist in the real world. All of these locations have been illustrated beautifully and they are accompanied by an ambient soundscape, using both visuals and sound to draw the player into the game's world. These spaces also have interactable background objects, as seen in figure 32.

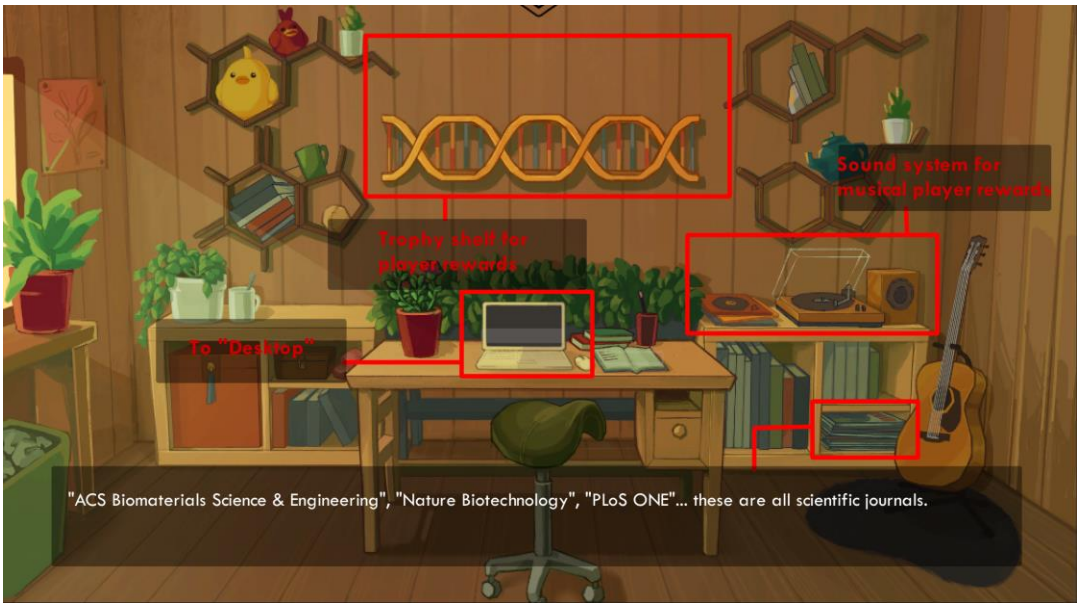

Figure 32. A screenshot of the office. Art by Alli Loikkanen.

Interacting with background objects gives the player a of pleasant surprise: a sound effect, an animation, a humorous monologue of what the player character sees from their point of view, or most importantly, relevant information that ties into the subject matter of SynBio in some way. This not only makes these spaces feel more real but also acts as another avenue for education or enjoyment.

Another source of enhancing the player experience is the player agency. Giving the player freedom both in when and how to approach the game allows them to experience it in a way they want to, and makes the game feel like it accounts for their actions (Cole and Gillies, 2021). Player agency has been considered when designing gameplay from the very beginning, and the player is given freedom in various modes of gameplay.

Since the projects in the game have been built to support multiple solutions, this allows the players to solve these projects in their own way, in addition to simulating how a singular outcome can be achieved in various ways in the real world. The desired outcome and limitations create a framework, but within this framework the players can express themselves freely.

Another source of player agency is dialogue choices, which allow the player to change the direction of a conversation with the various characters in the game. This comes in the form of reacting to what the characters are saying, or in the form of questions the player may present to these characters if they wish to do so. A screenshot of dialogue choices can be seen in Figure 33.

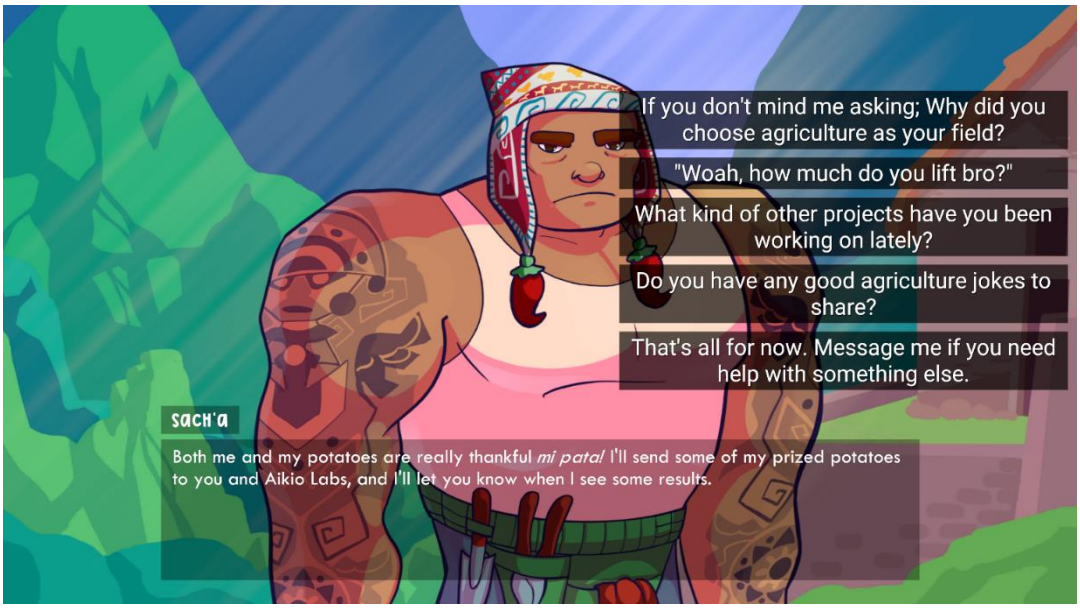

Figure 33. A screenshot illustrating choices the player can make in dialogue. Character art by Zsófia Kecskés, background art by Alli Loikkanen.

These questions allow the players to freely learn more about the subject matter or the characters themselves. This way, the players feel like they are willingly making a choice by asking specific questions and learning more, but in reality, the player is expected to ask all of these questions. If an option to interact presents itself, players often choose it just to see the outcome.

The glossary and introduction of the game also allow a greater degree of freedom for the players, enabling them to choose how much information they wish to be presented with. This is an especially valuable aspect of video games, as their interactive nature supports a wide audience, unlike traditional approaches that are directed towards a specific audience.

The most important factor in enhancing the play experience, however, is to make the game feel rewarding both extrinsically, and intrinsically (Johnson et al., 2018). This is achieved through both by actual in-game rewards given to the player after completing projects and using the game's presentation to make each interaction as rewarding as possible.

The in-game rewards, provided by the characters, push the player to persevere through the projects and make them feel like they have helped someone. After each project, the player is given a new set of questions they may present to the characters, allowing them to both learn more and get to know the characters better. After completing the first project for each client, the player also gains access to their profile pages, seen in Figure 34.

Figure 34. A screenshot of the client Natalia Paukova's profile page. Art by Zsófia Kecskés and Alli Loikkanen. UI art by Tuomas Hellsten.

These profile pages not only provide more information about the characters and the education and career paths they have taken, but also make these characters feel more like real people, encouraging the player to be more willing to help them with their projects.

Some other rewards, such as music albums and UI skins, allow the player to customize their play experience by changing the background music and UI of the design tool. Finally, after completing all projects from a single client, the player is rewarded with a SynBio-themed trophy. These trophies are displayed on a shelf in the office, seen previously on Figure 27, and the shelf will look incomplete unless the player gathers all the trophies. This helps motivate the player at the later stages of the game by allowing them to complete their collection.

When it comes to making each interaction feel rewarding using the game's presentation, this is mainly achieved through the use of sound effects and animations (Grimshaw et al., 2013; Praveen and Srinivasan, 2022). Whenever the player interacts with a BioBrick, a button, or other interactable objects, the action itself should have a sense of weight, and these repeated interactions should feel rewarding every time, so they don't get stale as the game progresses. This naturally leads into the next set of features that facilitate the flow state: features that avoid player frustration and fatigue.

One important aspect of avoiding player frustration is the game's difficulty curve (Pujol et al., 2024). The curriculum of the game was covered in more detail in chapter 2.4, and it was built with the difficulty curve in mind. Splitting the subject matter into chapters separated by tutorials helps pace the information into longer intervals, allowing the players to internalize it before the next tutorial.

The design tool also contains a hint button, allowing players to receive hints related to their current task. The hint button has been included in the game to avoid players getting stuck or frustrated during a task. For the purpose of learning about design, it does not matter if the player does it with help or not; they will remember what they did regardless and can utilize this information in future projects. Each task has three hints associated with it, and the hint button has a 5-minute timer between each hint. This way, the players are forced to think about the problem on their own as well, rather than relying solely on the hints.

Another approach to avoiding player frustration lies in the UI and user experience (UX) of the game. Whenever possible, the UI and UX of the game avoid repetition, and provide shortcuts to the player, allowing them to proceed without needing to do repetitive actions. For example, when the player chooses a task in the project screen that requires the laboratory, the game instantly transitions to the laboratory. This way, the player does not need to leave the computer and navigate to the laboratory separately.

The design tool has gone through multiple iterations to enhance the UI and UX during development, and this has significantly cut the time it takes to complete actions within it. According to early playtesting, reducing the number of interactions the player needs to perform to achieve their goal has cut the time it takes for a player to finish a design by around 50-75%. As development progresses, the UX and UI will continue being streamlined to make the game as efficient as possible.

Variety is also important in both keeping the player invested and avoiding fatigue. Variety is supported in the game by frequently switching the gameplay modes, alternating between the cognitively taxing projects and lighter activities, such as exploration, browsing the in-game database and conversations with the characters. This also enhances the pacing of the game, ensuring players aren't stuck doing the same thing repeatedly.

Finally, the game's overall presentation is especially important in facilitating flow. All the UI elements, backgrounds, character art, sound effects and music have been chosen both for aesthetic reasons, and to clearly communicate important information to the player. Music is especially effective at inducing

a flow state (Levy, 2015), and all the music for the design tool has been chosen with this in mind.

These examples should clearly show that InGENEous has been crafted with the facilitation of the flow state in mind. However, once the test version of the game is released to play testers, player feedback will ultimately provide evidence of whether the game succeeds in keeping the player invested or not.

More details about the game and its presentation are available in Appendices B and C.

3 Results & Discussion

Initially, this chapter was supposed to cover a pre-game and post-game survey presented to the game testers, as well as an analysis of the survey data. The first section of both surveys would have gathered information about the playtesters themselves, such as their age, occupation, education level, and their interest and experience in the game's subject matter and video games in general, both before and after playing the game. This section of both surveys has been completed and will be used when playtesting begins.

The second section of these surveys would have also tested the players' knowledge of the game's educational material both before, and after playing the game using a quiz, quantifying how well the game performed in its educational goals across a wide variety of testers. An initial set of 20 quiz questions related to BioBricks was created, but these require further refining since the content of these questions should relate exactly to the materials taught in the test version.

Additionally, the post-game survey contains a feedback form that allows the players to report problems with the game, as well as suggest improvements for the game, offering direction and good ideas for future iterations of the game. This feedback form has also been finished and will be used in the post-game survey of the test version.

The test version of the game is currently unfinished, however, and this chapter will instead focus on lessons learned about the development process itself, the current state of the game's prototype, how the framework of the game supports scientific purposes beyond education, and other useful results.

3.1 Game Development Post-mortem

The initial development work on InGENEous began with a team of three people to create a simple test version of the game in 5 months, which would have included a few projects the players could complete using the design tool. This would have allowed the game to be tested, and data to be gathered from playtesters using surveys to see how well the game succeeds in its educational goals.

As the development of the test version progressed, my enthusiasm for the game project took over and the design of the game exceeded its initial scope significantly, despite repeated reminders from my supervisor/advisor Prof. Merja Penttilä to keep the work focused and within reasonable timeframes.

This led to delays in development and as a result, the test version of the game is still unfinished.

Although the grown scope of the game is the main reason for the delays in development, a few other key reasons are important to highlight as well, allowing this game development post-mortem to help future pioneers in educational video game projects avoid similar pitfalls in game development. Identifying these reasons is important for continuing this game project as well, since all of these factors can be taken into consideration in project management in the future.

After extensive discussion with other team members about the development process, a handful of reasons were identified as the core issues that hindered the development:

- **Overly Ambitious Scope**: The project's scope was ambitious for a master's thesis to begin with, and my inability to stop learning more about SynBio and designing new features to incorporate these insights further exacerbated this issue. This led to multiple redesigns of the game's UI and screens, significantly extending the development process. Doing this work early will however benefit development in the future, since the features of the game are built on a more robust foundation.

- **Underestimated Development Time**: My initial familiarity with SynBio, awareness of online databases, and the seemingly straightforward game design of InGENEous gave me a false sense of optimism about the time required to develop a test version of the game. This is a common problem in most software development however, since all problems cannot be predicted (Mancas, 2014).

- **Excessive Drive**: The collective excitement of the team to release a more substantial test version pushed us to exceed our mental and physical limits, leading to significant fatigue. Multiple team members have had to take breaks from development to recover from periods of excessive work. This problem is persistent and prevalent across the game industry (Bourscheid and Steil, 2024; Escudero, 2008) and is often discussed at game industry events and among industry veterans (Moberly, 2024).

- **Constraints of Educational Game Development**: Unlike traditional game development, where design can adapt to programming constraints, the need for scientific accuracy in an educational game meant programming shortcuts could not be taken. Educational games

are more challenging to develop than normal games in general due to more limitations on development (Ghosh, 2022).

- **Limited Programming Resources**: While Toni Saarela has done extraordinary work with the game's programming, the initial plan for the game included me in game engine programming as well. Due to the larger scope and team size, I was unable to contribute as much as initially planned to game engine programming since I needed to focus on project management and creating the educational framework of the game.

Overall, the development process has been quite difficult when considering reasonable timetables and a variety of issues have delayed the project. Once the development of the game continues, however, these issues will be considered in project management, and the increased scope of the game will provide excellent foundations for an educational game with considerable impact.

3.2 Current State of the Game

At present, the test version of the game remains incomplete. The game currently comprises approximately 50000-60000 lines of code contributed by the development team. However, addressing existing bugs and implementing entirely new features will require tens of thousands of additional lines of code to bring the game to completion.

The current prototype of the game has a partially working in-game database, which allows players to browse and place BioBricks in the design tool, select a chassis, and inspect transcriptional factors and external inputs associated with the regulatory promoters currently available. Players can also modify the chosen chassis using 20 template devices. These template devices, which are constitutive protein generators, enable regulatory promoters to function by producing the necessary transcriptional factors, surface proteins or messenger molecules.

Overall, the current database offers players 14 chassis options and 45 core BioBricks to work with. Additionally, devices created by players are saved into the database, allowing them to be reused as templates in the future projects. While all other database categories and their descriptions have been designed and written, they currently lack entries and they have not yet been programmed. These categories will become relevant once players has gained access to the laboratory.

The design tool has undergone multiple overhauls for each view, but the current in-game design tool has not been updated with these improvements yet. Despite this, the current design tool possesses several functional features. It is capable of validating whether BioBricks are in the correct order and compatible with each other. Additionally, external inputs can be toggled on or off, and the integrated truth table enables comparison of Boolean values with project requirements, allowing players to successfully complete projects. These functionalities operate across the device, circuit and gate views. However, the system view is still under development. The goal is to make the systems horizontally scrollable, while keeping vertical layout compact to ensure clear text visibility.

The automatic line-drawing feature, which connects internal inputs, external inputs, outputs and devices, currently works with the older version of the design tool. This feature will need to be rebuilt to align with the overhauled design tool. Fortunately, this task is straightforward since the new design tool is vertically locked, eliminating the need to scale lines based on the zoom level.

The sequence view is currently accessible for individual BioBricks, but its functionality remains limited. While players can scroll through the sequence and select individual bases, buttons are not yet operational. Additionally, the sequence view does not currently allow players to modify DNA sequences or mark scars. At this stage, the sequence view is also not functional for devices, circuits, gates or systems. However, implementing these features is relatively straightforward as the sequence view is essentially a glorified text editor. Enhancing this feature will be a priority as development progresses to ensure a seamless and intuitive user experience.

The laboratory has been partially designed, and both the programming structure and features have been planned for the bioreactor, PCR and SDS-Page. These features weren't part of the test version's scope, however, so they haven't been prioritized. Further examination is needed to determine what should be included in the laboratory, in consideration to the available development resources and time.

The glossary function and documentation are considered to be core features, since all text in the game has been written with these features in mind. They have been designed but haven't been implemented yet due to the complexity of text implementation in the Unity-engine. However, a solution for these features has been designed, and the upcoming test version of the game should include these features, ensuring that the massive amounts of text in the game do not need to be re-written.

When it comes to in-game text, the game currently includes the introduction and all of its chapters, first 3 dialogues with each client and LAB.RR, the character bios, descriptions for database categories and existing parts in the database, 3 tutorials, dialogue associated with the interactable objects, and an ending scene for the test version of the game. The game still requires more tutorials, part descriptions and character dialogues, but these should be made in association with the projects themselves.

The game's presentation, however, is nearly finished, at least in consideration of the current scope of the game. Most in-game music, sound effects, art and UI elements have been produced, and these assets can be reused in various ways to create more educational and fun content for the game. When development of the game continues, the presentation of the game does not require considerable development time and direction anymore and the focus is now solely on bringing functionality to everything.

3.3 InGENEous as a Tool for Science

The game was designed with various online databases, repositories and tools in mind, including SynBioHub, the iGEM Parts Repository and the SBOL standard glyphs for BioBricks. Originally, the plan was to scrape all the data from the iGEM Parts Repository and use it in the game. However, in the end, an in-game database was created to provide the players a simpler, smaller and more standardized library of curated BioBricks from the iGEM Parts Repository instead. The iGEM Parts Repository is also currently being reworked, which would have hindered the development of the game (iGEM, 2024c).

Since the BioBricks in the game's database have a substantial amount of data associated with them, and the game verifies the part order and compatibility whenever the player builds a device, this allows us to leverage the game's programming in a plethora of interesting ways to benefit researchers in the field. Once this potential became apparent, all further development of the game was done with these features in mind.

Free Draw Mode

The first of these features was to include a free draw mode into the game, allowing players or researchers to use BioBricks and other glyphs from the database to make high quality visual diagrams of their devices, circuits, logic gates and systems for their own use. Since the game already has data associated with the BioBricks, players can opt to display information about the chassis, BioBrick codes, names or abbreviations of their parts.

They can also display their chosen inputs, required transcriptional factors, and desired outputs and show how these interact with their design using the game's automated line drawing feature. This also includes the truth table, which the player can export as a summary of the logic associated with their designs. Once the player has chosen what they want to display with their design, the free draw mode allows the player to export a high-quality image out of the game.

Data Exporting Function

The players can also export all the data associated with their designs as a set of parameters in data sheets. This includes all the parameters associated with each BioBrick in their design, as well as the chassis. The player can also export their truth tables as data sheets.

The BioBricks within the iGEM Parts Repository have a certain degree of standardization, with individual entries typically including the DNA sequence, scars, length designers etc. However, there are no strict rules on the inclusion of details. As a result, some BioBricks may only include a sequence, name and short description while others may have extensive descriptions, details about the structure and other important factors to consider.

When a player creates a device in the game and saves it into the game's database, most of the details and metadata associated with the device can be automatically included through the game's programming. This includes the sequence, length, name, abbreviation, type etc. If this data could then be exported from the game and imported into a separate scientific database, such as the iGEM Parts Repository, it could serve as an additional step of standardization, automatically filling in the missing information.

Most of these details can be pulled from the database itself, such as the sequence, length, scars, type, design date, chassis compatibility, RCA assembly standard, and a short description. I also developed a way to automate naming and abbreviation for devices. However, this standardization does not extend to circuits, logic gates or systems. The reasoning behind this decision was to make it possible to identify what a device does based on its name or abbreviation alone, without needing to refer to an online source based on a BioBrick code.

Whenever the player creates a device in the game, it is named using the following template:

(CELL TYPE) + STRENGTH + PROMOTER + OUTPUT + DEVICETYPE

Which will then output device names such as these:

(P)strong constitutive LacI-generator
(E)medium-strong TetR-repressible eGFP-reporter
(P) Lead-inducible mCherry -measurement device

P stands for prokaryote, and E for eukaryote in these names. As an additional note, the word protein is not used with protein generators, as in the example "LacI-generator". These devices are categorized as protein generators in the database however, and this naming convention is used to shorten the length of the text in names.

The game also generates an abbreviation for devices, which was originally intended to be displayed under each device in the design tool. However, this was replaced with the display of the selected part's name in the selection window. These abbreviations can still be used in the free drawing tool if the player prefers not to use or does not have the BioBrick codes for their parts. These abbreviations follow a similar structure:

(CELL TYPE) + STRENGTH + PROMOTER + OUTPUT + DEVICETYPE

Which outputs abbreviations such as these:

(P)sCon_LacI_Gen
(E)ms_TetR(r)_eGFP_Rep
(P)m_Pb2+(i)_mCherry_Md

These naming conventions are still a work in progress, but as the development of the game continues, I will strive to improve these templates to ensure they function effectively. Regardless, the current naming structure provides a good starting point and will be refined further based on feedback and testing. Some initial concepts of data standardization and where this data could be sourced from are available in Table 2.

Table 2. Examples of exportable data.

Metadata	**Source**	**Example**
BioBrick Code	Game, iGEM, Author	BBa_J11000 (if made by an iGEM-team) BBa_Zxxxxxx (game template code) BBa_Zyyyyy (if made by iGEM-template)
Name	Game	BioBrick: LacI-repressible promoter BioDevice: Strong constitutive lacI-generator BioCircuit: Lead-regulated eGFP-reporter BioGate: Hrp[R+S]-Regulated AND-gate
Abbreviation	Game, iGEM, Author	BioBrick: pLac BioDevice: (P)sCon_LacI_Gen BioCircuit: (P)sPb2+Ind_eGFP_Rep BioGate: (P)sInput1Input2_Output_Gatetype
Sequence	Game	TAAGCATCGACAGCGTAGCTAGC…
Length	Game	3561 bp
Scars	Game	17-28, 670-692 (scar base-pair index numbers)
Plasmid	Game, iGEM	x-57
Type	Game	BioBricks/CDS/Biosynthesis BioDevices/Protein Generator BioGates/AND-gates
Designer(s)	Game, Author	Billy Bacterium, Fiona Fungus, Parthus Prokaryotus
Team	Author	Team Aalto-HY 2024
Design Date	Game	27/09/2024
Upload Date	Author, iGEM	12/10/2024
Chassis Comp.	Game	Designed for: *E.coli strain KRX* Predicted Compatibility: Prokaryotes Tested Compatibility: *E.coli strain KRX*
Burden	Author	*Requires testing in laboratory conditions*
Categories	Game, iGEM	BioBricks/CDS/Lysis/ **or** Devices/Reporters/
RCA Standard	Game, Author	The template RCA in the game is RCA10. Author can change RCA in iGEM parts.
Description	Game, Author	Short description from game. More details can be added by Author on iGEM.

Since database entries can have any number of parameters tied to them, the amount of data that can be stored with the parts is essentially limited only by the data structures within a nested JSON file. However, if larger quantities of data in the form of arrays or matrices need to be stored, the database will require a different data structure altogether to handle this more efficiently.

Automated Search Engine

While developing the game, an idea for automating device design emerged, and a piece of code was written to support this function. This feature allows players to define a desired output and input, after which the game assembles each combination of BioBricks that achieves this goal while filtering out results where the part order is incorrect. This process generates a list of all part combinations that fulfill the given parameters with thousands of devices listed.

If further developed, this could evolve into an automated search engine that leverages the part order verification script of the game and the database to automate the design process. Additional parameters could be incorporated to refine the search. For example, players could defined a chosen chassis or a chassis type, which would filter out all incompatible BioBricks. Specifying a desired expression strength could filter out a selection of RBS and terminator choices, and choosing if the device is constitutive or regulated would eliminate undesirable promoters. Additionally, the CDS choice could be determined by defining either a specific output, or a category of biomolecules, such as reporters etc.

With further definition of parameters, the output of the design could be structured as logical arguments, enabling this tool to be used for creating circuits, logic gates and systems as well. With enough parameters, this feature could be used to provide the player with a strong baseline template for the desired functionality of their design, allowing them to focus more on building, testing and learning how to modify this initial template “backbone”.

However, this functionality is not intended for use within the game itself, as the game’s goal is to teach the players how to independently complete the design process. Additionally, the game’s database is relatively small and curated with the specific game projects in mind, which limits the possible outcomes of this search engine. That said, if this code were repurposed and implemented in HTML5, it could become a valuable tool of various online databases, such as the iGEM Parts repository. This would allow users to save significant amounts of time during the design process, and could potentially open up additional applications in the future.

Citizen Science Projects

The project structure and creation process was covered previously in chapter 2.4.1. Currently, projects are created by filling **key-value pairs** into the **ProjectData** file. This isn't, however, the final vision on how projects would be created for the game.

Since the game should allow educators to create their own projects, a fully integrated project editor should be included with the game. This project editor will allow the author of the project to either input **values** to **key-value pairs** or choose values out of dropdown menus when applicable. Images, such as project, author and analytical data images, can be added to the project as well, and the project author should be able to create a set of requirements the player needs to pass to complete the project, one task at a time.

The project editor should also allow project authors to add their own BioBricks, chasses, plasmids etc. into the in-game database. This way, if a player opens a custom project, all the required parts will be added into the player's database to allow the completion of a project.

This custom project creation would facilitate citizen science projects as well, since researchers could add designs that require optimization as custom projects into the game. Players would receive a pre-made design to optimize, and their designs could be scored based on a set of metrics, such as the number of parts used in the design, RCA-standard compatibility, orthogonality, burden etc. If a player-made design manages to score higher in these metrics than the current top design, this player-made design could then replace the design offered to players in this current project, similarly to how an algorithm is used to keep the players up to date on the most optimized design yet in **Phylo** (Kwak et al., 2013; Singh et al., 2017).

This feature would require an online server for the game to store all of these citizen science projects and player-made designs, but it could provide researchers with a useful avenue for harnessing the player's design-skills for scientific purposes. Players could also continue playing the game after its completion, and competitive elements could be introduced as an additional motivation for the players.

3.4 Other Contributions

The development process of the game has also led to other results that are harder to categorize, but I consider their inclusion here valuable, nonetheless. As an additional note on the introduction scene; the plan is to create an

informational video based on it. In this video, the text and player interactivity from the introduction scene will be replaced with subtitles, full animations and narration. Since the introduction of the game spans from prehistoric life to BioBricks, this video could be used to offer the viewer the necessary background information required to understand SynBio and how BioBricks function based on their DNA-sequences. This video could be used for educational purposes, as well as for promoting SynBio.

The glyphs and icons in the design tool and database were based on the SBOL Standard Visual glyphs and icons used in the iGEM Part Repository. These original glyphs and icons were quite small and had a low resolution. Since the game needs to scale for different resolutions and monitor sizes, we remade these glyphs and icons as high resolution, transparent image files. Once all these glyphs and icons are finished, they will be released alongside the game into the public domain under the CC0 license, allowing anyone to freely use these glyphs commercially or non-commercially without attribution, allowing the SynBio community to benefit from this work outside the game.

While on the topic of glyphs, I realized during development that BioBricks for internal ribosome entry sites (IRESs) have used the same glyphs as RBSs so far. When multiple proteins are produced under a single promoter in prokaryotes, an RBS is required upstream from each CDS. In eukaryotes however, IRESs are required between each CDS for successful gene expression. To address this, I designed a new glyph to be used with IRESs, enabling the viewer to distinguish whether a device is intended for a prokaryote or eukaryote by the glyphs alone. This new glyph can be seen alongside with the original glyph for RBS in figure 35.

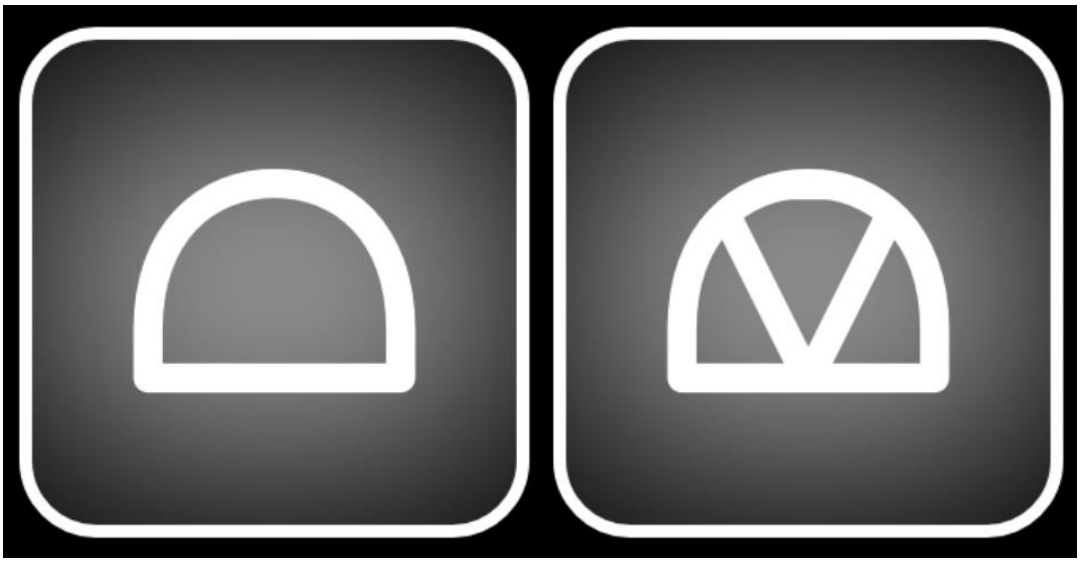

Figure 35. RBS on the left, IRES on the right. Glyphs by Julia Rässa, IRES design by Aarni Aspi.

During development, I also came up with some new language and puns related to SynBio. While these are not scientifically accurate terms, they add value by expanding the language spoken among SynBio researchers. For example, the name of the game, InGENEous, is a play on an existing word in the English language, ingenious, which is defined as “having or showing an unusual aptitude for discovering, inventing or contriving”. Since the purpose of the game is to make the player an ingenious SynBio enthusiast, this name is quite fitting. The same applies to EnGENEer, which is a play on the word engineer.

Other fun word associations include the name of the communication platform, Forum Quorum. This is a play on the historical Forum Romanum Plaza in Rome, which served as a public center for social gatherings of all kinds. Quorum comes from quorum sensing, which in biology refers to intercellular communication. This fits perfectly for a communication platform that is related to synthetic biology.

A few other words came to mind during the process of writing this thesis. These include SynBiosis, a play on the word symbiosis and SynBio and SynBiosium, a play on the word symposium and SynBio. Both words, along with Forum Quorum could be used as a name for a SynBio conference, and I hope to see this become reality during my lifetime.

4 Conclusions

The original aim of this thesis was to produce a simple educational game about SynBio and then test how well the players learned the subject through the game and how much the video game approach increased their enjoyment of the learning process. This became unfeasible, however, so the subject and approach of this thesis was pivoted into a new perspective, along with new research questions. Considering these radical changes to this thesis, it still manages to provide something technically worthwhile and useful for science education.

Regarding the first set of research questions associated with the literature reviews, this thesis adequately addresses these questions and topics. The literature review defines what SynBio is and how it relates to other biosciences, while offering a glimpse on how research is done in the field through the use of databases, models and the assembly of synthetic devices. BioBricks and their use cases are also adequately defined, and examples of BioBrick-based applications are provided through the lens of iGEM.

The literature review on games defines what games are and how they function, while also identifying the flow state as a common aspect of games, that allows the player to progress in a highly focused and motivated state. The literature review also highlights the global need for interactive digital education, and the examples and definitions of various types of existing digital education provide different approaches on how games and game-related media is used for science and education.

The second set of research questions related to the development of the game is also covered in detail. The second chapter discusses various aspects of SynBio and how they could be taught using a video game. These aspects can be covered well via various gameplay loops, but they can additionally enhance the play experience by providing an interesting and unique setting. The setting enhances player immersion, and allowing the player to design useful real-world applications on a tool that could potentially be used by researchers to design with BioBricks gives a real sense of weight to the player's actions.

The second chapter also effectively covers how the game’s features facilitate the flow state in various ways. By focusing the game design of an educational game on both enhancing player enjoyment and minimizing frustration, it can provide a great platform for focused and enjoyable education.

The third chapter offers insight into the production of the game itself and subsequently provides examples on challenges associated with an iterative development process. The development of InGENEous has been arduous due to its educational nature, and the growing scope of the game certainly has made its development in a rapid and efficient manner quite difficult. The use of games in education is becoming more established across various fields, and further research into their development might provide useful insights into efficient production approaches.

Additionally, the third chapter provides a variety of examples on how the programming and structure of InGENEous can be used for scientific purposes as well. These include citizen science initiatives, functionality as a design tool with accurate data, and repurposing some of the game's programming to create an automated search tool. If these features are developed further, they could provide researchers with a valuable software tool for SynBio, with various useful functions.

4.1 Outlook

The prospects of InGENEous as a platform for teaching SynBio are promising and continued work on this game is essential to fulfil its ambitious scope and practicality as a tool for education and science. A working group that incorporates multiple programmers is required to complete a game of this scope, as the programming workload is the most significant hurdle opposing the future use of InGENEous. The game project and its continued development also has potential in expanding into a full doctoral thesis, including the publication of articles focusing on the game's development process, results of player surveys and the game's use for scientific purposes.

The next natural step for the game's development itself should involve transitioning the current prototype from Unity 2022 to Unity 6. The new Unity 6 version offers significant advantages, including a more efficient production pipeline for programming, user interface (UI) design, and in-engine animations. It also provides extended support, ensuring better compatibility with future extensions, plugins, and operating systems in the future. Additionally, runtime fees were included into Unity 2022 after the production of InGENEous began, which might have endangered the free and accessible nature of InGENEous. Runtime fees are fortunately not a part of Unity 6, ensuring the game would stay free for all users in its finished state. (Unity, 2024)

During this transition, the game's code should also be streamlined, improving its maintainability and accessibility for external developers, which would in turn enhance its status as a piece of open-source software. Once this transition is complete, the first test version of the game should be finalized and it should include projects ranging from simple devices to more complex systems, utilizing the design tool to of the "design" stage within the DBTL (Design-Build-Test-Learn) cycle. Gathering player feedback during this phase will be crucial for refining the "design" stage and "build," "test," and "learn" stages of the DBTL cycle in further iterations of the game.

Following the first test version of the game and the integration of player feedback, the development of the game should proceed into additional development-testing cycles, ultimately leading to a full release of an educational game about SynBio. After the game's release, further consideration should be given to introducing features that would potentially enable its use as a tool for research and citizen science initiatives as well, while potentially expanding the educational aspects of the game further.

References

Ambron, E., Buxbaum, L. J., Miller, A., Stoll, H., Kuchenbecker, K. J. & Coslett, H. B. (2021). *Virtual reality treatment displaying the missing leg improves phantom limb pain: A small clinical trial*, Neurorehabilitation and Neural Repair, 35(12), pp. 1100–1111. https://doi.org/10.1177/15459683211054164

Ambron, E., Miller, A., Kuchenbecker, K. J., Buxbaum, L. J. & Coslett, H. B. (2018). *Immersive Low-Cost Virtual Reality Treatment for Phantom Limb Pain: Evidence from Two Cases*, Frontiers in Neurology, 9(67), pp. 1-7. https://doi.org/10.3389/fneur.2018.00067

Ausländer, S., Ausländer, D. & Fussenegger, M. (2017). *Synthetic Biology — The Synthesis of Biology,* Angewandte Chemie International Edition English, 56(19), pp. 6396–6419. https://doi.org/10.1002/anie.201609229

Averell, L. & Heathcote, A. (2011). *The form of the forgetting curve and the fate of memories,* Journal of Mathematical Psychology, 55(1), pp. 25–35. https://doi.org/10.1016/j.jmp.2010.08.009

Bailey, C. D. (1989). *Forgetting and the Learning Curve: A Laboratory Study,* Management Science, 35(3), pp. 340–352. https://doi.org/10.1287/mnsc.35.3.340

Bamparopoulos, G., Konstantinidis, E., Bratsas, C. & Bamidis, P. D. (2016). *Towards exergaming commons: composing the exergame ontology for publishing open game data*, Journal of Biomedical Semantics, 7(1), 4. https://doi.org/10.1186/s13326-016-0046-4

Barone, J., Bayer, C., Copley, R., Barlow, N., Burns, M., Rao, S., Seelig, G., Popovic, Z. & Cooper, S. (2015). *Nanocrafter: Design and Evaluation of a DNA Nanotechnology Game*, In: Proceedings of the 10th International Conference on Foundations of Digital Games, Pacific Grove, CA, USA. Available at: https://www.khoury.northeastern.edu/home/scooper/index_files/pub/barone2015nanocrafter.pdf [Accessed 16 January 2025].

Bitzer, D., Braunfeld, P. & Lichtenberger, W. (1961). *PLATO: An Automatic Teaching Device*, IRE Transactions on Education, 4(3), pp. 157–161. https://doi.org/10.1109/TE.1961.4322215

Bohannon, J. (2008). *Flunking Spore*, Science, 322(5901), pp. 531–531. https://doi.org/10.1126/science.322.5901.531b

Bourscheid, D. & Steil, A. (2024). *Crunch on the playground: long hours of work in game development*. In: Proceedings of the DiGRA 2024 Conference: Playgrounds, Guadalajara, Jalico, Mexico. Available at: https://dl.digra.org/index.php/dl/article/view/2343/2340 [Accessed 16 January 2025].

Bouvier, P., Lavouë, Ë., Sehaba, K. & George, S. (2013). I*dentifying Learner's Engagement in Learning Games - A Qualitative Approach based on Learner's Traces of Interaction*. In: Proceedings of the 5th International Conference on Computer Supported Education, SciTePress - Science and and Technology Publications, Aachen, Germany, pp. 339–350. https://doi.org/10.5220/0004386903390350

Bouvier, P., Sehaba, K. & Lavoué, É. (2014). *A trace-based approach to identifying users' engagement and qualifying their engaged-behaviours ,in interactive systems: application to a social game*. User Modeling and User-Adapted Interaction, 24(4), 413–451. https://doi.org/10.1007/s11257-014-9150-2

Caillois, R. (1961). *Man, play, and games*. Translated by M. Barash, Free Press of Glencoe, New York. (Original work published in 1958 by Gallimard, Paris, as *Les jeux et les hommes*). Available at: https://voidnetwork.gr/wp-content/uploads/2016/09/Man-Play-and-Games-by-Roger-Caillois.pdf [Accessed 16 January 2025].

Caillois, R. (1955). *The Structure and Classification of Games*, Diogenes, 3(1), pp. 62–75. https://doi.org/10.1177/039219215500301204

Cameron, D. E., Bashor, C. J. & Collins, J. J. (2014). *A brief history of synthetic biology*, Nature Reviews Microbiology, 12(7), pp. 381–390. https://doi.org/10.1038/nrmicro3239

Carl, J. (2009). *Industrialization and Public Education: Social Cohesion and Social Stratification*. In: Cowen, R., Kazamias, A.M. (Eds.), International Handbook of Comparative Education. Springer Netherlands, Dordrecht, pp. 503–518. https://doi.org/10.1007/978-1-4020-6403-6_32

Carter, M. & Gibbs, M. R. (2013). *eSports in EVE Online: Skullduggery, fair play and acceptability in an unbounded competition*. In: Proceedings of the 8th International Conference on Foundations of Digital Games, SASDG, pp. 47-54. Available at: https://www.academia.edu/70081742/eSports_in_EVE_Online_Skullduggery_Fair_Play_and_Acceptability_in_an_Unbounded_Competition [Accessed 6 February 2025].

Cazier, A. P. & Blazeck, J. (2021). *Advances in promoter engineering: Novel applications and predefined transcriptional control*. Biotechnology Journal, 16(10), 2100239. https://doi.org/10.1002/biot.202100239

Chen, J. (2007). *Flow in games (and everything else)*, Communications of the ACM, 50(4), pp. 31–34. https://doi.org/10.1145/1232743.1232769

Cole, T. & Gillies, M. (2021). *Thinking and Doing: Challenge, Agency, and the Eudaimonic Experience in Video Games*, Games and Culture, 16(3), pp. 187–207. https://doi.org/10.1177/1555412019881536

Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., Leaver-Fay, A., Baker, D., Popović, Z. & Players, F. (2010). *Predicting protein structures with a multiplayer online game*, Nature, 466, pp. 756–760. https://doi.org/10.1038/nature09304

Dede, C. (2009). *Immersive interfaces for engagement and learning*, Science, 323(5910), pp. 66–69. https://doi.org/10.1126/science.1167311

Diercks, C. S., Dik, D. A. & Schultz, P. G. (2021). *Adding new chemistries to the central dogma of molecular biology*, Chem, 7(12), pp. 2883–2895. https://doi.org/10.1016/j.chempr.2021.09.014

Djaouti, D., Alvarez, J., Jessel, J. -P. & Rampnoux, O. (2011). *Origins of Serious Games*. In: Ma, M., Oikonomou, A., Jain, L. C. (Eds.), Serious Games and Edutainment Applications, Springer, London, pp. 25–43. https://doi.org/10.1007/978-1-4471-2161-9_3

Dolan, S. K. & Matilla, M. A. (2022). *Tools of the trade: plasmid repositories and standardized plasmid manipulation for molecular and synthetic biology*, Microbial Biotechnology 15(5), pp. 1318–1320. https://doi.org/10.1111/1751-7915.14012

Dong, J., Zhou, C. & Wang, Q. (2020). *Towards active self-assembly through DNA nanotechnology*. In: Fan, C., Ke, Y. (Eds.), DNA Nanotechnology: From Structure to Functionality. Springer International Publishing, Cham, pp. 1–25. https://doi.org/10.1007/978-3-030-54806-3_1

Dunlap, P. & Pecore, J. L. 2009. *The effects of gaming on middle and high school biology students' content knowledge and attitudes toward science*. Studies in Teaching Research Digest, pp. 19–36. Available at: https://files.eric.ed.gov/fulltext/ED508150.pdf [Accessed 16 January 2025].

Escudero, A. (2008). *Working Smarter, Not Harder - Achieving work-life balance in the Video Games Industry. Available at:* https://doi.org/10.13140/RG.2.2.17380.07046

Ferguson, R., Barzilai, S., Ben-Zvi, D., Chinn, C., Herodotou, C., Hod, Y., Kali, Y., Kukulska-Hulme, A., Kupermintz, H., Mcandrew, P., Rienties, B., Sagy, O., Scanlon, E., Sharples, M., Weller, M. & Whitelock, D. (2017). *Innovating Pedagogy 2017: Exploring new forms of teaching, learning and assessment, to guide educators and policy makers*. Available at: https://iet.open.ac.uk/files/innovating-pedagogy-2017.pdf [Accessed 16 January 2025].

de Freitas, S. & Maharg, P. (2011). *Introduction: Digital games and learning*, Digital Games and Learning, pp. 1–16, Continuum Publishing, London. Available at: https://www.researchgate.net/publication/288846345_Introduction_Digital_games_and_learning [Accessed 6 February 2025].

García-Granados, R., Lerma-Escalera, J. A. & Morones-Ramírez, J. R. (2019). *Metabolic Engineering and Synthetic Biology: Synergies, Future, and Challenges*, Frontiers in Bioengineering and Biotechnology, 7, 36. https://doi.org/10.3389/fbioe.2019.00036

Ghosh, A. (2022). *Challenges in Educational Game Development*, International Journal of Darshan Institute on Engineering Research & Emerging Technology, 11(1), pp. 54–60. https://doi.org/10.32692/IJDI-ERET/11.1.2022.2209

Goujet, R. (2018). *Hero.coli : a video game empowering stealth learning of synthetic biology : a continuous analytics-driven game design approach. Hero.coli, un jeu vidéo d'apprentissage furtif de la biologie de synthèse : une méthodologie de conception de jeu guidée par l'analyse continue de données*. PhD thesis, Université Sorbonne Paris Cité. Available at: https://theses.hal.science/tel-02524484 [Accessed 16 January 2025].

Grimshaw, M., Tan, S. -L. & Lipscomb, S. D. (2013). *Playing with sound: The role of music and sound effects in gaming*. In: Tan, S. -L., Cohen, A. J., Lipscomb, S. D., Kendall, R. A. (Eds.), The Psychology of Music in Multimedia. Oxford University Press, Oxford, The United Kingdom. https://doi.org/10.1093/acprof:oso/9780199608157.003.0013

Guardiola, E. (2016). *The Gameplay Loop: a Player Activity Model for Game Design and Analysis*. In: Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology, pp. 1–7. https://doi.org/10.1145/3001773.3001791

Guo, J. (2014). *Transcription: the epicenter of gene expression*. Journal of Zheijiang University Science B, 15, pp. 409–411. https://doi.org/10.1631/jzus.b1400113

Gurdo, N., Volke, D. C., McCloskey, D. & Nikel, P. I. (2023). *Automating the design-build-test-learn cycle towards next-generation bacterial cell factories*, New Biotechnology, 74, pp. 1–15. https://doi.org/10.1016/j.nbt.2023.01.002

Hali, K., Manzo, M. A., Koucheki, R., Wunder, J. S., Jenkinson, R. J., Mayo, A. L., Ferguson, P. C. & Lex, J.R. (2024). *Use of virtual reality for the management of phantom limb pain: a systematic review*. Disability and Rehabilitation, 46(4), pp. 629–636. https://doi.org/10.1080/09638288.2023.2172222

Hanczyc, M. M. (2020). *Engineering Life: A Review of Synthetic Biology*, Artificial Life, 26(2), pp. 260–273. https://doi.org/10.1162/artl_a_00318

Hossam, M. (2016). *Niche, a survival game based on the real science of genetics*, Kill Screen - Previously. Available at: https://killscreen.com/previously/articles/niche-survival-game-based-real-science-genetics/ [Accessed 16 January 2025].

Huizinga, J. (1967). *Homo ludens: a study of the play-element in culture*. Translated by R. Bryan. Beacon Press, Boston. (Original work published in 1938 by H. D. Tjeenk Willink, The Netherlands as *Homo Ludens. Proeve eener bepaling van het spel-element der cultuur*). Available at: https://archive.org/details/homoludensstudyo00huiz/page/n3/mode/2up [Accessed 6 February 2025].

iGEM, (2024a). *iGEM*. Available at: https://igem.org [Accessed 16 January 2025).

iGEM, (2024b). *2024 Insights: iGEM Alumni Startups Funding Trends and SynBio x AI Innovations*, iGEM Blog. Available at: https://blog.igem.org/blog/2024/9/17/2024-insights-igem-alumni-startups-funding [Accessed 16 January 2025].

iGEM, (2024c). *iGEM Registry*. Available at: https://technology.igem.org/registry [Accessed 16 January 2025].

iGEM - Team HongKong-JSS, (2024). *HongKong-JSS: Metalytic*. Available at: https://2024.igem.wiki/hongkong-jss/ [Accessed 16 January 2025].

iGEM - Team Marburg, (2024). *Marburg: Tarakate*. Available at: https://2024.igem.wiki/marburg/ [Accessed 16 January 2025].

iGEM - Team UAlberta, (2024). *UAlberta: Renewool*. Available at: https://2024.igem.wiki/ualberta/ [Accessed 16 January 2025].

iGEM - Team VIT Vellore, (2011). *VIT Vellore: In Vivo Drug Factory - Project*. Available at: https://2011.igem.org/Team:VIT_Vellore/Project [Accessed 16 January 2025].

Jacobs, R., Jansz, J. & de la Hera, T. (2017). *The Key Features of Persuasive Games: A Model and Case Analysis*. In: New Perspectives on the Social Aspects of Digital Gaming, Routledge eBooks, pp. 147-151. https://doi.org/10.4324/9781315629308-10

Johnson, D., Klarkowski, M., Vella, K., Phillips, C., McEwan, M. & Watling, C. N. (2018). *Greater rewards in videogames lead to more presence, enjoyment and effort*, Computers in Human Behavior, 87, pp. 66–74. https://doi.org/10.1016/j.chb.2018.05.025

Juul, J., 2010. *The game, the player, the world: looking for a heart of gameness*. In: Proceedings of the DiGRA 2003 Conference: Level Up, Utrecht, The Netherlands. Available at: https://dl.digra.org/index.php/dl/article/view/65 [Accessed 23 January 2025].

Juul, J. & Norton, M. (2009). *Easy to use and incredibly difficult: on the mythical border between interface and gameplay*. In: Proceedings of the 4th International Conference on Foundations of Digital Games, FDG '09. Association for Computing Machinery, New York, NY, USA, pp. 107–112. https://doi.org/10.1145/1536513.1536539

Kang, S. H. K. (2016). *Spaced Repetition Promotes Efficient and Effective Learning: Policy Implications for Instruction*, Policy Insights from the Behavioral and Brain Sciences, 3(1), pp. 12–19. https://doi.org/10.1177/2372732215624708

Kawrykow, A., Roumanis, G., Kam, A., Kwak, D., Leung, C., Wu, C., Zarour, E., Players, P., Sarmenta, L., Blanchette, M. & Waldispühl, J. (2012). *Phylo: A Citizen Science Approach for Improving Multiple Sequence Alignment*, PLOS ONE, 7, e31362. https://doi.org/10.1371/journal.pone.0031362

Kelwick, R., Bowater, L., Yeoman, K. & Bowater, R. (2015). *Promoting microbiology education through the iGEM synthetic biology competition*, FEMS microbiology letters, 362(16), fnv129. https://doi.org/10.1093/femsle/fnv129

Khlebnikov, A., Risa, O., Skaug, T., Carrier, T. A. & Keasling, J. D. (2000). *Regulatable arabinose-inducible gene expression system with consistent control in all cells of a culture*, Journal of Bacteriology, 182(24), pp. 7029–7034. https://doi.org/10.1128/jb.182.24.7029-7034.2000

Kiili, K., de Freitas, S., Arnab, S. & Lainema, T. (2012). *The Design Principles for Flow Experience in Educational Games*, Procedia Computer Science, 15, pp. 78–91. https://doi.org/10.1016/j.procs.2012.10.060

Kitano, S., Lin, C., Foo, J. L. & Chang, M. W. (2023). *Synthetic biology: Learning the way toward high-precision biological design*, PLoS Biology, 21, e3002116. https://doi.org/10.1371/journal.pbio.3002116

Knight, T. F. (2003). *Idempotent Vector Design for Standard Assembly of Biobricks*. Available at: https://dspace.mit.edu/bitstream/handle/1721.1/21168/biobricks.pdf?sequence=1 [Accessed 6 February 2025].

Koellner, C.M., Mensink, K. A. & Highsmith, W. E. (2018). *Basic concepts in human molecular genetics*. In: Molecular Pathology, Elsevier, The Netherlands, Amsterdam, pp. 99–120. https://doi.org/10.1016/B978-0-12-813257-9.00005-X

Kwak, D., Kam, Alfred, Becerra, D., Zhou, Q., Hops, A., Zarour, E., Kam, Arthur, Sarmenta, L., Blanchette, M. & Waldispühl, J. (2013). *OpenPhylo: a customizable crowd-computing platform for multiple sequence alignment*, Genome Biology, 14(10), R116. https://doi.org/10.1186/gb-2013-14-10-r116

Lee, J., Kladwang, W., Lee, M., Cantu, D., Azizyan, M., Kim, H., Limpaecher, A., Gaikwad, S., Yoon, S., Treuille, A., Das, R. & EteRNA Participants. (2014). *RNA design rules from a massive open laboratory*, Proceedings of the National Academy of Sciences, 111(6), pp. 2122–2127. https://doi.org/10.1073/pnas.1313039111

Levy, L. (2015). *The effects of background music on video game play performance, behavior, and experience in extraverts and introverts*, Master's thesis, Georgia Institute of Technology. Available at http://hdl.handle.net/1853/54462 [Accessed 23 January 2025]

Liu, Y., Liu, L., Li, J., Du, G. & Chen, J. (2019). *Synthetic Biology Toolbox and Chassis Development in Bacillus subtilis*, Trends in Biotechnology, 37(5), pp. 548–562. https://doi.org/10.1016/j.tibtech.2018.10.005

Mancas, C. (2014). *Will Software Development Projects Always Risk Delays?*, Journal of Information Technology & Software Engineering, 4(1). Available at: https://www.researchgate.net/publication/289928548_Will_Software_Development_Projects_Always_Risk_Delays [Accessed 6 February 2025]

Mayes, T. (1995). *Learning technology and groundhog day*. In: Proceedings of Hypermedia at Work: Practice and Theory in Higher Education, 10-12 January 1995, University of Kent. Available at https://www.researchgate.net/publication/243786732_Learning_Technology_and_Groundhog_Day [Accessed 23 January 2025]

Menard, J., Diep, P., Sheikh, F., Escobar, A., Dykstra, C. B., Sajtovich, V. A., Ahmadi, A., Bodyreva, E., Boucinha, A., Chandrasekharan, S., Duan, J., Emond, C., Lu, T., McLean, I., Morse, L., Serra, D., Stancescu, A., Suresh, S. & Ingalls, B. P. (2024). *Synthetic biology education and pedagogy: a review of evolving practices in a growing discipline,* Frontiers in Education, 9. https://doi.org/10.3389/feduc.2024.1441720

Michailidis, L., Balaguer-Ballester, E. & He, X. (2018). *Flow and Immersion in Video Games: The Aftermath of a Conceptual Challenge*, Frontiers in Psychology, 9, pp. 1682. https://doi.org/10.3389/fpsyg.2018.01682

Moberly, L. (2024). *GDC 2024: Occupational Burnout 2024: Causes, Impact, and Solutions*. Available at: https://www.thewritegame.com/post/gdc-2024-occupational-burnout-2024-causes-impact-and-solutions [Accessed 23 January 2025].

Naninovel, (2024). *Naninovel*. Available at: https://naninovel.com/ [Accessed October 24 2024]

Newtonsoft, (2024). *Json. NET*. Available at: https://www.newtonsoft.com/json [Accessed October 24 2024]

Owens, T. (2012). *Teaching intelligent design or sparking interest in science? What players do with Will Wright's Spore*, Cultural Studies of Science Education, 7, pp. 857–868. https://doi.org/10.1007/s11422-012-9383-5

Pashler, H., McDaniel, M., Rohrer, D. & Bjork, R. (2008). *Learning Styles: Concepts and Evidence*, Psychological Science in the Public Interest, 9(3), pp. 105–119. https://doi.org/10.1111/j.1539-6053.2009.01038.x

Piaget, J. (1970). Science of education and the psychology of the child. Translation by D. Coltman. Orion, Oxford, England. (Originally published as *La science de l'éducation et la psychologie de l'enfant* in 1969 by Presses Universitaires de France, Paris.) Available at: https://archive.org/details/scienceofeducati00piag [Accessed 6 February 2025].

Praveen, C. K. & Srinivasan, K. (2022). *Psychological Impact and Influence of Animation on Viewer's Visual Attention and Cognition: A Systematic Literature Review, Open Challenges, and Future Research Directions*, Computational and Mathematical Methods in Medicine, 2022, pp. 1-12. https://doi.org/10.1155/2022/8802542

Prensky, M. (2012). *From Digital Natives to Digital Wisdom: Hopeful Essays for 21st Century Learning*, Corwin Press, Thousand Oaks, CA, USA. https://doi.org/10.4135/9781483387765

Pujol, M., Caroux, L. & Lemercier, C. (2024). *Flow State Requires Effortful Attentional Control but Is Experienced Effortlessly by Video Game Players*, Interacting with Computers, 36(6), pp. 371–382. https://doi.org/10.1093/iwc/iwae026

Purnick, P. E. M. & Weiss, R. (2009). *The second wave of synthetic biology: from modules to systems*, Nature Reviews Molecular Cell Biology, 10, pp. 410–422. https://doi.org/10.1038/nrm2698

Røkke, G., Korvald, E., Pahr, J., Øyås, O. & Lale, R. (2014). *BioBrick assembly standards and techniques and associated software tools*. In: Methods in Molecular Biology, Humana Press, Totowa, NJ, USA, pp. 1–24. https://doi.org/10.1007/978-1-62703-764-8_1

Ryan, M. -L., Emerson, L. & Robertson, B. J. (2014). *The Johns Hopkins Guide to Digital Media*, Publishing Research Quarterly, 30, pp. 330–331. https://doi.org/10.1007/s12109-014-9372-5

Salis, H. M., Mirsky, E. A. & Voigt, C. A. (2009). *Automated design of synthetic ribosome binding sites to control protein expression*, Nature Biotechnology, 27, pp. 946–950. https://doi.org/10.1038/nbt.1568

Schneider-Poetsch, T. & Yoshida, M. (2018). *Along the central dogma—controlling gene expression with small molecules*, Annual Review of Biochemistry, 87, pp. 391–420. https://doi.org/10.1146/annurev-biochem-060614-033923

Schrader, P., Deniz, H. & Keilty, J. (2016). *Breaking SPORE: Building Instructional Value in Science Education using a Commercial, Off-the Shelf Game*, Journal of Learning and Teaching in Digital Age, 1(1), pp. 63–73. Available at: https://dergipark.org.tr/en/download/article-file/1175536 [Accessed 23 January 2025]

Singh, A., Ahsan, F., Blanchette, M. & Waldispühl, J. (2017). *Lessons from an Online Massive Genomics Computer Game*. In: Proceedings of the AAAI Conference on Human Computation and Crowdsourcing, 5(1), pp. 177–186. https://doi.org/10.1609/hcomp.v5i1.13309

Singh, V. (2014). *Recent advances and opportunities in synthetic logic gates engineering in living cells*, Systems and Synthetic Biology, 8, pp. 271–282. https://doi.org/10.1007/s11693-014-9154-6

Slomovic, S., Portnoy, V. & Schuster, G. (2008). *Chapter 24: Detection and Characterization of Polyadenylated RNA in Eukarya, Bacteria, Archaea, and Organelles*. In: Methods in Enzymology, Academic Press, San Diego, CA, USA, pp. 501–520. https://doi.org/10.1016/S0076-6879(08)02224-6

Solomon, C., Harvey, B., Kahn, K., Lieberman, H., Miller, M. L., Minsky, M., Papert, A. & Silverman, B. (2020). *History of Logo*, Proceedings of the ACM on Programming Languages, 4(HOPL), pp. 79:1-79:66. https://doi.org/10.1145/3386329

Sweetser, P. & Wyeth, P. (2005). *GameFlow: A Model for Evaluating Player Enjoyment in Games*, Computers in Entertainment, 3(3), pp. 3. https://doi.org/10.1145/1077246.1077253

Symes, C. (2004). *A sound education: the gramophone and the classroom in the United Kingdom and the United States, 1920–1940*, British Journal of Music Education, 21, pp. 163–178. https://doi.org/10.1017/S0265051704005674

Tang, S., Hanneghan, M. & El Rhalibi, A. (2007). *Describing Games for Learning: Terms, Scope and Learning Approaches*. Available at: https://www.semanticscholar.org/paper/Describing-Games-for-Learning-%3A-Terms-%2C-Scope-and-Tang-Rahman/8e570db44e30f50d18e03bf95c7ae4db04295ce4 [Accessed 23 January 2025]

Teferra, T. F. (2021). *Should we still worry about the safety of GMO foods? Why and why not? A review*, Food Science & Nutrition, 9(9), pp. 5324–5331. https://doi.org/10.1002/fsn3.2499

The Biology of Subnautica | Full Documentary, (2022) Youtube video, added by Curious Archive [Online]. Available at: https://www.youtube.com/watch?v=1xBrxWzKzDQ [Accessed 23 January 2025]

Unity (2024). *Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine*. Available at: https://www.unity.com [Accessed 30 January 2025].

Unity TMP (2024). *TextMesh Pro Documentation*. Available at: https://docs.unity3d.com/Packages/com.unity.textmesh-pro@4.0/manual/index.html [Accessed 24 October 2025].

van Eck, R. N. (2006). *Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless*. EDUCAUSE Review, 41(2), pp. 16-30. Available at: https://er.educause.edu/articles/2006/3/digital-game-based-learning-its-not-just-the-digital-natives-who-are-restless [Accessed 30 January 2025].

van Zanden, J. L., d'Ercole, M., Baten, J., Rijpma, A., Timmer, M. (2014). *How Was Life?: Global Well-being since 1820*. OECD Publishing, Paris, France. https://doi.org/10.1787/9789264214262-en

Wang, X., Liang, Q., Luo, Y., Ye, J., Yu, Y., Chen, F. (2024). *Engineering the next generation of theranostic biomaterials with synthetic biology*. Bioactive Materials, 32, pp. 514–529. https://doi.org/10.1016/j.bioactmat.2023.10.018

Wang, Y. -H., Wei, K. Y., Smolke, C. D. (2013). S*ynthetic biology: Advancing the design of diverse genetic systems*. Annual Review of Chemical and Biomolecular Engineering, 4, pp. 69–102. https://doi.org/10.1146/annurev-chembioeng-061312-103351

Wilkinson, N., Ang, R. P., Goh, D. H. (2008). *Online Video Game Therapy for Mental Health Concerns: A Review*. International Journal of Societal Psychiatry, 54(4), pp. 370–382. https://doi.org/10.1177/0020764008091659

Zhai, W., Duan, Y., Zhang, Xiaomei, Xu, G., Li, H., Shi, J., Xu, Z., Zhang, Xiaojuan. (2022). *Sequence and thermodynamic characteristics of terminators revealed by FlowSeq and the discrimination of terminators strength*. Synthetic and Systems Biotechnology, 7(4), pp. 1046–1055. https://doi.org/10.1016/j.synbio.2022.06.003

Appendix A: Hardware and Software Considerations

This appendix contains information about the hardware and software used to make and run the game.

Hardware

The game was developed and tested on high-end computers with a Windows 10 operating system (OS) but tested regularly on an old low-end Windows 10 laptop and a Linux system. The game functions mostly using mouse inputs, with keyboard only required for naming player-made database entries and naming a save file. This allows for the possibility of porting the game to mobile devices running Android/iOS due to Unity's wide hardware support, since mouse inputs translates well into touch screen controls and writing can be done using a virtual touchscreen keyboard. This would be beneficial for educational purposes, but the game's UI would need a redesign for touch devices.

The current minimum system requirements are listed below for personal computers (PCs) and laptops, although the game will most likely require less computing power after completion and further optimization for a wider variety of devices.

Minimum system requirements

OS: Windows 10/11 or Linux

Processor: Entry-level dual-core processor

Random Access Memory (RAM): 2 GB

Graphics: 1 GB VRAM, Entry-level or integrated GPU

Graphics API: DirectX, Vulkan or OpenGL

Storage: 4 GB

The game does not support Mac OS X yet, since Unity requires a Mac computer to build a functioning build for Mac computers and none of the team members own one. Windows and Linux are the only supported operating systems currently, but Mac OS X support will be included as soon as possible.

Headphones are also recommended for the best experience of the game. The music and audio of the game is meant to both increase player immersion, as well as help the player focus.

Unity Game Engine, Plugins and Extensions

To develop InGENEous, various computer-based development tools, such as a game engine, plugins, extensions and databases, were utilized. Like most software, InGENEous builds upon existing software with Unity as the core program, with a few essential extensions and plugins.

Unity

Unity is a proprietary, cross-platform game engine and game development toolkit for Microsoft Windows and Mac OS X. Unity supports both 2D- and 3D-application development and includes e.g. full rendering and physics engines, a UI system for menus and other user interfaces, and supports C# and JavaScript scripting across all platforms (Unity 2024). As of 2024, Unity supports more than 20 platforms on desktop systems, mobile devices, game consoles, smart devices and VR and AR platforms. The Unity engine is free to use on non-commercial projects and familiar with the development team, which is why it was chosen as the development platform. Since Unity is also so widely used globally, it's easy to find solutions for developmental problems as they arise.

For its cross-platform support, Unity uses Mono developed by Xamarin. Mono is a platform-agnostic open-source toolkit for compiling and using Microsoft's .NET framework outside the Windows operating system. Before running an application, Unity compiles the C# or JavaScript code written by the developer to dynamically linked library (DLL) files. At runtime, these are compiled using the .NET virtual machine on Windows or the Mono virtual machine on other platforms using a Just-in-Time (JIT) compiler. This system allows Unity developers to use the popular C# language in programming applications for all platforms. The downside to this appears to be the deep integration of Mono into the Unity engine and the resulting slow pace of updates to the feature set offered by Mono. InGENEous supports Mono up to 4.0.

As the initial development of InGENEous began, Unity version 2021.3.17f1 was chosen since this build was well-known to us and InGENEous uses the 2.1 .NET standard and .NET framework. However, the next step of the development process will include updating the game to the newest Unity revision,

Unity 6, which will allow a more well-structured programming pipeline for the game, thus reducing programming and asset bloat (Unity 2024).

Scenes

All games made with Unity are essentially a collection of “scenes”. A scene principally contains all program data currently loaded into the computer’s random-access memory (RAM). In video games, scenes are usually used to load and unload different sections of the game to decrease RAM requirements, and to enhance the performance of the application. InGENEous contains 6 main scenes currently, and the game swaps between active scenes when necessary.

Objects

Scenes are populated by objects, which can be considered as data containers for individual entities within the application. Video games made with Unity usually have objects for characters and physical objects within the game, but also for abstract, non-visible functions such as the camera, audio controllers and user input devices. Objects can also contain child objects that can be either enabled or disabled. Disabling an object hides the object and its children from the program and halts the execution of scripts associated with the object. InGENEous has a wide variety of objects, ranging from UI elements, such as buttons, to BioBricks used for constructing DNA devices.

Components

Components are the data that objects contain. Each component in an object gives the object a property, such as a 3D mesh that gives the object a shape. Components also include collision meshes for physics simulations, light emissions and particle effects, shaders for object color and surface material and so on. Components are also responsible for all program logic in the form of scripts.

Scripts

All of Unity’s programming is done using scripts, which can be either used as components linked to objects or separately on their own. There is no accessible main loop within Unity for doing general computing and therefore all

code must be instead written into scripts and tied to objects in the object hierarchy. Scripts in Unity can be thought of as C# classes and can hold variables and contain methods.

Methods

Methods in scripts can be invoked by two procedures: from special built-in methods and via events. At specific points of execution, like before rendering a new frame, the Unity engine calls an associated method in each enabled script. For example, the Update() method is called between each frame and can be used for animating a BioBrick's movement for example. Unity also supports C# events which are used in InGENEous for user interactions. Events are used to notify objects and classes of a predetermined set of conditions being met, after which an appropriate action is undertaken. We have created an event system where user actions, such as clicking on a BioBrick, send out an event that an event listener has subscribed to. The event listener then handles the different events by calling the appropriate methods in other scripts.

Extensions

One of the core features of the Unity SDK is extensive support for 3rd-party extensions and plugins. Extensions, in this case, refer to downloadable packages of Unity-native objects and scripts while plugins refer to externally compiled .NET code assemblies. InGENEous currently uses a single extension from Unity's Asset Store and two open-source 3rd-party extensions. This section covers the purpose of these components and their licensing terms.

Naninovel

Naninovel is a comprehensive solution for creating full-featured visual novels using writer-friendly tools. Initially our plan was to create our own visual novel scene for character interactions and the introduction in InGENEous, but in the end, two licenses of Naninovel were acquired to hasten the process. The purchase of a license allows the user to create commercial and non-commercial games, as well as use Naninovel scenes in associated media, such as videos and pictures. (Naninovel 2024)

Newtonsoft JSON Extension

Newtonsoft JSON extension for Unity is a popular high-performance JSON framework for .NET. This extension allows for the use of more complex data structures within the .json file format. The databases and save files can then store more data per data entry, than Unity's own JSON reader would allow.

Generally, .json files are used to make text-based language-independent lists, which simplifies the workflow when developing for multiple operating systems through Mono and .NET. This extension is open-source and doesn't require licensing. (Newtonsoft 2024)

TextMesh Pro Extension

TextMesh Pro Extension for Unity allows better formatting, more fonts and higher quality text to be displayed within the Unity game engine. As an educational game, InGENEous contains a lot of text, and Unity's own text formatting style is not capable of displaying large quantities of scalable text with high fidelity. This extension is open-source and does not require licensing. (Unity TMPro 2024)

Appendix B: Additional Details about the Game

This appendix offers additional information about some of the game's features that aren't covered in the main text.

Main Menu

When the player opens InGENEous, they are greeted with a mandatory Unity logo, followed by the logos of Aalto-university, VTT, CYSS and Jenny and Antti Wihuri foundation. After this, the player is presented with the main menu scene of the game, seen on Figure 36.

Figure 36. A screenshot from the main menu of the game. UI Art by Julia Rässa and Aarni Aspi.

From this screen the player can access various external parts of the game using the on-screen buttons, such as the level editor, custom levels created by educators and other players, and the credits of the game. Links to more educational and scientific games related to SynBio, such as **FoldIt**, **EteRNA**, **CellCraft** and **Hero.coli** can be accessed by using the "More Games" button. The logos in the bottom right corner also function as links that will take you to the website of these organizations to access more information.

The options menu allows the player to adjust audio and video settings, as well as allowing the player to choose user interface (UI) themes for the game, unlocked as rewards for project completion. Audio settings are separated into

master, voice-over, music and sound effects, so the player may disable parts of the game's audio if needed. The only video settings available now are **Full-screen on/off** and **resolution**. All the art and UI in the game has been made scalable, so the visual elements of the game are sharp and readable regardless of the screens resolution the game is being played on.

If the player clicks on the play button, they are prompted to either pick an existing save file, or to create a new save file. If the player picks an existing save file, the game continues from when the game was previously turned off. If the player decides to create a new save file, they are prompted to name it, which creates a JSON file which updates the player's progress throughout the game. A copy of the **DataBase** file is also created to store the database entries created by the player on this save file specifically. This was done to not have parts made by the player showing up on another player's save file on the same device, and for the player to be able to save the file and the parts in it elsewhere if needed.

The save file contains a few variables related to video and audio settings, as well as the name of the save file. The most important variable however is the **ProgressTracker**. This variable is an integer number, that starts from 0. Whenever the player progresses beyond a checkpoint, a script verifies the check to be completed, and the integer grows by one thus unlocking more content to the player. Examples of checks and unlocks can be seen in table 3.

Table 3. Functionality of the game's save file.

ProgressTracker	Check	Unlocks
0	Dialogue with LAB.RR	Tutorial 1 available in EnGENEer
1	Tutorial 1 completed	Email 1 from 1st client
2	Email 1 read	Call 1 with 1st client available
3	Call 1 completed	Project 1 available in EnGENEer
4	Project 1 completed	Call 2 with 1st client available
5	Call 2 completed	1st client bio available (reward), Email 1 from 2nd client

Image Gallery

This section contains a gallery of additional images from the game.

Figure 37. A screenshot of the lobby and the previous iteration of the main UI. Art by Alli Loikkanen, UI art by Julia Rässa.

Figure 38. A screenshot of the greenhouse. Image cropped for text visibility. Art by Alli Loikkanen and Zsófia Kecskés.

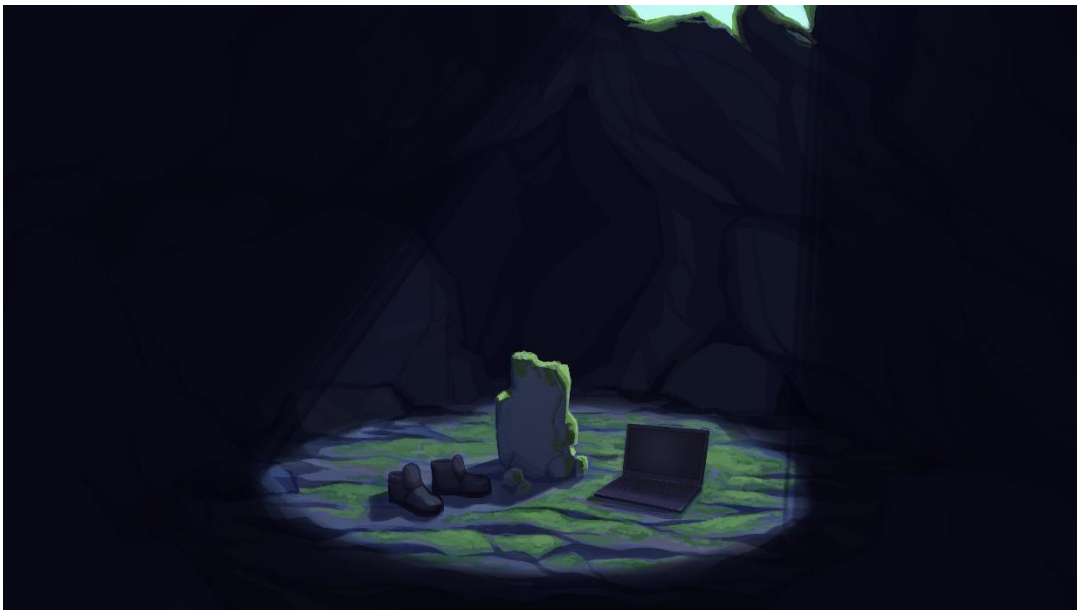

Figure 39. The professor's shoes and laptop in a secret location. Art by Alli Loikkanen.

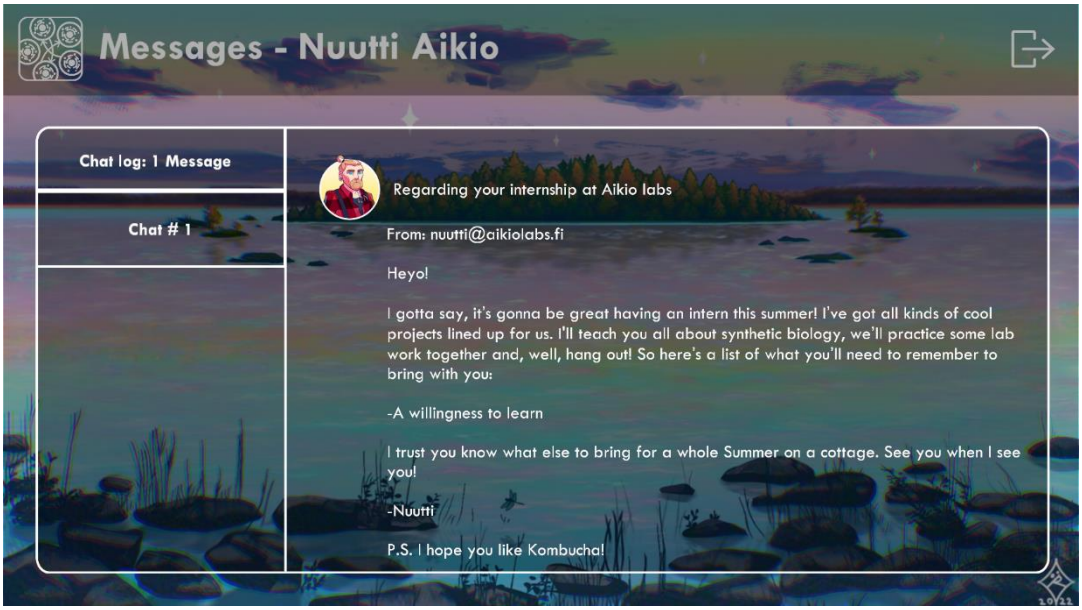

Figure 40. A screenshot of an email from the professor in the message function of Forum Quorum. Art by Julia Rässa and Zsófia Kecskès. UI art by Tuomas Hellsten.

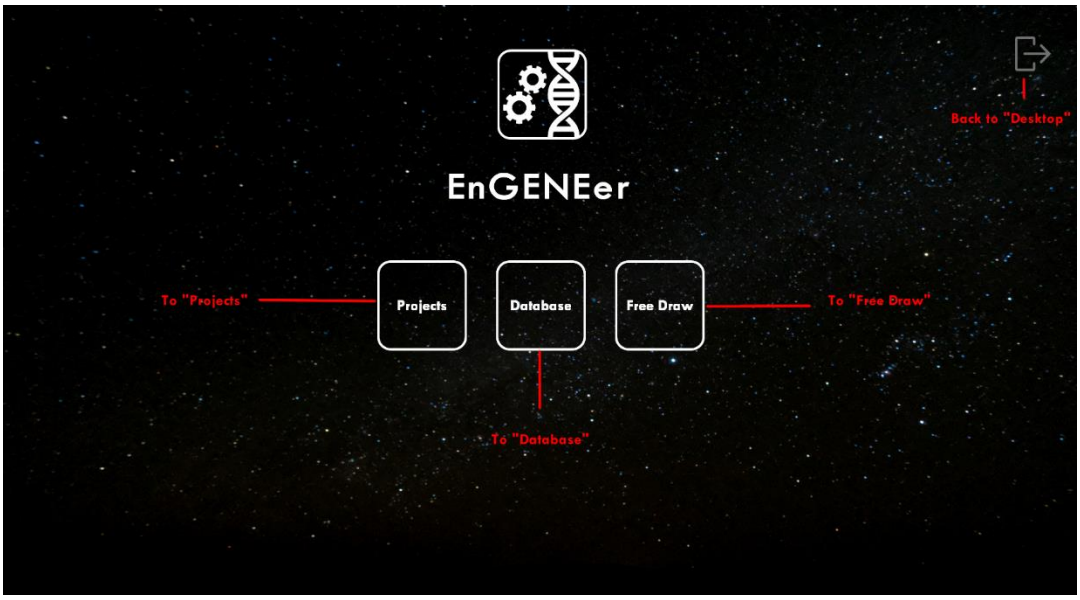

Figure 41. A screenshot of the main screen of EnGENEer. UI art by Tuomas Hellsten and Aarni Aspi.

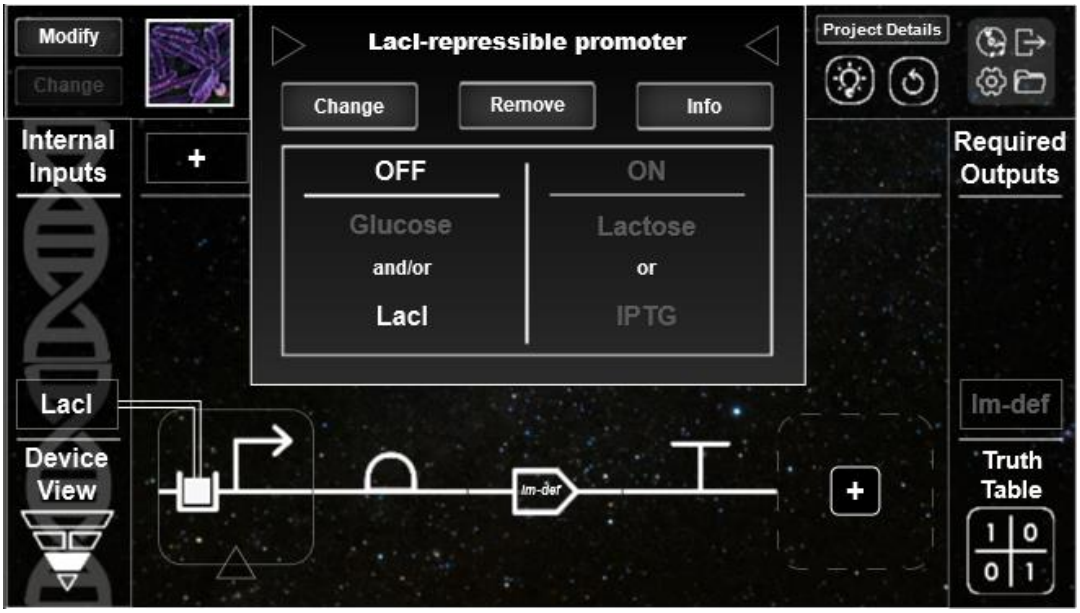

Figure 42. A mock-up of the device view, LacI-repressible promoter currently selected. UI elements by Julia Rässa and Tuomas Hellsten. UI by Aarni Aspi.

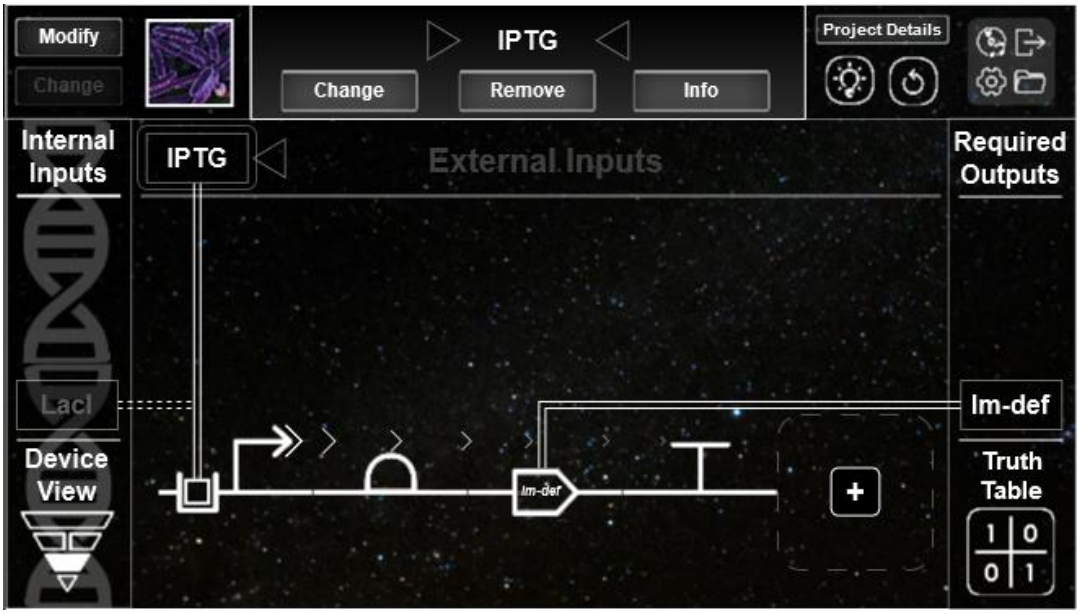

Figure 43. A mockup of the device view, with a functional device and IPTG currently being selected. Some animations are also visible. UI elements by Julia Rässa and Tuomas Hellsten. UI by Aarni Aspi.

Database File Structure

A few images illustrating the structure of the database entries of the game.

Figure 44. A few chassis entries from the database.

```
[
  {
    "deviceName": "(P)Strong_Constitutive_LacI_Generator",
    "deviceNickname": "(P)sCon_LacI_Gen",
    "deviceCategory": "Protein Generators",
    "deviceID": "BBa-Y05062410561",
    "sequence": "",
    "length": "",
    "isDeviceInputInducible": false,
    "inputs": [
      "(P)pStrong"
    ],
    "outputs": [
      "LacI"
    ],
    "optionalDescription": "",
    "invidualParts": [
      {
        "partMaincategory": "Promoters",
        "partname": "(P)pStrong",
        "parentSlotPos": "0"
      },
      {
        "partMaincategory": "Ribosomal Sites",
        "partname": "(P)sRBS",
        "parentSlotPos": "10"
      },
      {
        "partMaincategory": "CDS",
        "partname": "LacI",
        "parentSlotPos": "20"
      },
      {
        "partMaincategory": "Terminators",
        "partname": "(P)sfDT",
        "parentSlotPos": "30"
      }
    ]
  },
```


Figure 45. A device entry in the database.

Appendix C: Notes on Presentation

The presentation of a game is crucial to keeping the players invested and immersed within the world of the game. Great care was taken while designing the various aspects of presentation within InGENEous, and various aspects of the presentation are covered below.

User Experience (UX)

The UX of InGENEous has been under scrutiny ever since development began. For example, originally when placing BioBricks in the design tool, the players would need to drag and drop the BioBricks where they want them. This introduced multiple unneeded mouse clicks for the players, and now, players can simply click once, choose a BioBrick from the database, and place the BioBrick. The UX of a game is very important, because forcing unnecessary "work" upon the player tends to increase frustration, thus decreasing player immersion and motivation.

Accessibility

The amount of new video game releases has grown substantially ever since their inception, as well as the number of people playing them. This has led to a rising demand for accessibility features in video games, with accessibility features becoming both more prevalent and robust in the last few decades. The topic of accessibility has been covered several times in the Game Developers' Conference (GDC) lectures in the last decade, as well as projects such as Game Accessibility Guidelines offering developers a list of accessibility guidelines to follow. So much can be done with a few slight changes when it comes to accessibility, with impaired players' suggestions acting as the arbiter of change when it comes to accessibility.

InGENEous has been designed with accessibility in mind since the first day of development. Most learning materials in the game are available as text, with supporting visuals and audio. This allows for heightened accessibility, as well as more support for different types of learners. Colors aren't used for indicating state changes or interactions with colorblind players in mind. InGENEous does not require reaction speed or complex input combinations either, allowing for more freedom for physically impaired players.

The in-game text and its sources have also been thought out in case anyone wants to make a translation of the game in the future, allowing non-English-speaking players to learn through the game as well. Optimizing the game and

decreasing system requirements also enables more accessibility with a wider variety of supported devices.

More can always be done in this regard, but more accessibility features aren't planned currently. Once the test version of the game is launched, all player suggestions will be taken into consideration.

Art

When it comes to the art direction of InGENEous, the figures in this thesis provide clear examples of its execution. For the painted illustrations, characters and backgrounds; bright colors, low detail and cartoonish linework were the focus to decrease the time required for their production. The characters were designed with cartoonish proportions, giving the all the characters a distinct silhouette.

When it comes to the UI art of the game, minimalism and a clear color palette for visibility were the main design goals. This also mirrors the often simple and minimalist UI design of modelling/design software, giving the game a more "professional" look.

Animations

When it comes to animations, InGENEous is currently quite limited in that regard. All buttons in the game are animated to indicate intractability and allow some input responsivity to the players. Characters also have limited animations in dialogue for enhanced characterization.

Quite a lot could be done with more animations, however. The direction of chemical reactions could be indicated by moving shapes and more player interactions, such as placing a BioBrick, should have a pleasing animation tied to them. The mouths of characters could be rigged to move whenever the characters speak, allowing a higher degree of immersion for the players.

The tutorials and the introduction of the game would also benefit from some animations. The illustrations in the introduction could be animated to indicate changes or interactions, and the using animations in tutorials would help guide the players eyes towards a point of interest. The water feature located in the greenhouse would also benefit with the flow of water being animated.

Sound

The sound design of InGENEous was designed with a few core attributes in mind.

The sound effects should be simple and pleasing and should act as small rewards for the player interacting with the game. Choosing quality over quantity and focusing on adding pleasing sound effects to interactions repeated often by players while designing will allow for a high degree of perceived interactivity, while avoiding unnecessary development time. Dialogue scenes with the game's characters contain some sound effects, which adds a bit of extra color to characterization.

The ambient background sounds used in exploration serve a dual function: they both help in making these spaces more immersive and believable, while giving the players a relaxing background ambience between sections of cognitively-taxing gameplay.

The music of the game also serves multiple purposes. It increases player enjoyment, increases immersion, enables stronger characterization and helps the players in achieving a flow state. With the characters of the game being as colorful as they are, the dialogue scenes with them felt lacking with simple, generic background music. Lukas Baldvinsson was commissioned to compose a unique theme for all the characters to make them feel more individual from each other. These musical themes were crafted with the personality and nationality of the characters in mind, and they certainly are one of the highlights of the game.

The current placeholder gameplay music used in the design projects has been chosen from a bought collection of royalty-free music tracks. These tracks are all simple instrumental tracks and were made to loop to avoid distracting cuts in audio. The purpose of these tracks is to give a bit more color to the different projects, as well helping the player stay in a flow state. These tracks will hopefully not stay in the game, since the cohesion between them is lacking.

The plan also includes narrating all dialogue in the game. This allows better learning for auditory learners, as well as allows better characterization for the characters of the game. The characters voice actors should be chosen with the cultural background of the characters in mind, with different English accents and a delivery of voice lines based on the different personalities of the characters.