PROJECT G - EDUCATIONAL GAME FOR LEARNING GENETIC ALGORITHM

MR. THANAPAT MAHISKHAMIN
MR. POJNARIN NANTA
MR. SIRAPOP APANANDA

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)
FACULTY OF ENGINEERING
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI
2022

Project G - Educational Game for Learning Genetic Algorithm

Mr. Thanapat Mahiskhamin
Mr. Pojnarin Nanta
Mr. Sirapop Apananda

A Project Submitted in Partial Fulfillment
of the Requirements for
the Degree of Bachelor of Engineering (Computer Engineering)
Faculty of Engineering
King Mongkut's University of Technology Thonburi
2022

Project Committee

.... (Assoc.Prof. Natasha Dejdumrong, D.Tech.Sci.)	Project Advisor
.... (Taweechai Nuntawisuttiwong, Ph.D.)	Project Co-Advisor
.... (Prapong Prechaprapranwong, Ph.D.)	Committee Member
.... (Asst.Prof. Suthathip Maneewongvatana, Ph.D.)	Committee Member
.... (Assoc.Prof. Thumrongrat Amornraksa, Ph.D.)	Committee Member

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

In this chapter, we will describe the basic information of the project including background, objectives, scope of work, and project schedule.

1.1 Background

Nowadays, human technologies have been growing exponentially with no sign of slowing down; with the ongoing pandemic, many things have been shifted to the digital world, disrupting older technologies and forcing people in this generation to improvise, adapt, and overcome their capabilities. One of the constantly-evolving technologies is the game industry.

In recent years, computer games have been getting more attention from the public as the game industry has been rapidly elevating nonstop. Usually, these media get acknowledged as relaxing activities; however, they also have great potential as learning tools [1]. They are easy to access by nearly everyone, especially younger generations who are accustomed to computers and technologies. Furthermore, they can easily get caught by entertainment media; games can serve as tools for education for them. There are lots of educational games coming out, and they cover almost every topic, from simple language or first-grade math to physics, chemistry, and computer science.

With the rapid growth of technologies and knowledge, education has become more radical than ever; to catch up with everything. Since the popularity of computer games is increasing, the interest in learning computer engineering is increasing. But However some content ~~of~~ in this field is hard to ~~understand~~ study ~~for some people~~, for example, the Genetic Algorithm. Integrating knowledge about the Genetic Algorithm with computer games provides friendlier ways to access and understand the Genetic Algorithm.

Since there is no known existing ? academic game teaching about the Genetic Algorithm [2], A after some consideration, planning, and designing, our group came up with a game to simulate how the Genetic Algorithm works ①

what ? and present them to the audiences in a simple manner; to give them more understanding of this topic, which ②

How?. dungeon crawler and animal-breeding simulator genres come in handy. Since the algorithm can be used in many aspects of those genres, the audiences will get acquire more vision of how it works in different ways and gain more experiences by trying it themselves. Moreover, a game platform holds more advantages than a traditional educational platform as it provides friendlier access to every generation, with no condition or pressure to pick a game and play, which suits the trend of lifelong learning. ดูเหมือนพูดลอยๆ ไม่มีหลักฐาน

1.2 Objectives

As said before, these objectives are separated into two parts, those for the organizers; and those for players.

1. To study and research Unity engine, C#, and the Genetic Algorithm.
2. To make a simple educational game for people who want to learn about the Genetic Algorithm.
3. To research and apply the Genetic Algorithm in the game-making field.
4. To assist players in practicing making and planning decisions and managing resources.

1.3 Scope of Work

In this part, we will list system requirements, systems included in the project, and educational topics we will cover.

1. Windows-supported game developed using Unity.
2. In-game System
   (a) Breeding Farm
   Simulating monster breeding using the Genetic Algorithm. The system consists of breeding farms and habitats. The breeding can only be done on breeding farms.
   (b) Weapon Factory
   มีแค่ knapsack
   ใช้เป็น knapsack เลย
   Real-world problem solving as a weapon creation using the Genetic Algorithm. There are 5 factories in total.
   (c) Research Lab
   The system where players must learn and test their knowledge about the Genetic Algorithm to unlock other new facilities.
   (d) Miscellaneous
      i. Shop
      ii. Time progression
      iii. Resources
      iv. Arena
      v. Quest

   อธิบาย เล็กน้อย
   feature มีอะไร แค่ไหน
3. Educational topic
   (a) Genetic Algorithms
      i. Selection
         A. Random Selection
         B. Tournament-based Selection
         C. Roulette Wheel Selection
         D. Rank-based Selection
         E. Elitism
      ii. Crossover
         A. Single-point Crossover
         B. Two-point Crossover
         C. Uniform Crossover
      iii. Mutation
         A. Bit Flip Mutation
         B. Flip Bit Mutation
         C. Boundary Mutation
   (b) Real-world problems
      i. Standard 0/1 Knapsack Problem
      ii. Multiple 0/1 Knapsack Problem

1.4 Project Schedule

We use Scrum as a development process. We will list details about development activities, product backlog, and draft schedule.

1.4.1 Development Activities

Within the Scrum process, there are 3 important activities which are Sprint, Weekly Meeting, and Sprint Review.

1. Sprint
   All development processes are divided into nine 3-week sprints. The total estimated effort for each sprint is roughly equal to 27 man-half-days.

2. Weekly Meeting
   Meeting 15 minutes or less, repeatedly on Friday, when each member has their own responsibilities. The members report their own accomplishments, will accomplish, and Impediments.

3. Sprint Review
   Meeting at the end of each Sprint, 3 hours or less, to review completed tasks, design artifacts, and adjust Backlog and plans for the next Sprint.

1.4.2 Product Backlog

We will list the task breakdown in a form of the product backlog. This also includes the priority and the estimated effort for doing each task.

Table 1.1 Product Backlog

ID	Story	Effort Estimate (Man-Half-Days)	Priority
1	Developers need to do project documents.	24	1
2	Players can learn about the Genetic Algorithm.	36	2
3	Players can apply knowledge to solve real-world problems.	36	3
4	Players can breed their monsters.	24	4
5	Players have mandatory goals for playing a game.	16	5
6	Players have other activities to spend their time during the breeding phase.	36	6
7	Developers integrate all the systems.	24	7
8	Developers add more aesthetics to the game.	12	8
9	Developers assure quality.	32	9
Total		240	

1.4.3 Draft Schedule

The draft schedule of the Sprints will be shown as follows.

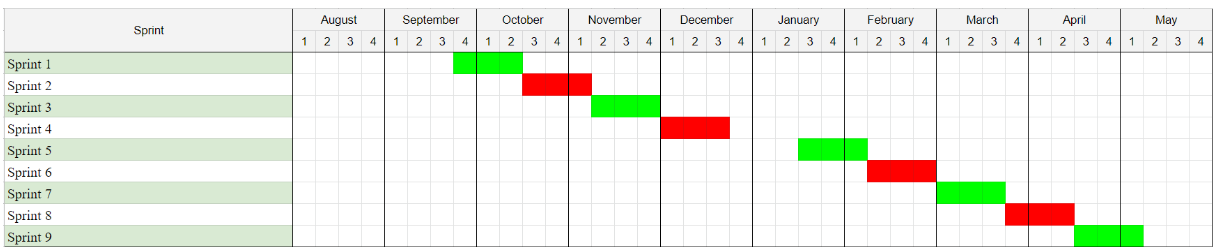


Figure 1.1: Draft Schedule of the Project

CHAPTER 2 BACKGROUND THEORY AND RELATED WORK

For comprehension, we will present more information needed to understand our work. This chapter begins with several theories involving the game design, education design, and engineering content like Genetic Algorithm. Then, we will describe the the tools and technologies required to develop the game. Finally, we will discuss the works that related to our project.

2.1 Game Design Theory

This topic covers the game theories we used in this project including Natural Funativity, Maslow's Hierarchy of Needs, and Eight Kinds of "Fun". We will apply these theories to increase the quality of our game in the aspect of "Fun".

2.1.1 Natural Funativity

According to Noah Falstein's theory [3], all fun derives from practicing survival and social skills, which consist of 3 categories.

ยกตัวอย่างเกม ที่มี Fun เหล่านี้ และมีอย่างไร

1. Physical Fun
   The fun of using the physical body which is strongly related to the survival instinct. Especially under depression or threatened situations which will instantly capture the attention.
2. Social Fun
   The fun of interacting with others. The interaction can be conversations, competitions, cooperation, exchanges, etc.
3. Mental Fun
   The fun of using intelligence. It could be achieved in many ways, such as reasoning, planning, decision-making, recognizing, solving problems or puzzles, etc.

Conclusion เกี่ยวกับ fun ที่เกมตัวเองมี มีครบ 3 หรือ ไม่ครบ

2.1.2 Maslow's Hierarchy of Needs

Maslow's Hierarchy of Needs implies that one's motivation has an order to satisfy based on different needs [4], representing each level in the Figure 2.1 below.

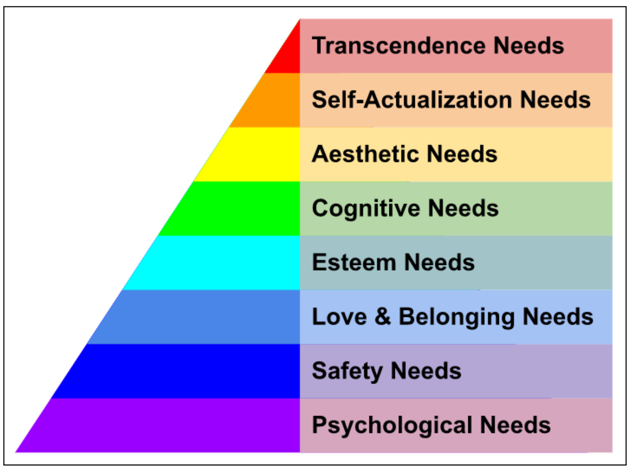


Figure 2.1: Maslow's Hierarchy of Needs

1. Physiological Needs
   It is the most essential need for every human to survive, and it is mandatory to fulfill this kind of need before going further, such as air, water, food, clothing, etc.

2. Safety Needs
   The need for stability in life, also regarded as the need for protection from threats, including personal security, shelter, order, law, etc.

3. Love and Belonging Needs
   After the previous two needs have been fulfilled, the next level of need involves a social and a feeling of belongingness. Humans need to have a sense of unity with the environment. An example includes friendship, intimacy, trust, etc.

4. Esteem Needs
   The esteem needs consist of two aspects, esteem for oneself and the desire for reputation or respect from others. This kind of need can be fulfilled by receiving dignity, achievement, status, prestige, etc

5. Cognitive Needs
   The need for learning, and gaining intelligence. This includes skill, knowledge, experience, understanding, curiosity, exploration, etc.

6. Aesthetic Needs
   This level of need is about seeking the thing that satisfies humans, as they need to search for and appreciate beauty, including balance, form, etc.

7. Self-Actualization Needs
   This need is about realizing personal potential. Causing humans to need for personal growth, and trying to be everything one person could be.

8. Transcendence Needs
   The ultimate need a person could wish. This level of need is motivated by values outside of self. This may be in a form of helping others or an ability to perform something mystical that a typical person can't do.

Explain how to use them in game

2.1.3 Eight Kinds of "Fun"

Marc LeBlanc proposed the idea of different kinds of "Fun" as a set of vocabularies to describe the kinds of fun that people are experiencing to be more specific and precise [5].

1. Sensation - Game as sense-pleasure
   This kind of fun involves the physical senses of players, including visual, sound, sometimes physical movement, or even game pace.

2. Fantasy - Game as make-believe
   It also acknowledged as escapism or immersion. It was gained by immersing oneself into the game world and possessing the ability to do things that cannot be done in the real world.

3. Narrative - Game as drama
   It revolves around unfolding stories that the game has to offer. With narration, players get a sense of direction they can look forward to, sometimes even with their narration.

4. Challenge - Game as obstacle course
   Overcoming courses of obstacles, solving difficult puzzles, defeating difficult enemies, or anything that provides highly competitive value to the player so they can challenge themselves.

5. Fellowship - Game as social framework
   This kind of fun comes from gaining social interactions and cooperating with other players. A multi-player game is a good example.

6. Discovery - Game as uncharted territory
   Exploring new things in the game world or within oneself could be fun. Players could explore new area of maps, search for secret items, or dive through side stories.

7. Expression - Game as self-discovery
   This kind of enjoyment comes from getting a chance to express oneself creatively, such as role-playing as a character in RPG games or conveying one's thoughts into creation in sandbox games.

8. Submission - Game as pastime
   Doing daily tasks repetitively and playing the game casually, which are the opposite side of Challenge, also count as fun. Sometimes, just simply enjoying a game and relaxing are enough.

2.2 Education Design Theory

This topic covers the education theories we used to design the education part of the game. We will apply these theories the ensure that our game will actually educate the player. The theories involved in this topic are Bloom's Taxonomy, Outcome-based Education, and the ADDIE model.

2.2.1 Bloom's Taxonomy

The theory published by Benjamin Bloom presents the 6 levels of learning [6]. These levels indicate the depth of the learning and are used widely in the education field. Each level of learning consists of a set of action verbs, the verb indicating whether the learner reaches the learning level. We use this theory to design the learning outcomes of the game. พูดถึง Figure 2.2 ด้วย

1. Remember
   The lowest level of learning. The learner who achieves this level will be able to remember and recall the basic concepts of the knowledge. The action verbs in this level include duplicate, list, memorize, repeat, state, etc.

2. Understand
   The learner is able to explain the concepts they have learned. The action verbs in this level include classify, demonstrate, explain, identify, illustrate, etc.

3. Apply
   The learner is able to apply the knowledge to new situations they never met before. The action verbs in this level include implement, solve, use, operate, model, etc.

4. Analyze
   The learner is able to compare the different knowledge and see connections among the ideas. The action verbs in this level include compare, contrast, distinguish, etc.

5. Evaluate
   The learner is able to use the knowledge to judge the value of the idea and justify the decision. The action verbs in this level include appraise, judge, value, etc.

6. Create
   The highest level of learning. The learner is able to assemble and produce a new work using the combination of previous knowledge. The action verbs in this level include build, develop, formulate, etc.

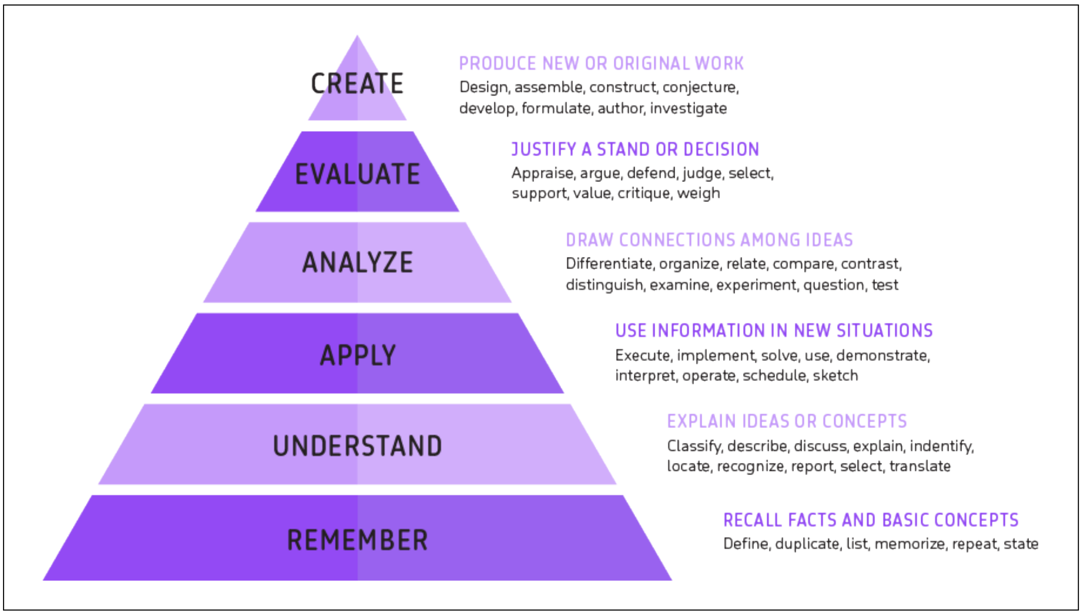


Figure 2.2: Bloom's Taxonomy with Action Verbs [6]

2.2.2 Outcome-based Education

Outcome-based education (OBE) is an education concept focusing on the learner's ability to perform the desired task at the end of the learning, or the outcome [7]. To achieve this, the curriculum is designed backward.

How?

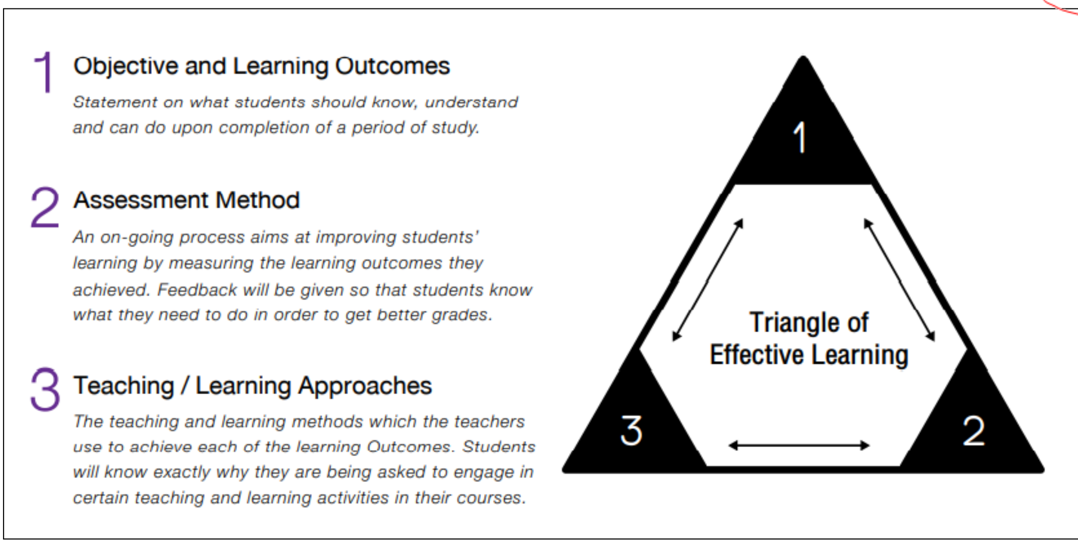


Figure 2.3: Triangle of Effective Learning [7]

According to the triangle of effective learning in Figure 2.3, the first step of designing is setting the learning outcome which is a skill we wish the learner to gain. The general format of the learning outcome is the action verb + object + qualifying phrase. An action verb can be derived from Bloom's taxonomy. A good learning outcome should match the SMART(TT) characteristics [8] which is an abbreviation for

- Speak to the learner: The outcome should specify what the learner will be able to do.
- Measurable: The outcome indicates how it will be assessed.
- Applicable: The outcome addresses the way the learner uses the gained skill.
- Realistic: The learner should be able to demonstrate the skill addressed in the outcome.
- Time-bound: The outcome should set the specific duration of the learning.
- Transparent: The learner can easily understand the outcome.
- Transferable: The skill in the outcome could be used in a wide range of contexts.

After the learning outcome is specified, design the assessment method to measure the learner's ability. The assessment criteria or a rubric must be specified. There is no need that one outcome consists of only one criterion. A good rubric should be observable and measurable.

Finally, the teaching and learning activities are designed based on such learning outcomes and assessments. This includes not only the materials but also the activities. We adopt this concept in our educational game design mainly in the education parts.

2.2.3 ADDIE Model

ADDIE model is a model for designing and developing processes used widely in many fields, including education [9]. The model provides a systematic approach to the work process consisting of analysis, design, development, implementation, and evaluation, respectively. We apply the process of this model as the process for doing the project.

1. Analysis

   The first step in the process focuses on identifying the problem, the learning environment, and the deliveries.

2. Design

   The main task of this phase is defining the learning objectives, evaluation tools, and content development to suit the analyzed problem.

3. Development

   After the design, the next step is developing the learning resources including the learning material and learning activities. The developed content can be used later for the pilot test involving stakeholders to collect the feedback necessary for content revising.

   How? ใครบ้าง

4. Implementation

   This step focuses on preparing the people involved in education and implementing the actual learning

   เช่น ? ← resources that have already been analyzed and designed. In the aspect of educational games, the main task of this step can be implementing the actual computer game software.

5. Evaluation

   The last step of the model is an evaluation. The assessment is done to ensure that the learner actually achieves the learning outcomes.

   มีอะไรบ้าง , วัดอย่างไร

/includegraphic [width =]

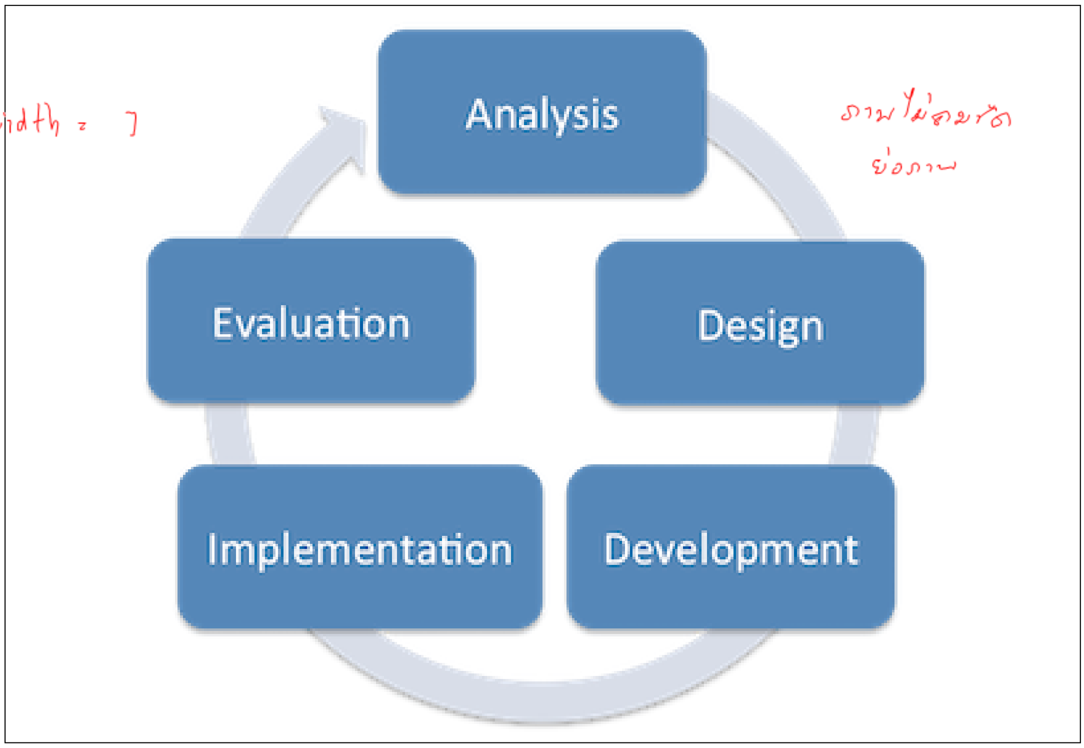


Figure 2.4: The phrases of the ADDIE Model [9]

what types ?

2.3 Genetic Algorithm

The Genetic Algorithm is the computer algorithm adapted from the Darwinian theory of Natural Selection [10]. The algorithm introduces a way to solve problems considered difficult or costly in terms of time and processing. The possible solutions to the problem must be in some data structure called a chromosome or gene. Then the genetic algorithm will be conducted repeatedly to evolve the answer to find the optimal solution. Commonly, the genetic algorithm is used to generate high-quality solutions to optimization and search problems. The flow of the algorithm is as follows.

→ How

→ GA

How ?

How ?

GA is meta-heuristic

→ near optimal

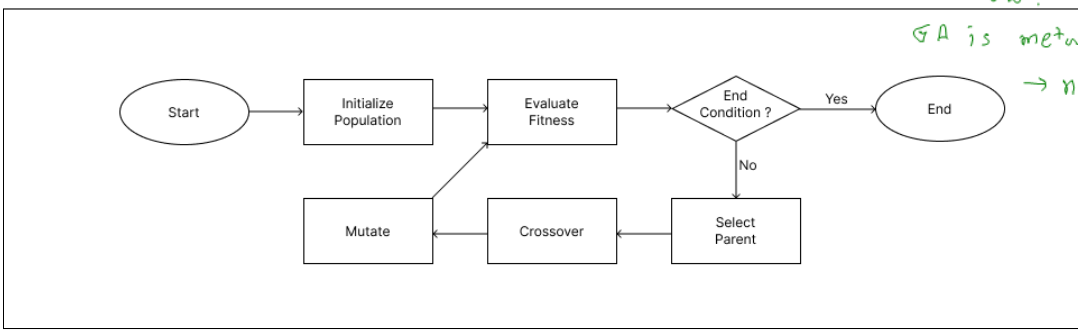


Figure 2.5: Flow Chart of Genetic Algorithm

2.3.1 Population Initialization

satisfied constraints

Usually, the population initialization will be done via randomizing the set of possible solution chromosomes. The population size must be large enough to maintain the diversity of the solution; otherwise, it will not be able to reach the global optimum solution in the end.

objective fn. and constraints

2.3.2 Fitness Evaluation

The fitness of a gene will be evaluated using a specific function called the fitness function, which is the evaluation function that indicates how close to the optimum solution of the chromosome is. In the natural selection rule, the survivor is one with the fittest properties. The chromosome with a higher fitness value tends to "survive" (selected by the algorithm), inheriting part of its chromosome to the offspring in the next generation. Meanwhile, the chromosome with lower fitness is likely to be destroyed

which process? when?

2.3.3 Genetic Operations

The population will be evolved by generating a new population of the existing ones through genetic operations consisting of selection, crossover, and mutation.

- Selection

  producing

  The subset of chromosomes will be selected from the current population as a parent for the next generation based on the fitness value. There are several basic ways to choose the parent (parents), including random selection, tournament-based selection, roulette wheel selection, and ranked-based selection. Apart from the parent selection, there is also a technique that improves the algorithm called Elitism.

  1. Random Selection

     This is the most straightforward method, which is a uniformly random selection of a chromosome out of the population. The probability of being selected is not related to the fitness value of an individual chromosome. pros & cons → review ด้วย

  2. Tournament-based Selection

     This method begins by randomly dividing the population into equal groups with the number of K chromosomes in each group. Then, the chromosome with the best fitness value will be selected from each group. pros & cons เหมาะกับปัญหาแบบไหน

     ตัวเป็นตัวแปร ใช้ k

     ใส่ figure ก็เขียนโยงไปที่ figure ด้วย

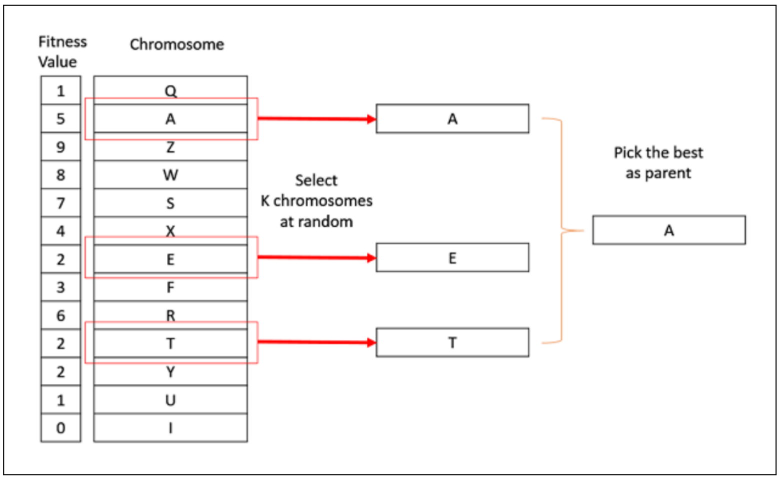


Figure 2.6: Tournament-based Selection [10]

  3. Roulette Wheel Selection

     The method is based on proportionate randomization. It can be interpreted as a Roulette Wheel where each slot represents the chance of an individual chromosome to be chosen. As the chance of a chromosome to be selected proportionate to its fitness. The higher fitness of the chromosome, the larger space on the wheel. บอกวิธีการให้ละเอียดกว่านี้ pros & cons

     ความเชื่อมโยงของ figure.

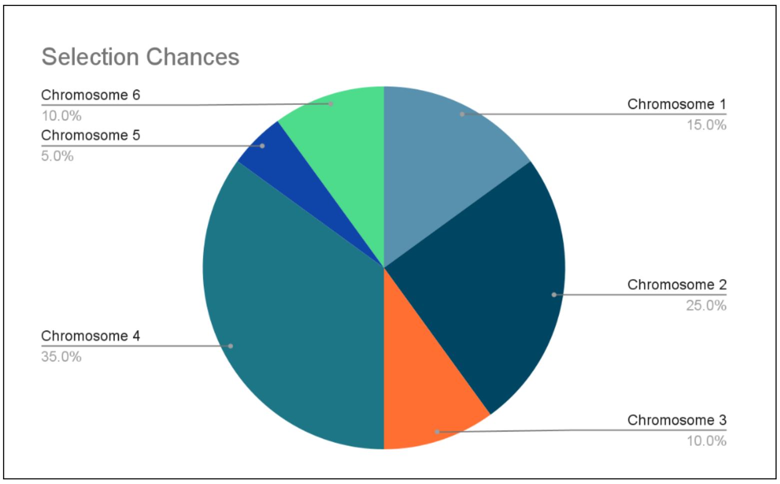


Figure 2.7: Roulette Wheel Selection

4. Rank-based Selection

   The selection method is based on the rank of the chromosome. The individual in the population is ranked according to their fitness value first. Then, assign a probability to be selected proportional to the individual rank. pros & cons

5. Elitism

   Elitism is a method that helps preserve some set of chromosomes with the highest fitness value across the generations without any change. It improves the performance of the algorithm by ensuring the preservation of the best solution. ปล่อยให้ rate เท่าไร pros & cons

- Crossover

  The crossover is the process where a pair of parent chromosomes exchanges part of their information to produce a new pair of offspring. There are several basic ways to perform the crossover, including one-point crossover, two-point crossover, and uniform crossover. PMX

  n-point crossover

  1. Single-point Crossover

     The single point from both parents' chromosomes is randomly selected as a crossover point. The gene information from each side of that point is exchanged with the other parent. The result is the pair of offspring chromosomes that inherit the gene information from both parents.

pros & cons
เหมาะกับอะไร
ความเชื่อมโยงกับภาพ

ใช้ภาพแล้วไม่เข้าใจ
อธิบายภาพ
เพิ่มรายละเอียดภาพ

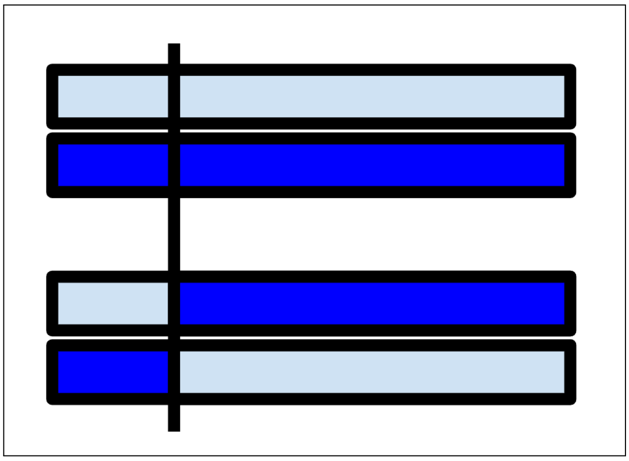


Figure 2.8: Single-point Crossover

  2. Two-point Crossover

     Two points from both parents' chromosomes are randomly selected to be crossover points. The gene information between these points is swapped between the chromosomes to create two new offspring.

pros & cons
เหมาะกับอะไร
ความเชื่อมโยงกับภาพ

ใช้ภาพมาไม่เข้าใจ
อธิบายภาพ
เพิ่มรายละเอียดภาพ

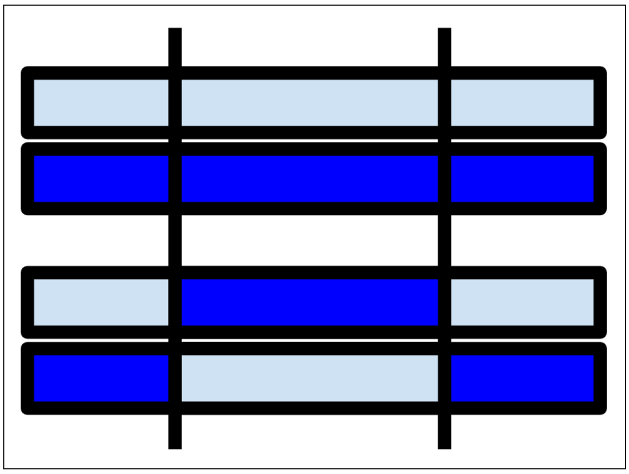


Figure 2.9: Two-point Crossover

3. Uniform Crossover
   In this type, the offspring are created by randomly swapping every piece of information in the parent' chromosome. The chance of switching each point is independent of another

pros & cons
เหมาะกับอะไร
ความเชื่อมโยงกับภาพ

ใช้ภาพมาไม่เข้าใจ
อธิบายภาพ
เพิ่มรายละเอียดภาพ

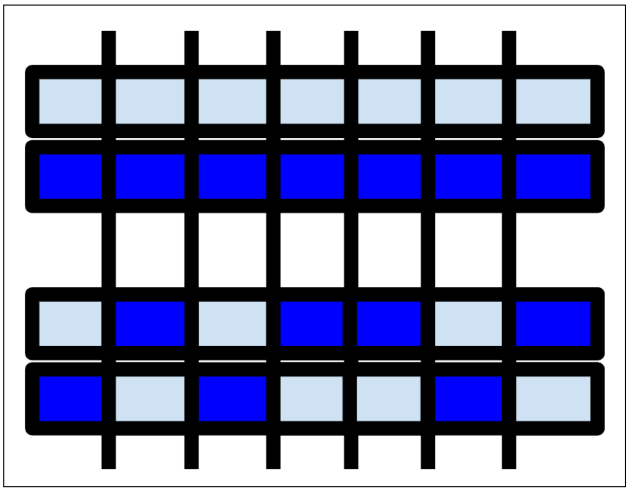


Figure 2.10: Uniform Crossover

- Mutation
  A mutation is a technique of altering the gene information of an individual chromosome. The procedure of altering the information varies and depends on how the solution is encoded. By performing this process, the chromosomes in a population can get better diversity. There are different types of mutation including bit flip mutation, flip bit mutation, boundary mutation, etc. ปกติควร set rate เท่าไร

  1. Bit Flip Mutation
     This type of mutation is used to alter the binary bit string chromosome, the chromosome consisting of only 0 and 1. The mutation can be done by randomly flipping the individual bit in the chromosome. The flipping of a bit can be done by changing its value from 0 to 1 or 1 to 0 depending on what the value it is before. For example in Figure 2.11, the highlighted bit is changed from 0 to 1. pros & cons เหมาะกับอะไร

ภาพไม่ชัด กำหนดขนาดภาพใหม่

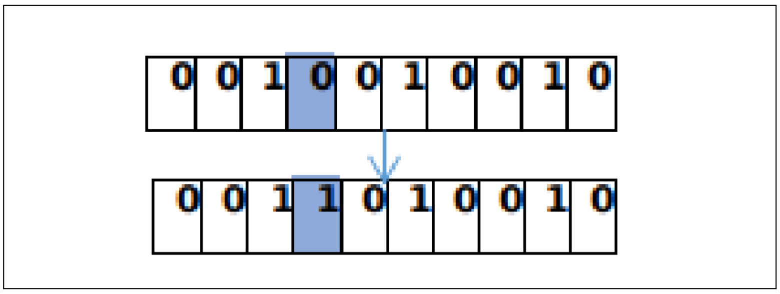


Figure 2.11: Bit Flip Mutation [10]

2. Flip Bit Mutation

   This type of mutation is also used with the binary bit string chromosome. Performing this type of mutation, instead of flipping an individual bit in an individual chromosome, the whole chromosome is randomly selected and every bit within the chromosome will be flipped. pros &cons เหมาะกับอะไร

3. Boundary Mutation

   This type of mutation suits the chromosome with an integer or float value. The process can be performed by replacing a specific value in the chromosome with the new random value of a lower or an upper range. pros & cons เหมาะกับอะไร

Knapsack

2.3.4 ~~Real-world~~ Problems → มีแต่ knapsack

The real-world problem involving our project is the knapsack problem and its variant, it's one of the most popular optimization problems that be solved using Genetic Algorithm [11]. To be accurate, the problems include Standard 0/1 Multidimensional Knapsack Problem (MKP) and Multiple 0/1 Multidimensional Knapsack Problem. The word 0/1 indicates that the problem can be encoded as a binary bit string chromosome as the solution shown in [12]. The setting and goal of each problem will be described below.

1. Standard 0/1 MKP

   Generally, the standard knapsack problem is a problem that deals with packing items in a bag while maximizing the value of the bag. Each item has its weight and value, and the bag can hold some amount of maximum weight. This maximum weight can be interpreted as a constraint that the solution must follow. The solution is invalid when the total weight of the picked items exceeds the maximum weight of the bag. The word Multidimension indicates the dimension of the knapsack or the number of constraints. For example, the Standard 0/1 MKP that has only the constraint of the weight limit can be described as only 1 dimension knapsack. The binary bit string chromosome of length equal to the number of items can be used as a solution representation. Using this type of chromosome, the value of each bit indicates whether the item with the corresponding index is selected or not.

   มีตัวอย่างจะดีมาก

2. Multiple 0/1 MKP

   The setting and the goal of the Multiple 0/1 MKP are similar to the Standard 0/1 MKP. The difference between them is the number of knapsacks in Multiple 0/1 MKP is more than one. The goal of this problem is not only the decision about whether a specific item is selected but also involves choosing the knapsack such an item should be in. The binary bit string chromosome of the length of the number of items (N) multiplied by the number of knapsacks (M) can be used as a solution representation. The first N bit in the chromosome indicates whether the item with the corresponding index is selected to be in the first knapsack or not. The next N bit of the chromosome is the indicator for the second knapsack and so on. The solution is also invalid if the same item is selected to be in more than a single knapsack.

   มีตัวอย่างจะดีมาก

2.4 Languages and Technologies

We will use multiple programs and platforms to complete this project, including Unity, the game engine of choice, visual studio, the code editor compatible with Unity, C#, the programming language for Unity, GitHub, the version control platform, and Figma, for model creation.

ความเห็นของใคร ผู้เขียนหรือทุกคน

1. C#

   C# (pronounced "See Sharp") is one of the beginner-friendly programming languages used in many game engines including Unity. It is a modern, object-oriented, and type-safe programming language based on the C family of languages similar to C, C++, Java, and JavaScript. It can be used for creating secure and robust applications, with many useful features such as nullable types, exception handling, and lambda expressions to serve multiple purposes of programming. ทำไมถึงใช้

2. Unity ✓ cite ? ทำไมใช้ Unity (เปรียบเทียบ engine อื่นด้วย (เล็กน้อยพร้อม cite)

   Unity is a popular free game engine used to create both 3D and 2D games for multiple platforms. Unity may be described as Integrated Development Environment (IDE), providing several prominent features for creating games such as physics, 3D rendering, collision detection, etc. Besides a simple drag-and-drop interface, Unity also provides a way to customize the game through C# coding. Due to its easy-to-use tools and powerful performance, Unity has become one of the most popular game engines in the world.

3. Visual Studio

   Visual Studio is the Integrated Development Environment (IDE) software for writing and editing code. It can work with various types of programming languages such as C#, C++, Python, etc. It is one of the most popular IDE to work with Unity since it includes many extensions and powerful features to C# programming.

4. GitHub

   GitHub is a code hosting platform for version control and collaboration through git repositories. Users can edit codes without harming the main project by creating a new branch to work. After committing changes, a branch can be merged back to the main branch by creating a pull request to let other collaborators review the code before pulling it into the master branch.

5. Figma

   Figma is a web-based app for designing and working with graphics. It can be used to design all kinds of graphics including websites, mobile app interfaces, prototyping designs, etc. The fact that Figma is completely free-to-use and has easy-to-use interfaces makes it to be one of the most widely-used graphic tools.

   เอามาจากไหน ใครพูด cite ด้วย

2.5 Related Works

These are some works and research relating to our project, which will be our reference.

2.5.1 Similar Games Example → ทุกเกมที่ review → ทำไม review มีความรู้จากตาม Game theory ยังไง

There are many similar games to our final project, which can be used as references or ideas for other people to fortify the images of the project.

เราได้แนวคิดอะไร เอามาใช้กับเกมเรา

1. Dragon City

   Dragon City is a free-to-play breeding simulator game developed and published by Social Point. This game lets players build a farm of dragons on floating islands, which players can breed and collect variances of dragons to battle with other players for rewards. Each dragon can be fed and evolve into a stronger dragon with many skills to learn.

Figure 2.12: Game Example: Dragon City
Source: Google Play

2. Princess Connect! Re:Dive
   Princess Connect! Re:Dive (Priconne) is a free-to-play fantasy RPG game developed by Cygames. Players can build teams of characters they pulled from gacha and use them to clear dungeons and battle with other players. Characters can be categorized into multiple groups based on various classification, such as standing position (front, middle, back) or roles (tanker, attacker, supporter). During battle, characters will battle automatically with their preset moves, and players can use their ultimate move when the corresponding gauge is filled.

Figure 2.13: Game Example: Princess Connect! Re:Dive
Source: gachafix.com

3. Omega
   Omega is a puzzle game developed by ORIGIN Systems, Inc in 1989 [13]. The game lets players program tanks by using a built-in text editor with a script that is similar to BASIC. The player can learn how to write a program by using it in the gameplay to pass the game's missions and progress through the game. This game is similar to our project because the project was designed to teach the players about the genetics algorithm by incorporating it with the gameplay and game progression and make the player learn by applying the lesson through the puzzle we designed.

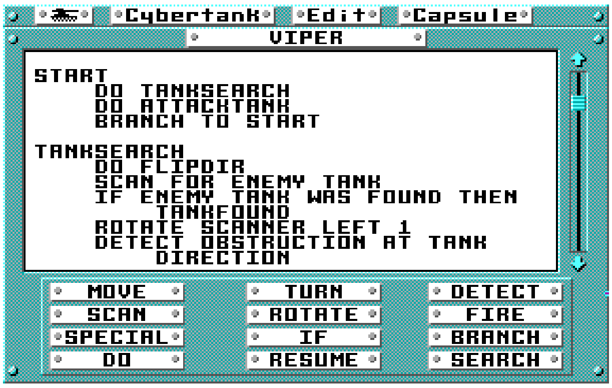


Figure 2.14: Game Example : Omega (Code Writing Section) [13]

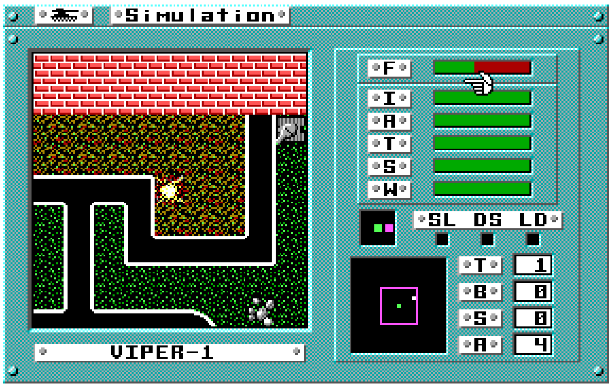


Figure 2.15: Game Example : Omega (Simulation Section) [13]

CHAPTER 3 METHODOLOGY AND DESIGN

In general, developing Digital Educational Games (DEG) has 2 main approaches. The first way is to design Computer Assisted Instruction (CAI) based on the learning topics or outcomes, then adapt it by adding the gameplay to cover the learning content to create the educational game. The second approach is reversing the first one. The game is designed first, then add a component of the learning content into the existing game to change it to be an educational game. Since we focus on the education part and already have scoped the content, we decide to develop our game using the first approach. This is one of the reasons we apply the ADDIE model in our development process.

This chapter will cover the analysis and design of both the education and gameplay aspects. The design based on the theory will be discussed. We will explain what our game contains and how it works. To clarify our idea of a design, the various diagrams and images will be also shown.

3.1 Analysis

Using the ADDIE model, the first task to do is the analysis of the problem and related detail.

3.1.1 Problem Statement

As we mentioned in the background of the project in the first chapter, we recognize that game popularity is continuously increasing, and the attention to computer knowledge should be on the same trend. But some fields of computer knowledge like the Genetic Algorithm are hard to understand. Since the game has great potential as a learning tool and there is no educational game for learning such a topic, we decided to create an educational game to help people who are interested in such a topic can learn about Genetic Algorithms simply.

3.1.2 Target Audience

why? เจาะจงเป็น นศ. ปีอะไร เรียนอะไร
มีความรู้พื้นฐานอะไรบ้าง

People who want to learn about Genetic Algorithms, especially those aged 18 years and over. The target can be divided into 2 groups which are those who have never learned this topic before and those who have studied before but want to understand more.

3.2 Design

The main task of the designing phase focuses on defining the learning-related details which we will use the Outcome-based Education concept. Since our project is an educational game, there is also the design related to the game which will be described in the format of a game design document, for example, game overview, gameplay and mechanics, and so on.

3.2.1 Outcome-based Education (OBE) Design

Applying the OBE concept, the education module is designed according to the triangle of effective learning which starts with the designing of the learning outcomes, followed by an assessment, and the teaching and learning method.

- Learning Outcomes

  As we mentioned in the scope of the project, there are several educational topics involving genetic algorithms and real-world problems that can be solved using such algorithms.

  Using the OBE concept, we must focus on the outcome in a form of the ability that the learner should

gain [7]. So, we analyze those topics, summarize them, and figure out the core concept and skill the learner should be able to do. After consideration, we set our learning outcome using the concept of OBE together with a SMART(TT) characteristics and an action verb from the level of learning in Bloom's taxonomy [6] resulting in a single ultimate outcome which is:
The learner is able to model the algorithm for solving a real-world problem using the proper problem encoding and decoding, parent selection method, crossover method, and improvement technique including elitism and mutation.
The detailed criteria and method for measuring whether the learner achieves the outcome will be described in the next assessment session.

- Assessment
  After the learning outcome is designated, the next step is designing the assessment method. The typical way of the assessment is obviously a test in a form of an exam like the multiple-choice question or the writing exam. Since we decide to develop the educational game by designing the Computer Assisted Instruction (CAI) and adding the gameplay to it later, we create the criteria based on those typical assessing methods. For the sake of clarity and measurability, we break down the learning outcome into several criteria, grade each criterion using the Likert performance scale, and specify the condition to achieve each level in the scale using the action verb with a proper and enough detail qualifying phrase. The final assessment rubric consists of 4 criteria and at most 5 levels of performance as it is shown in Table 3.1.

Table 3.1: Rubric for Assessing the Learner

Criteria	Performance descriptors				
	Level 1	Level 2	Level 3	Level 4	Level 5
The learner is able to encode the knapsack problem as the chromosome and is able to decode it.	Able to explain the problem, constraint, and goal of the Standard and Multiple Knapsack Problem.	Able to demonstrate the chromosome encoding and decoding of a Standard Knapsack Problem.	Able to demonstrate the chromosome encoding and decoding of a Multiple Knapsack Problem.	Able to solve the chromosome encoding and decoding of the Standard Knapsack Problem.	Able to solve the chromosome encoding and decoding of the Multiple Knapsack Problem.
The learner is able to solve the parent chromosome selection problem in the Genetic Algorithm.	Able to explain the meaning and purpose of a parent chromosome selection in a Genetic Algorithm.	Able to demonstrate the process of the Tournament -based Selection.	Able to demonstrate the process of the Roulette Wheel Selection and Rank-based Selection.	Able to solve parent selection problems using Tournament -based Selection.	Able to solve parent selection problems using Roulette Wheel Selection and Rank-based Selection.

Table 3.1: Rubric for Assessing the Learner

Criteria	Performance descriptors				
	Level 1	Level 2	Level 3	Level 4	Level 5
The learner is able to solve the chromosome crossover problem in the Genetic Algorithm.	Able to explain the meaning and purpose of a chromosome crossover in a Genetic Algorithm.	Able to classify between Single-point Crossover, Two-point Crossover, and Uniform Crossover.	Able to demonstrate the process of the Single-point Crossover and Two-point Crossover.	Able to solve the chromosome crossover problem using the Single-point Crossover and Two-point Crossover.	
The learner is able to identify the basic method of improvement technique in the Genetic Algorithm including elitism and mutation.	Able to explain the meaning and purpose of Elitism and mutation in Genetic Algorithm.	Able to classify between Elitism, Bit Flip Mutation, Flip Bit Mutation, and Boundary Mutation.	Able to identify the type of improvement technique including Elitism, Bit Flip Mutation, Flip Bit Mutation, and Boundary Mutation.		

The assessment rubric alone cannot evaluate the learner's performance. It needs some evidence, the result of the learner's action, to pair up with the criteria for determining the actual level of the learner. We divide the type of evidence into 3 groups based on the level of learning. The assessment activity for each group is designed based on the appropriate alignment in [14]. For example, the type of the applying outcome should be evaluated using the activity causing the learner to determine what method to use and use such procedures to solve the problem. All the types of evidence we will use are listed below.

บอกด้วยว่า
Evidence นี้
อยู่ตรงไหนของเกม

1. The evidence involving explaining, classifying, and identifying is a choice that the learner chooses for the given question or situation.
2. The evidence involving demonstrating is a record of the process or sequence of processes the learner performs for a given situation. This type of evidence can be interpreted as a short answer writing exam which is an open space where the learner is free to choose what process to perform and free to decide when to perform it.
3. The evidence involving problem-solving is an answer from the player's actions for the given question or situation. The involved activity will be developed using the activity so that the learner has a free choice of what method to use and perform those procedures freely.

All the types of evidence will be adapted to gameplay which we will explain in the gameplay and mechanic design section later.

- Learning and Teaching

  When the learning outcome and assessment method are all designed, the final task of the education planning is to design the learning and teaching including the materials and activities. The material will be developed using multimedia mostly in the style of the image and description text. Figure 3.1 to Figure 3.3 are an example of the material we have already prepared for Natural Selection, one of the basic biology concepts the Genetic Algorithm is based on.

อธิบาย
แต่ละภาพด้วย

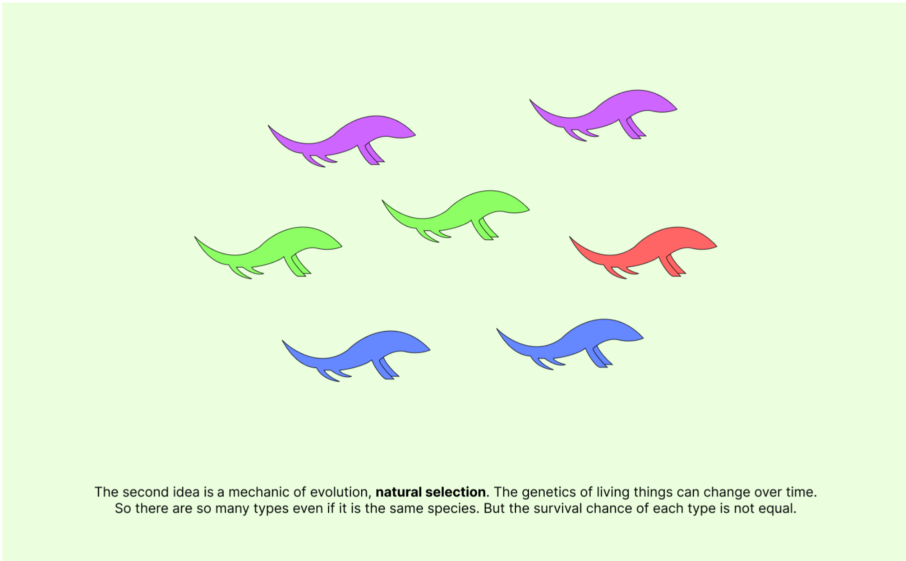


Figure 3.1: Example Material for Learning Natural Selection (1/3)

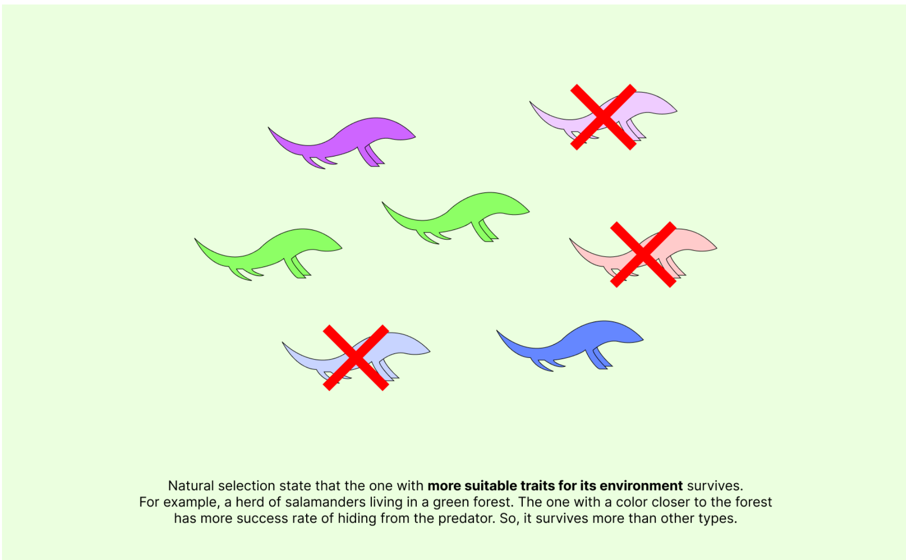


Figure 3.2: Example Material for Learning Natural Selection (2/3)

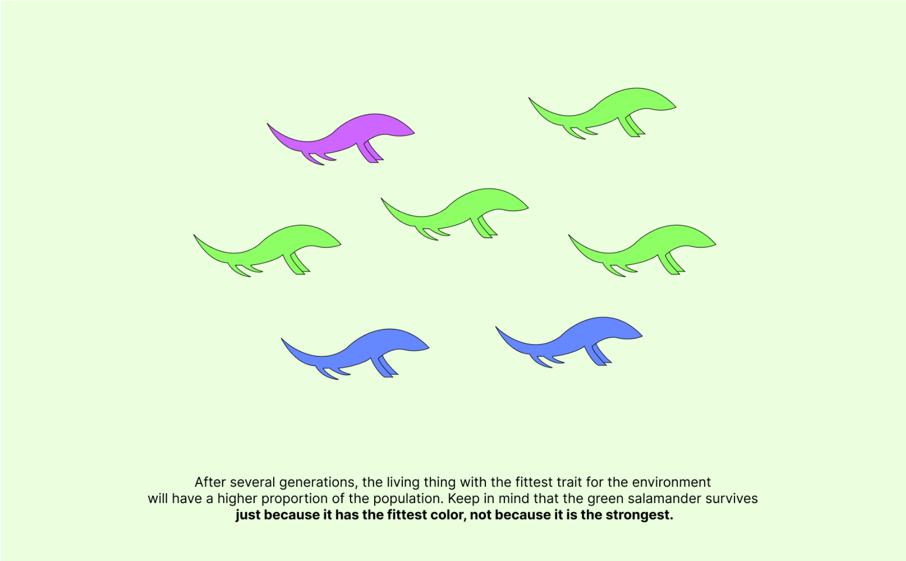


Figure 3.3: Example Material for Learning Natural Selection (3/3)

When adding the component of the game, the material will be adjusted and integrated with the game story. There also will be a Research Lab system involving the material which will be described in detail in the Game Overview section.

The next issue to concern is the teaching and learning activities. Since we aim to create an educational game, we use the great advantage of using the game, the high level of interactive activities, as the learning activity in the manner of experiential learning. To be clear, the learning activity is the puzzle gameplay similar to the gameplay used for the assessment. But in the aspect of learning and teaching, there will be a guide that tells the player what and how the puzzle works. For example, the puzzle for assessing the demonstration of the player will be evaluated based on the type of process and the sequence the player chooses. But when it is used as a learning activity, the game will guide the player on what process to choose at each time step.

3.2.2 Game Design Theory Application

Game Design Theories used for designing the game of the project are as follows:

- Natural Funativity

  The game we designed mainly consists of three of the categories from Natural Funativity by Noah Falstein. The categories we used will be listed as follows:

  1. Physical Fun

     We design the game to make players solve the puzzles by making the players move and click their mouse and use hand-eye coordination to solve the puzzles.
  2. Social Fun

     We design the game to have a story and interaction between the players and nonplayer characters (NPC).
  3. Mental Fun

     We design the game to make players solve the puzzles from the lesson and manage farms and factories with money and resources in order to progress through the game.

- Maslow's Hierarchy of Needs

  The Hierarchy of Needs we used for the game design will be listed as follows:

1. Physiological Needs
   The players need to use money and resource to breed stronger monsters in order for them to survive the battle inside the battle arena.
2. Safety Needs
   The players can improve the battle status of their monsters by breeding them to create a new generation of monsters and the players can improve the status of the weapon from the factories by creating a new generation of weapons.
3. Love and Belonging Needs
   The player can interact and talk to the nonplayer character in the game story that we have designed. The player can have a connection and feel more engaged in the story.
4. Esteem Needs
   The game will assign rank and reward for the success of the quest completion by the quality or closeness of the requested item and sent the item. The players can feel accomplished when getting high ranking and feel accomplished when finishing the game after meeting the requirements of all weapons from the factories.
5. Cognitive Needs
   The game was designed to teach players about genetic algorithms and how to solve the problem with this knowledge.
6. Aesthetic Needs
   The game will have a beautiful aesthetic with music to make it more pleasing to play for the players. The higher the ranking for completing the quest the more beautiful the rank is and the more powerful the weapon is the more elaborate and better it looks.
7. Self-Actualization Needs
   The game was designed to be open to the players' play style by not limiting how the players manage the farm, factories, money, and resources. The players can choose how to spend their resources and how to obtain them by themselves.
8. Transcendence Needs
   When the players developed their own technique on how to play the game, the player can share their knowledge with their peers by talking, video capturing, etc.

- Eight Kinds of "Fun"
  The kind of fun that the game we design provided will be listed as follows:

  1. Sensation - Game as sense-pleasure
     The game will have a sci-fi theme aesthetic to its visual, characters, music, and sound effect from the players' interactions.
  2. Fantasy - Game as make-believe
     The game will have its own story, world, and characters to help the players feel immersed in the game we designed. The player will have a role to take and objectives to accomplish in the story.
  3. Narrative - Game as drama
     The players will take the role of the owner of a weapon company in the future. The world will be invaded by aliens and the government orders every weapon company to create a set of ultimate weapons.

4. Challenge - Game as obstacle course
   In order for the players to manage and upgrade their farms and factories, they need to complete the quest which will be harder when the game progresses or battle in the arena which will have more difficult opponents over time.
5. Fellowship - Game as social framework
   The game is a single-player game but the players can interact and talk to nonplayer characters in the story and from the lesson.
6. Discovery - Game as uncharted territory
   The players can unlock more weapon factories to make different kinds of weapons with different effects when used in the battle arena.
7. Expression - Game as self-discovery
   The players are free to spend their money and resources and can choose how to obtain more money by completing quests or battles in the battle arena.
8. Submission - Game as pastime
   The player can choose to not send their monster to the arena and play the game by completing the quest and managing the farm and factories.

3.2.3 Game Overview

This topic involves game concepts overall and every main system in detail with diagrams for better visualization.

- Game Concept
  Our game will focus on three main aspects: monster farm, weapon factory, and research lab. The player will own a farm to breed monsters used in the arena; rewarded with money and resources to upgrade and maintain the farm.
  The monster farm is based on the Genetic Algorithm, which will be covered in detail in the next topic. The farming system will let players take control of the farm and breed monsters with the Genetic Algorithm to increase their battle status used in the arena.
  The weapon factory is based on how to solve real-world problems with the Genetic Algorithm. The factory can help increase monsters' battle status but it can be broken down over time so the player needs to use their knowledge about how to solve real-world problems with the Genetic Algorithm to fix the factory.
  There is also an educational section called research lab that will teach players about the basics of the Genetic Algorithm, how the algorithm works, and how to solve real-world problems with it. Player needs to use certain resources to unlock new topics which also unlock new factories when the topic is about real-world problems.

- Game Flow Summary
  The game begins with the prologue story which some of the learning content is integrated. The player will be introduced to each system in the game, especially the core system, the research lab, which is used for learning and unlocking new facilities.
  After the tutorial ends, the player is free to do any activities. But in the early game, most farms and factories will be locked. The player needs to complete certain research to get a certificate or the right to build a facility. Figure 3.4 show the main game page which some facility is unlocked. After players unlock and build the facility, they can use that facility in the way their want.

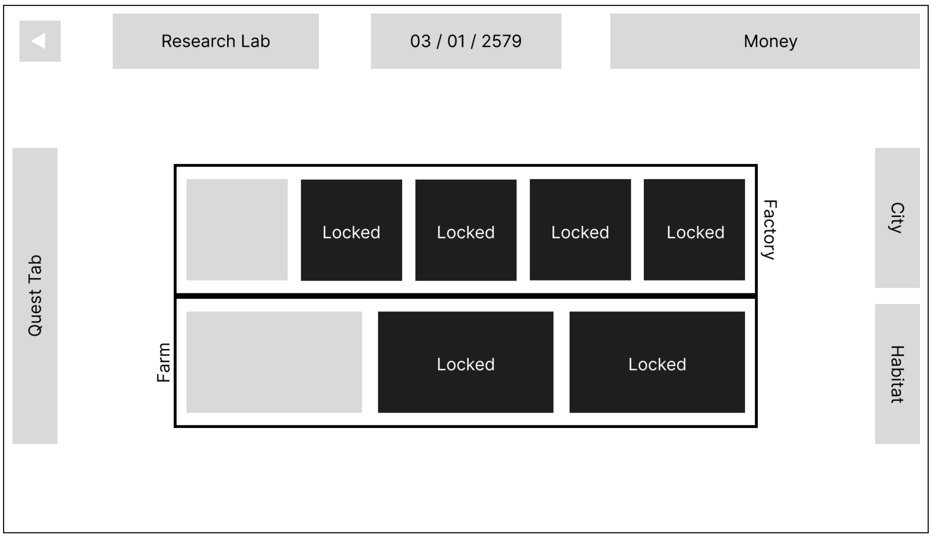


Figure 3.4: Main Game Page Interface

When? , How ?

As the game continues, the facilities would break down. The player will be asked to fix the facilities through puzzle-solving. This is where the player applies their knowledge and is assessed educationally. Figure 3.5 shows an example of the puzzle the player needs to solve to fix the facility.

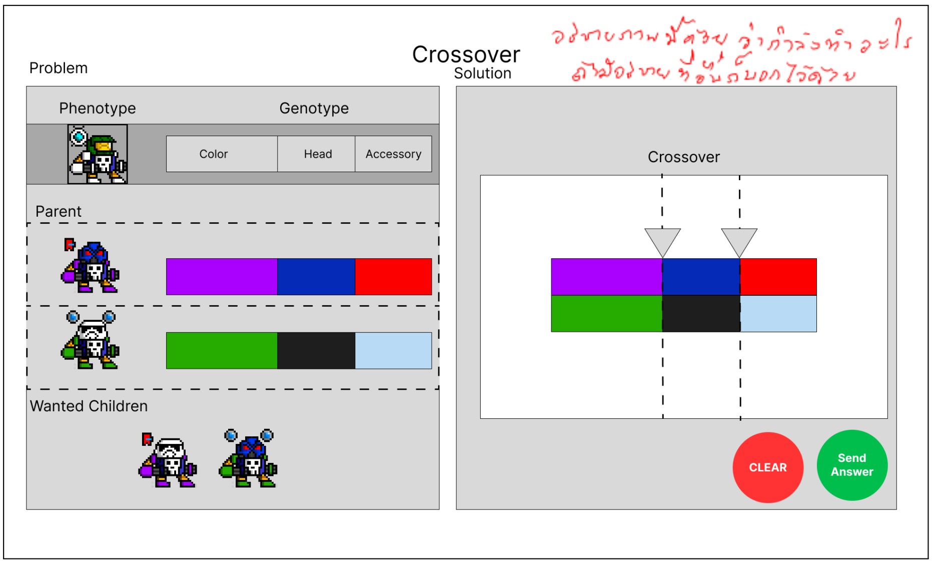


Figure 3.5: Example of Puzzle Gameplay

Besides the puzzle gameplay involving the research lab, monster farm and weapon factory, the player can go to the city to receive the quest, buy a new monster or the weapon chromosome, and battle in the arena using their own monster. Figure 3.6 shows an example of the city interface.

ในภาพมีอะไร
ทำอะไรได้บ้าง
ตัวอย่างแล้วก็บอกว่าอธิบายที่ไหน

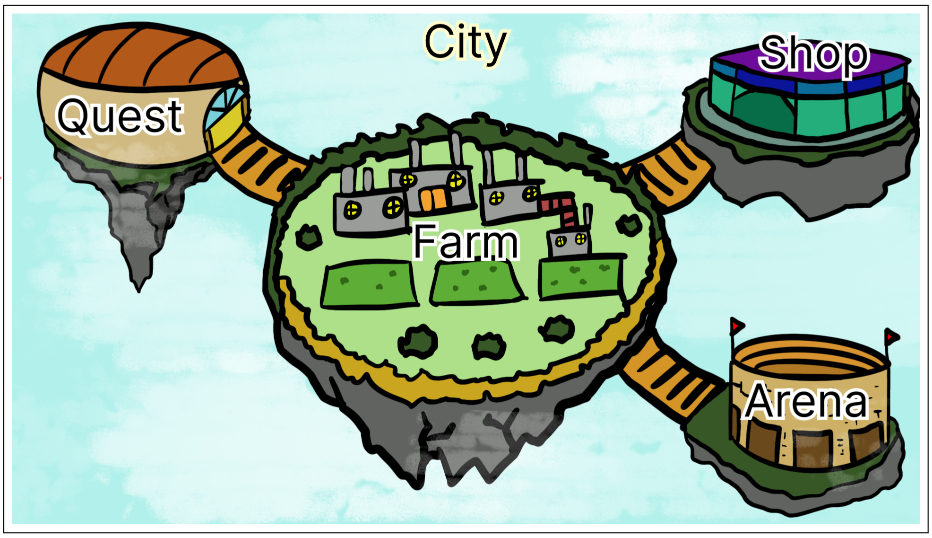


Figure 3.6: City Interface

When the game progresses to the late game, the last condition to indicate whether the player wins the game is the main quest that requires the player to deliver a very good weapon from every factory. If the player completes the main quest in time, the player wins the game. Otherwise, the player loses.

- System Details

  The game systems can be described as states of the game. The states of the game and its transition are shown the Figure 3.7 below.

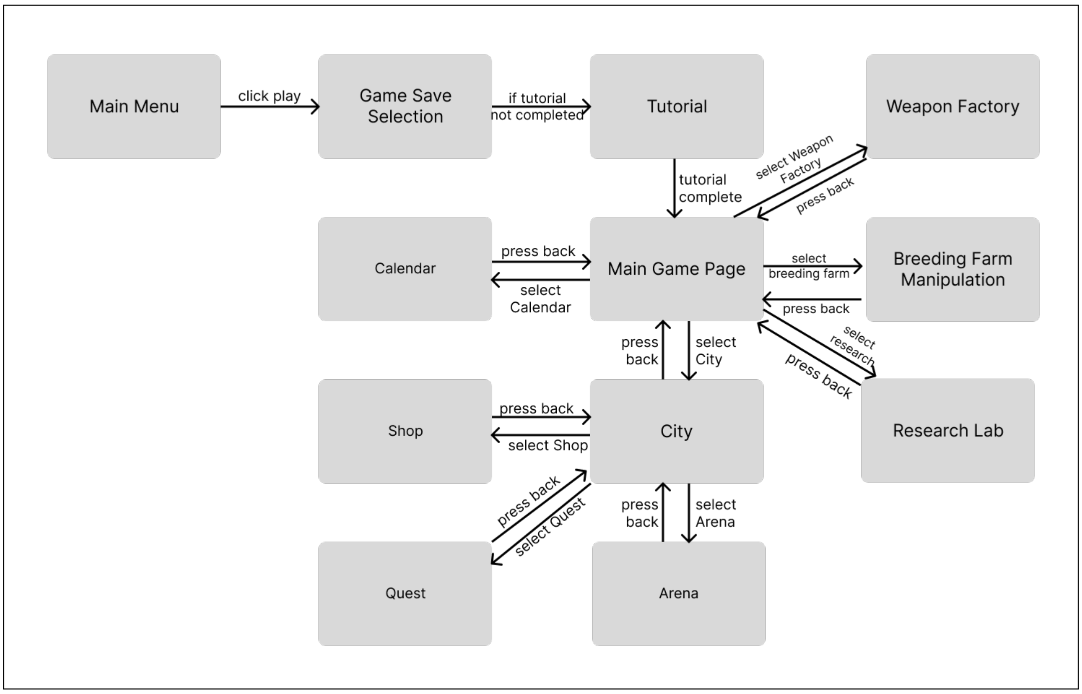


Figure 3.7: Overall Game State Diagram

1. Research Lab System

   This is a system where players can learn about the Genetic Algorithm and how to solve Real World Problems with the Genetic Algorithm. The player must pass a requirement on a specific topic to unlock the particular customization in their breeding farm, such as a roulette wheel selection. All the categories, learning topics, and learning content in the research lab are based on the OBE education design and will be shown in Table 3.2.

Table 3.2: The Category, Topic, and Content in Research Lab System

Category	Topic	Content
Basic Biology	On the Origin of Species	The basic concept of Evolution and Natural Selection based on Darwin's theory.
	The Genetic	The explanation of the chromosome, gene, phenotype, genotype, and biological crossover.
Genetic Algorithm	The Flow of Genetic Algorithm	The introduction to the Genetic Algorithm, the brief description of each step in the flowchart.
	Parent Selection	The process detail of the parent selection in Genetic Algorithm include Random Selection, Tournament-based Selection, Roulette Wheel Selection, and Rank-based Selection.
	Crossover	The process detail of the crossover in Genetic Algorithm include Single-point Crossover, Two-point Crossover, and Uniform Crossover.
	Improvement Techniques	The description of the improvement techniques in Genetic Algorithm include Elitism and the basic variant mutation which consist of Bit Flip Mutation, Flip Bit Mutation, and Boundary Mutation
Real-world Problems	Standard 0/1 MKP	The problem setting, goal, chromosome encoding, and fitness evaluation of the Standard 0/1 MKP will be described.
	Multiple 0/1 MKP	The problem setting, goal, chromosome encoding, and fitness evaluation of the Multiple 0/1 MKP will be described.

2. Monster Farm System

   This is a Genetic Algorithm simulation for monster reproduction in farms. The individual monster will be encoded as a chromosome consisting of 2 sections of a gene, each section representing each aspect of a monster including an appearance and battle status. The appearance is their shape and color. The battle status includes attack, defense, health points, and speed.

เกี่ยวกันยังไง

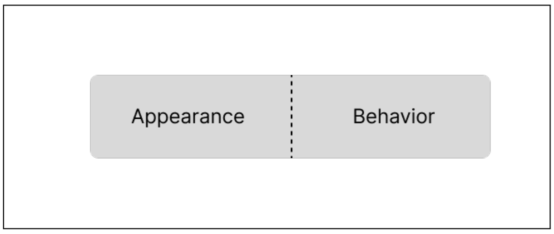


Figure 3.8: Chromosome Representation of the Monster

Players can complete research from the research lab for the farm to unlock new parent selections and new crossover methods and choose what setting to use in the farm. Besides the basic operation of the Genetic Algorithm, the player can use elitism to adapt their favorite monster marking which preserves its chromosome for the next generation. The number of monsters on a farm and the number of breeding generations affect the amount of currency consumed. Such currency is acquired from the battle in the arena, forcing players to learn and constantly manage their farm and use their Genetic Algorithm knowledge.

เกี่ยวกันยังไง

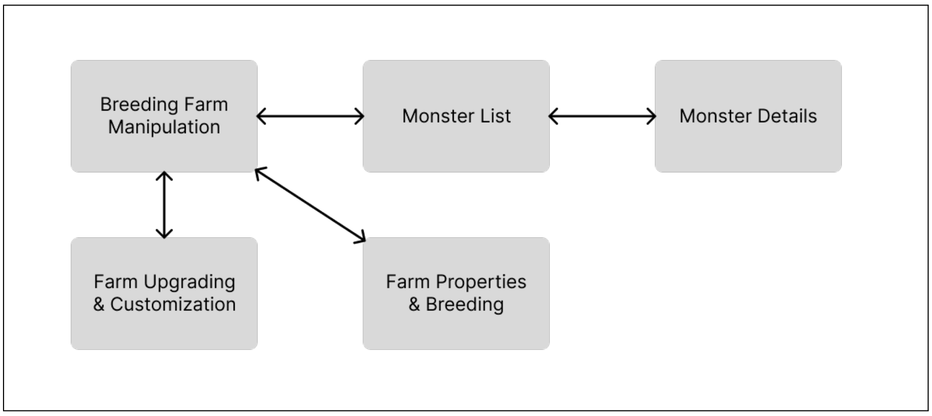


Figure 3.9: Breeding Farm Manipulation State Transition Diagram

Table 3.3: Farms Unlock Requirements

Farm	Unlock Requirement
Farm 1	Complete Research - On the Origin of Species - The Genetic - The Flow of Genetic Algorithm - Random Selection - Single-point Crossover - Elitism - Boundary Mutation Money
Farm 2	Unlock Factory 3 Money
Farm 3	Unlock Factory 5 Money

The player must meet the requirements on the table 3.3 to unlock more Farms.

3. Weapon Factory System
   This is a system where players can test themselves on how to solve Real-World Problems with the Genetic Algorithm.

Table 3.4: Weapon Factories Unlock Requirements

Weapon Factory	Unlock Requirement
Factory 1	Unlock Farm 1 Complete Research - Standard 0/1 MKP - Bit Flip Mutation Money

Table 3.4: Weapon Factories Unlock Requirements

Weapon Factory	Unlock Requirement
Factory 2	Unlock Factory 1 Complete Research - Tournament-based Selection - Flip Bit Mutation Money
Factory 3	Unlock Factory 2 Complete Research - Multiple 0/1 MKP - Roulette Wheel Selection - Two-point Crossover Money
Factory 4	Unlock Factory 3 Complete Research - Rank-based Selection - Uniform Crossover Money
Factory 5	Unlock Factory 4 Money

The player must meet the requirements on the Table 3.4 to unlock more factories. The factory can enhance monsters' battle status and make them stronger in battle.

4. Arena System

   This game provides places for players to use their bred monsters to earn more resources in order to maintain their farms and factories; one of them is the arena system. Players can create a team of three monsters equipped with weapons from the factory to battle with enemy teams.

เกี่ยวกันยังไง

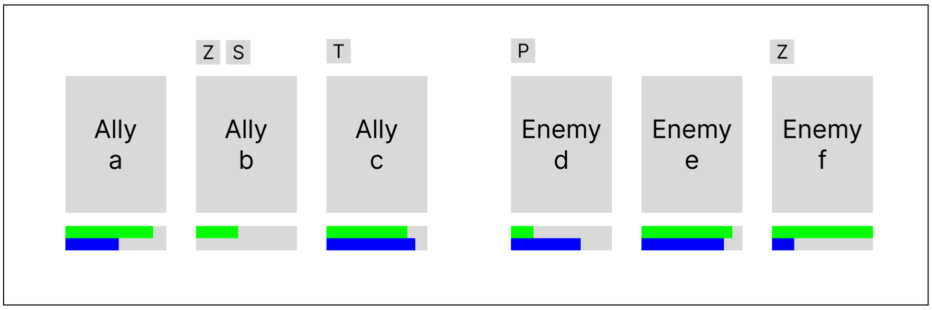


Figure 3.10: Player team (left) battling enemy team (right)

   Each monster will attack automatically after the speed gauge is filled (blue gauge). Attack targets will get chosen randomly depending on their corresponding space. If the health gauge runs out, that monster is out of combat. The battle will end when the whole team is unable to battle. There is a time limit to this combat; if time runs out but the battle does not finish, the overtime phase will begin, and every monster will get a boost for a small period of time. After the overtime phase, the team with more health will win. Finally, the reward given to players of that battle depends on the result.

5. Side Quest System

   Another system for players to gain resources is the side quest system. There are requests from various sources for players to fulfill by submitting monsters or weapons satisfying their requirements. The resemblance of those objects will affect the number of rewards directly. If players cannot accomplish quests they received within a time limit, those quests will fail.

example figure

6. Shop System
   After submitting quests, the population of monsters and weapons will decrease, and players could buy new ones from the shop to regain the population. The shop will refresh weekly with a new batch of randomized products for players to buy with money and resources.

7. Calendar System
   Time in this game will progress when players decide to end their day. The calendar system displays many details related to date and time, such as quest submission date, breeding completion date, and other events. The game will automatically save when players end their day, and the next day will begin.

3.2.4 Gameplay and Mechanics

Within this topic, the gameplay, main mechanics, and the replaying and saving system of our game will be explained.

- Gameplay
  In this topic, we will describe how we progress the game and what the structure of both the mission and the puzzle is like.

  1. Game Progression
     In this game, players have four years to develop the most powerful weapons to defend the earth. As the game progresses, players will have to manufacture weapons in their respective factories for the government to use.
     In the beginning, players will start with a brand new company, so they have to unlock new facilities (monster farms and weapon factories) by themselves. To do that, they must satisfy specific requirements, such as learning corresponding lessons about the genetic algorithm and real-world problems, having enough resources, and passing missions.
     When players finish a lesson, they must take an exam to check their understanding of that topic, and they might have to retake that lesson if they don't pass. Players can order each facility to manufacture new generations of its corresponding product. Maintaining factories requires money, so players also have to gather those by fulfilling requests from others or battling in the arena. To finish a request, they have to submit products from their company, resulting in a reduction in the overall population, which can be solved by buying new ones from the shop.
     After breeding the most powerful weapon possible of that factory, an apex, the government will buy every product of that factory, resetting everything. If players can submit apexes from every factory before the four years time limit reaches, they win the game.

  2. Mission Structure
     There are two types of missions in this game, main quest and side quest. Main quests will help players progress the game, and side quests will provide players with money and resources to maintain their facilities.
     Players are given one main quest at a time as a guideline to tell players what they could do next, such as unlocking new facilities, learning new topics, or gaining some specific resources.
     Side quests are requests from various sources ordering products from players' companies. The

reward quality depends on how the submitted product resembles the requirement of that order. If players cannot finish the request within a time limit, that side quest will fail, and players will receive a penalty in money.

3. Puzzle/Test Structure

   The game will use puzzles and tests to evaluate the players about the knowledge and understanding of the lessons as in the assessment Table 3.1 and use it as exercises to help players apply their knowledge after learning them.

   – Puzzle/Test Appearance

     The players will encounter puzzle in 2 different situations as following:

     (a) After the players finish learning the lesson from the research lab, The players will have a button to solve the puzzle to complete the lesson.

     (b) When the farms or factories break down, the players will have to solve the puzzle to fix and continue to use it.

   – Type of Puzzle/Test

     The puzzle/test that the players will encounter will have 3 types of puzzle/test as following:

     (a) Questions Test

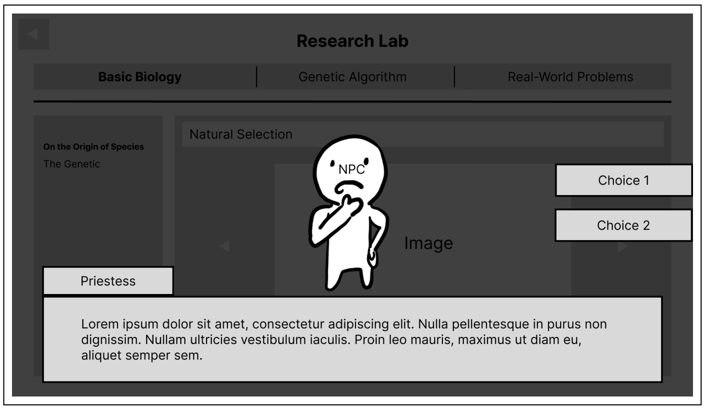


Figure 3.11: The Example of Questions Test Scene

In Figure 3.11, the players will have to choose the answer from the question. This type of test is for testing the players about how they explain, classify, or identify the lessons they have learned.

     (b) Demonstration

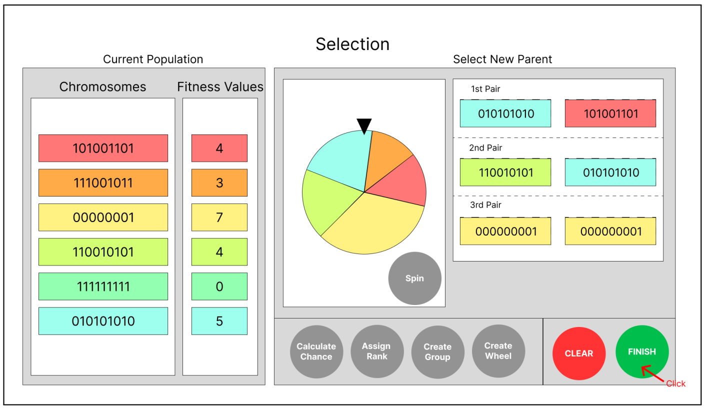


Figure 3.12: The Example of Demonstration Puzzle

In Figure 3.12, the puzzle/test is an open space where the learner is free to choose what process to perform and free to decide when to perform it. The puzzle will record the process the players perform to evaluate the players' understanding.

(c) Problem Solving

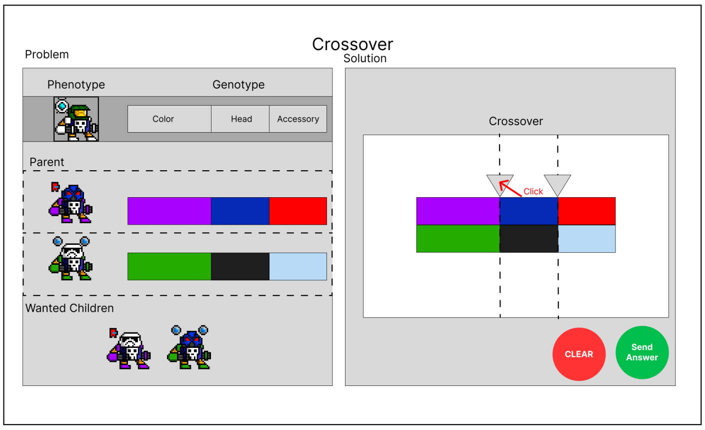


Figure 3.13: The Example of Problem Solving Puzzle

In Figure 3.13, the players will have to perform the action to solve the problem from the question or situation. The player is free to decide what method they will use and how the procedures perform. Designing this way, the puzzle would be able to assess the true learning level of the player [14].

- Mechanics

  Besides the mechanics described in the system detail section, the summary of the action the player can perform in our game will be described.

  1. Use Cases

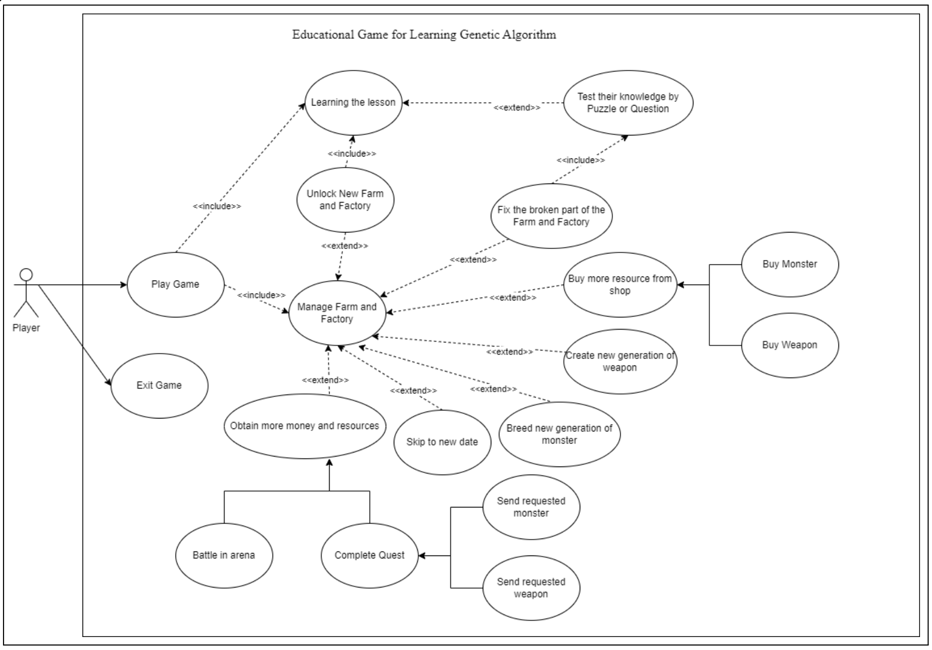


Figure 3.14: Use Cases Diagram

Players can start playing or exit the game when starting the program. After players choose to play the game, they can learn the lesson from the research lab or managing farms and factories in the main gameplay scene. After finishing a lesson, players must take a test to evaluate their knowledge of that topic. Also, they can unlock new facilities by fulfilling some requirements such as money, resources, and learning lessons. When farms and factories break, players have to solve puzzles using knowledge of the genetic algorithm to fix the broken part. Players can breed a new generation of monsters and weapons from corresponding farms and factories using their money or resources. They can buy more monsters or weaponry from the shop to increase the population in farms and factories. Maintaining facilities requires spending resources, so players have to manage and earn more in some way. In order to do that, they can build a team of monsters equipped with weapons to battle in the arena; or satisfy requests from people ordering monsters or weapons from players' companies.

2. Economy
   Money is the main resource in this game. It is used in many places, including maintaining the company, purchasing items, and entering the arena. Players can earn money by selling products and winning rewards from the arena.

- Replaying and Saving
  According to the calendar system, the game will save when players end the day and move on to the next day. Players can exit the game freely, but the game will start from the beginning of the day after the saving occurs.

CHAPTER 4 PRELIMINARY RESULTS

In this chapter, we will introduce the changing in our project. The problem and its solution will be described.

4.1 Problems and Solution in Term 1

During the operation in term 1, we encountered and solved several problem. There are three main problems that has a high impact and cause the major change on the project including project planning, workload management, and educational game development framework. All these problems will be described in details as follows.

1. Project Planning Problem
   After using the waterfall model for some time, we found that it was difficult to arrange any plans due to a lack of educational game development experience. As a result, we could not make detailed plans, and the work continued derailing.
   Therefore, we decided to switch to a more flexible development process, which is Scrum. With its agile value, the workflow becomes more versatile, and we can recheck and adjust the work process or any component freely.

2. Workload Management Problem
   First of all, we created a product backlog by listing the requirements of the project based on the objectives of players and developers by adjusting some topics from the waterfall model and adding new subjects from various viewpoints.

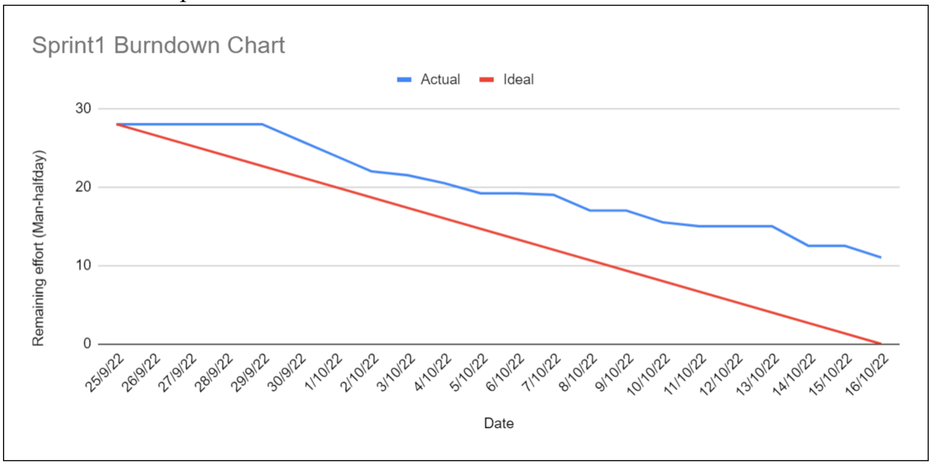


Figure 4.1: Sprint 1 Burndown Chart

   In the first sprint, we added too many scopes, causing the workload to become too overwhelming as in Figure 4.1. Consequently, we finished only some of the workloads and left it to the next sprint. Moreover, with the newly added group member, we decided to change the project scope and introduce five new real-world problems into the project.

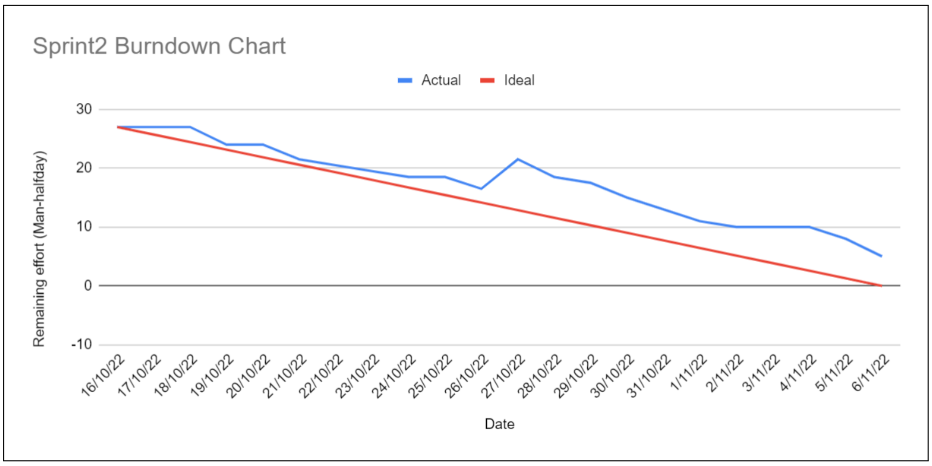


Figure 4.2: Sprint 2 Burndown Chart

The second sprint was improved from the last one, as we had one more member to work with and less workload picked for the sprint. However, we were ordered to deliver a new proposal and presentation for the new scope of work, resulting in an increased workload around the middle of the sprint as in Figure 4.2.

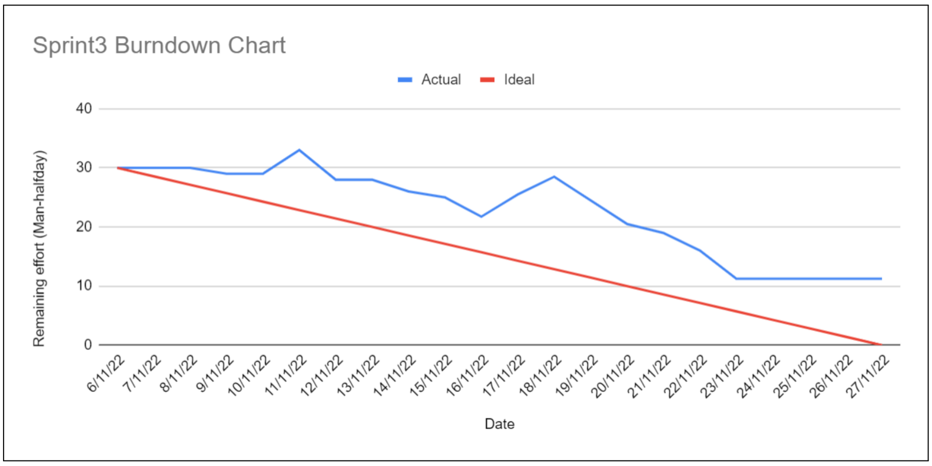


Figure 4.3: Sprint 3 Burndown Chart

บอกด้วยว่าทำไมลด real-world problem จาก 5 เป็น 1

Due to multiple unfinished works from the last two sprints, most of the workload in this sprint was to finish those leftover works. Nevertheless, we have communicated with the project advisors, suggesting decreasing the project scope to an acceptable amount. After some consideration, we discarded all but one real-world problem, the knapsack problem, and added more variants of that problem instead. Adjusting the scope also force we to redesign the large fraction of education section which resulting in a lot of increasing workload as is in Figure 4.3. Note that the burndown chart in Figure 4.3 is most updated at 23 November 2022.

3. Educational Game Development Framework Problem

   In the first two sprints, we designed the game by prioritizing educational topics over game elements via lecture-based learning; making it harder to design the gameplay and adapt those predesigned lectures into the game.

   In a lecture in the game design class (CPE467), we learned about educational game development frameworks, which are OBE and ADDIE models, so we decided to use them in our project. As a result, we could design the educational section of the game better than last time and integrate them into the gameplay more seamlessly by prioritizing which outcomes learners should have instead of what to teach them. Regardless, designing every lecture from the beginning delayed the overall design progress.

REFERENCES

1. Marcello A Gómez-Maureira, Max van Duijn, Carolien Rieffe, and Aske Plaat, 2022, "Academic Games-Mapping the Use of Video Games in Research Contexts," 2022.

2. Chioma Udeozor, Ryo Toyoda, Fernando Russo Abegão, and Jarka Glassey, 2022, "Digital games in engineering education: systematic review and future trends," **European Journal of Engineering Education**, pp. 1–19, 2022.

3. Steve Rabin, Ed., 2005, **Introduction to game development**, chapter 2, Hingham, MA : Charles River Media.

4. Saul McLeod, 2022, "Maslow's Hierarchy of Needs," https://www.simplypsychology.org/maslow.html#:~:text=From%20the%20bottom%20of%20the,attend%20to%20needs%20higher%20up., Last accessed 16 November 2022.

5. Chris Sniezak, 2016, "The Eight Tyeps of Fun," https://gnomestew.com/the-eight-types-of-fun/, Last accessed 16 November 2022.

6. Christine Persaud, 2021, "Bloom's Taxonomy: The Ultimate Guide," https://tophat.com/blog/blooms-taxonomy/, Last accessed 15 November 2022.

7. KMUTT C4ED, "Outcome Based Education," https://cilt.wu.ac.th/backEnd/myfile/attUKPSF/OBE%20WorkSheet%20CUPT.pdf, Last accessed 15 November 2022.

8. Regine Rafer, "CHARACTERISTICS OF GOOD LEARNING OUTCOMES," https://www.academia.edu/33889461/CHARACTERISTICS_OF_GOOD_LEARNING_OUTCOMES, Last accessed 24 November 2022.

9. ISFET, "What is the ADDIE Model?," https://www.isfet.org/pages/addie-model, Last accessed 15 November 2022.

10. Savio D Immanuel and Udit Kr Chakraborty, 2019, "Genetic algorithm: an approach on optimization," in **2019 International Conference on Communication and Electronics Systems (ICCES)**. IEEE, 2019, pp. 701–708.

11. Soukaina Laabadi, Mohamed Naimi, Hassan El Amri, Boujemâa Achchab, et al., 2018, "The 0/1 multidimensional knapsack problem and its variants: a survey of practical models and heuristic approaches," **American Journal of Operations Research**, vol. 8, no. 05, pp. 395, 2018.

12. Bhayani Arpit, 2022, "Genetic Algorithm to solve the Knapsack Problem," https://medium.com/@pokhreltejaswee/using-genetic-algorithms-to-solve-equations-8310b216e7a2, Last accessed 6 November 2022.

13. MobyGames, "Omega," https://www.mobygames.com/game/omega_, Last accessed 15 November 2022.

14. Eberly Center, 2022, "Why should assessments, learning objectives, and instructional strategies be aligned?," https://www.cmu.edu/teaching/assessment/basics/alignment.html, Last accessed 24 November 2022.

APPENDIX A
STORYBOARD

Storyboard

This appendix shows the overall storyboard of the game. The description will be attached to each scene.

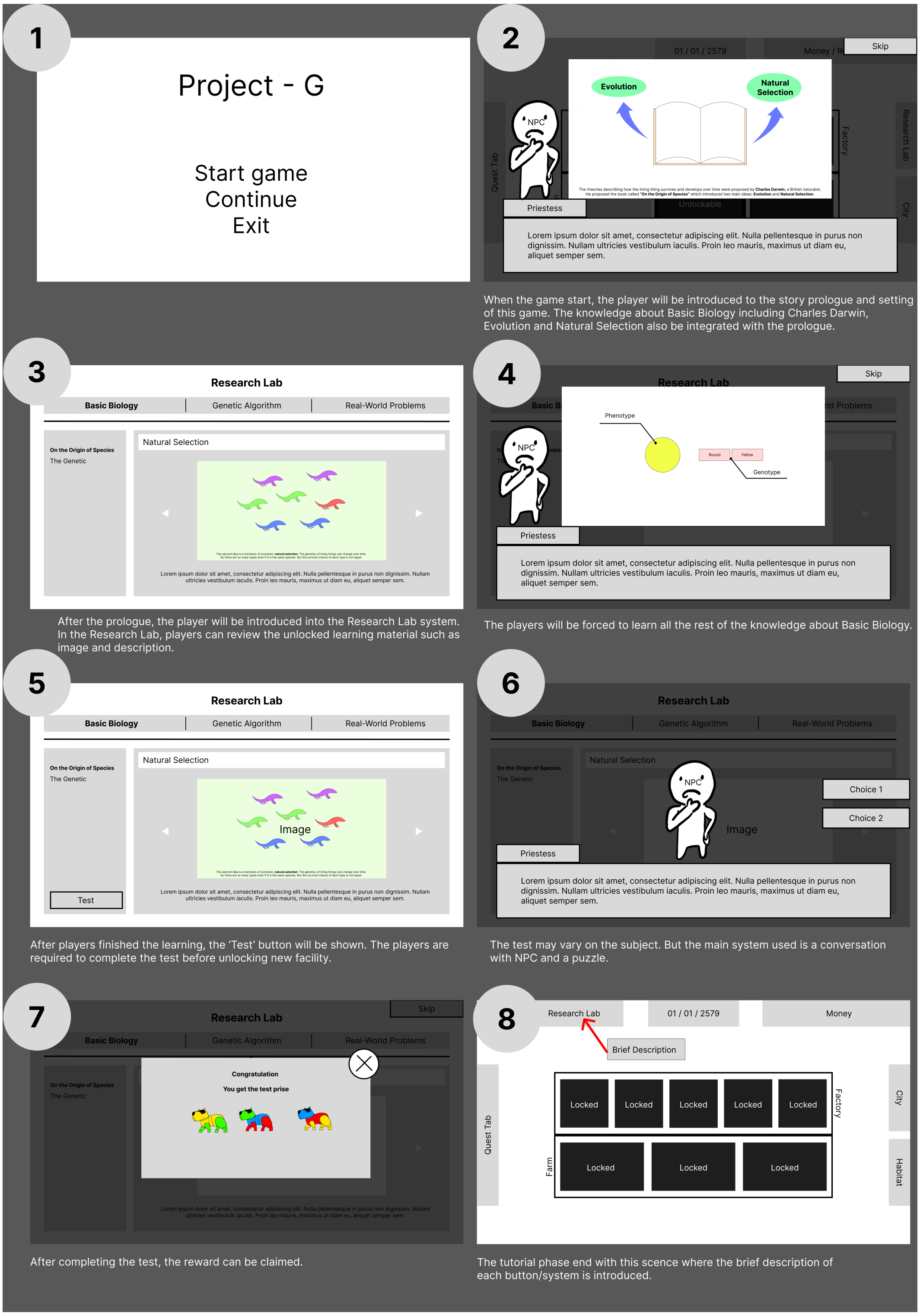


Figure A.1: Storyboard part 1 out of 4

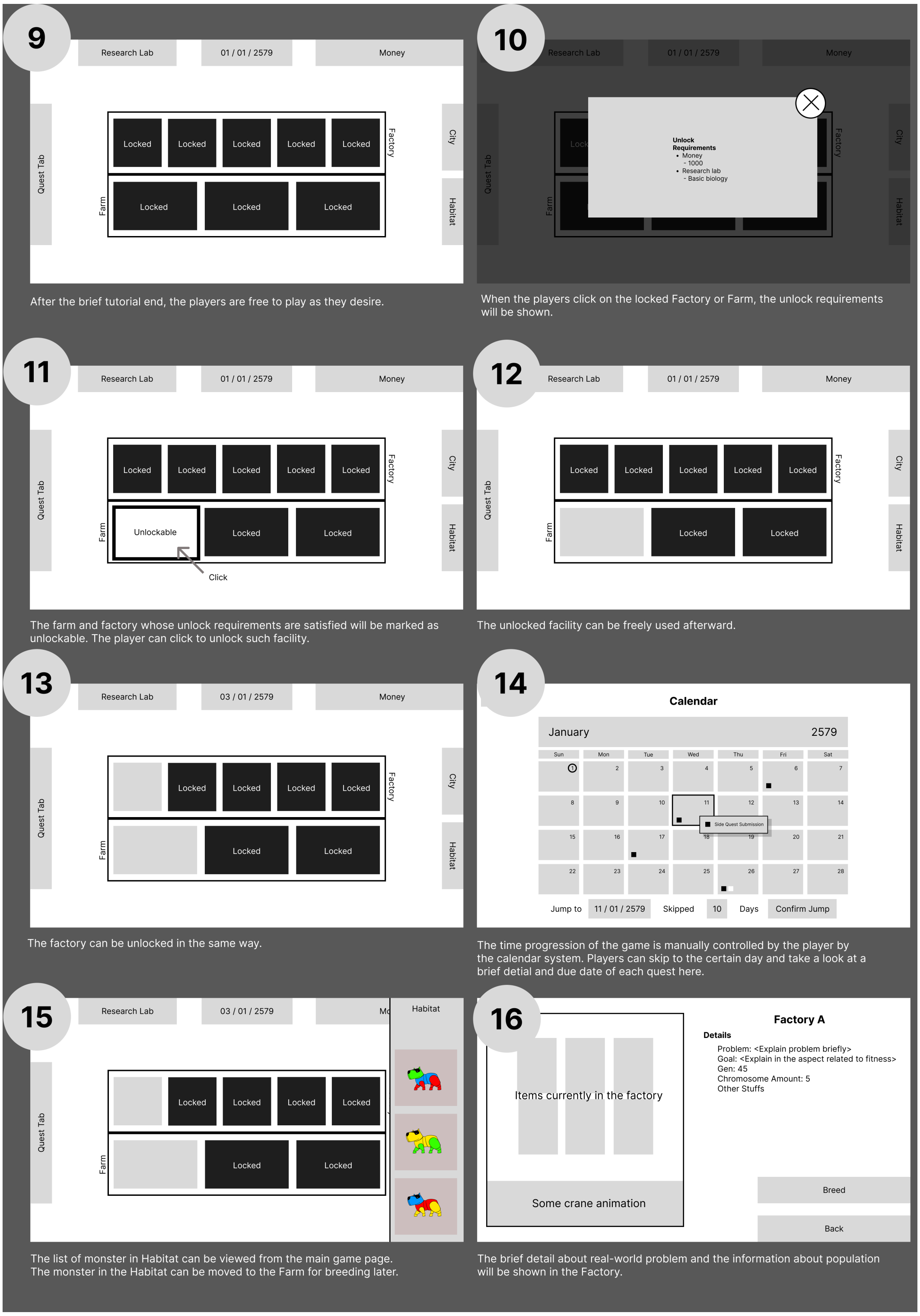


Figure A.2: Storyboard part 2 out of 4

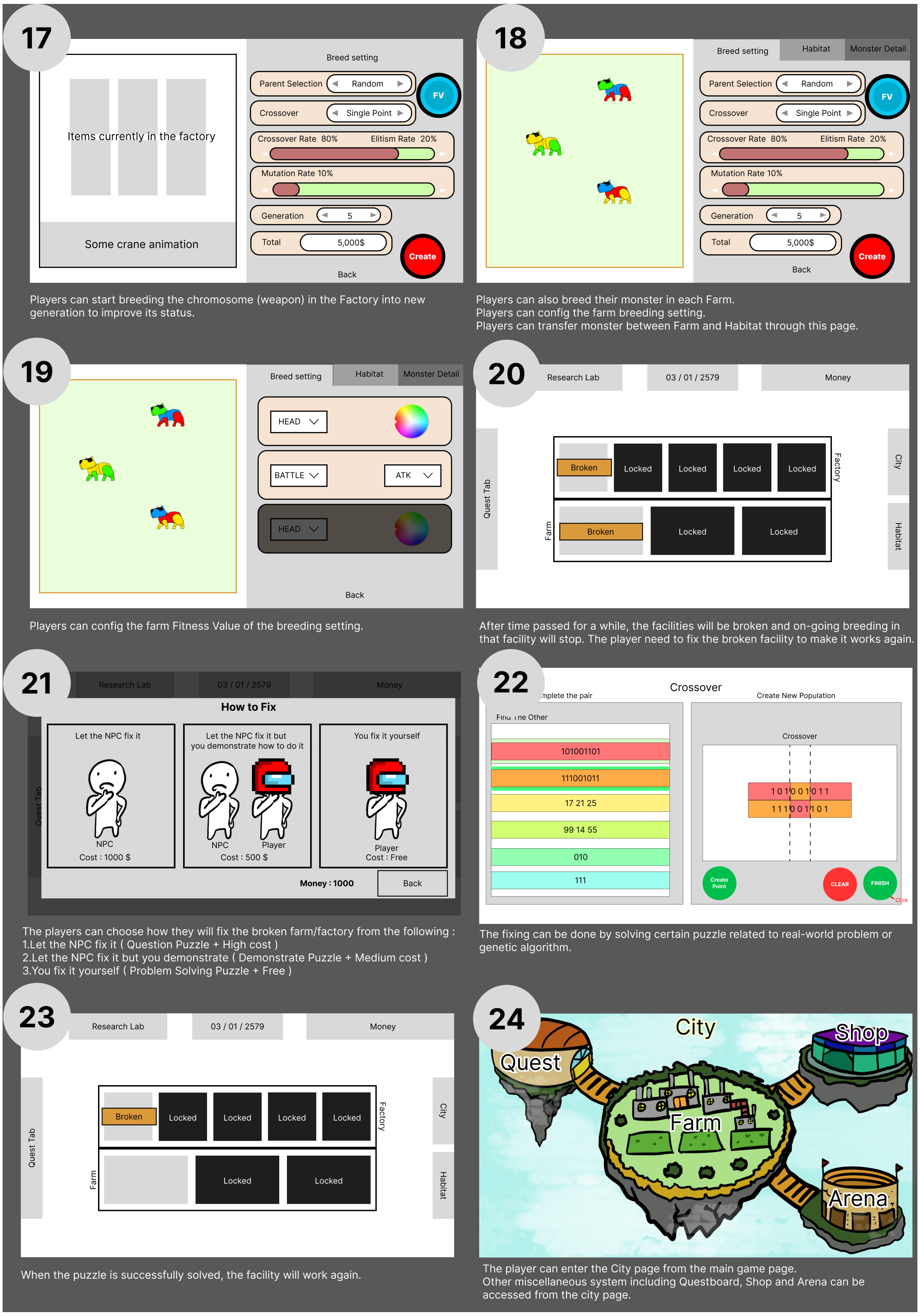


Figure A.3: Storyboard part 3 out of 4

Figure A.4: Storyboard part 4 out of 4

APPENDIX B
LEARNING MATERIAL

Learning Material in Research Lab

This appendix shows the learning material design which will be adapted and used in the Research Lab System. Some of the work might need more revision later since there is a change in the scope during the operation. The learning material will be grouped into a topic as we described in the system details under the game overview section.

- Basic Biology

  1. On the Origin of Species

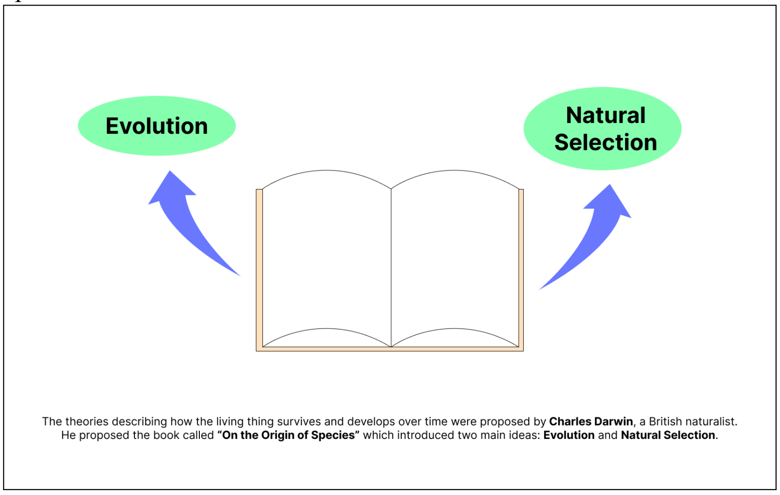


Figure B.1: Learning material 1 out of 6 of the topic: On the Origin of Species

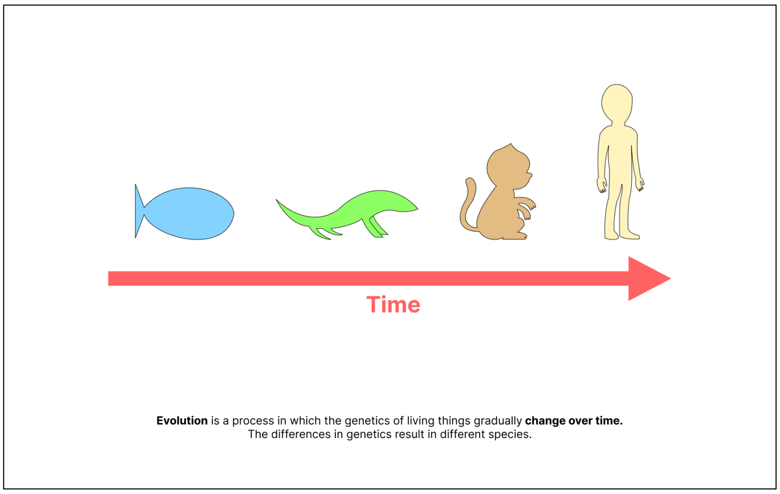


Figure B.2: Learning material 2 out of 6 of the topic: On the Origin of Species

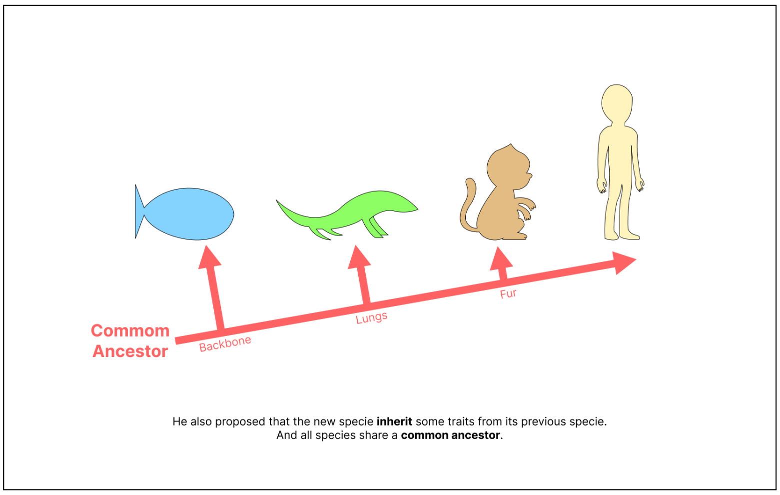


Figure B.3: Learning material 3 out of 6 of the topic: On the Origin of Species

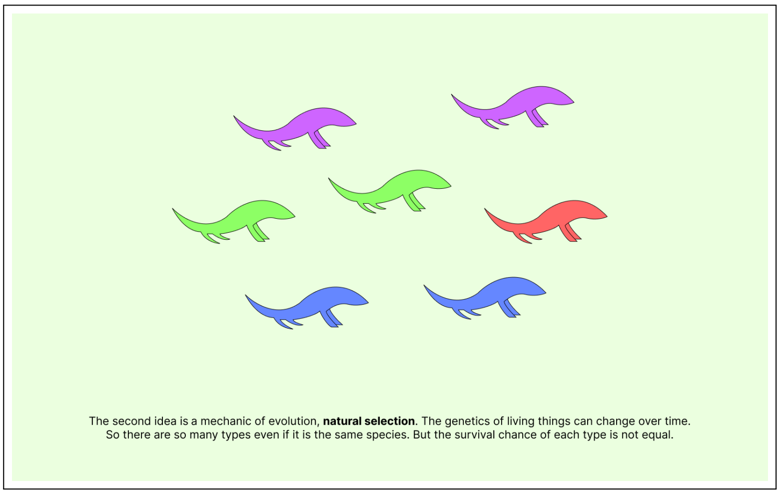


Figure B.4: Learning material 4 out of 6 of the topic: On the Origin of Species

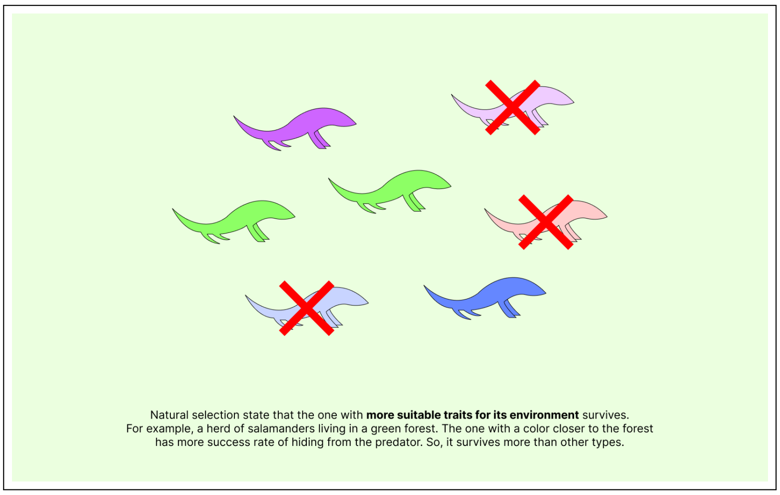


Figure B.5: Learning material 5 out of 6 of the topic: On the Origin of Species

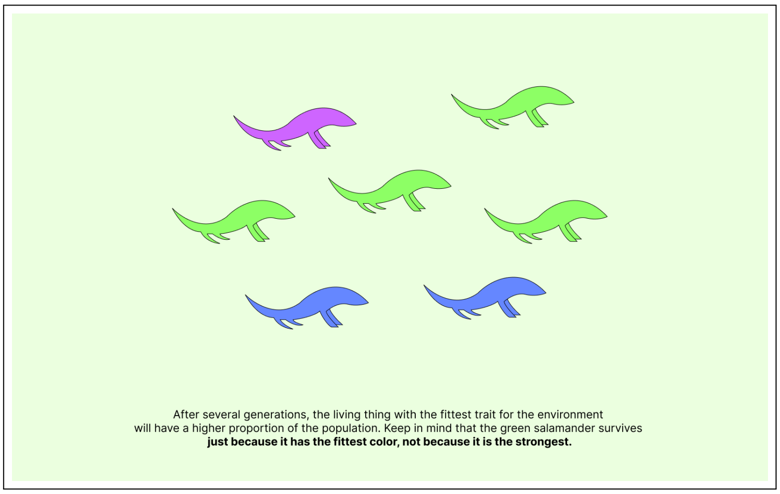


Figure B.6: Learning material 6 out of 6 of the topic: On the Origin of Species

2. The Genetic

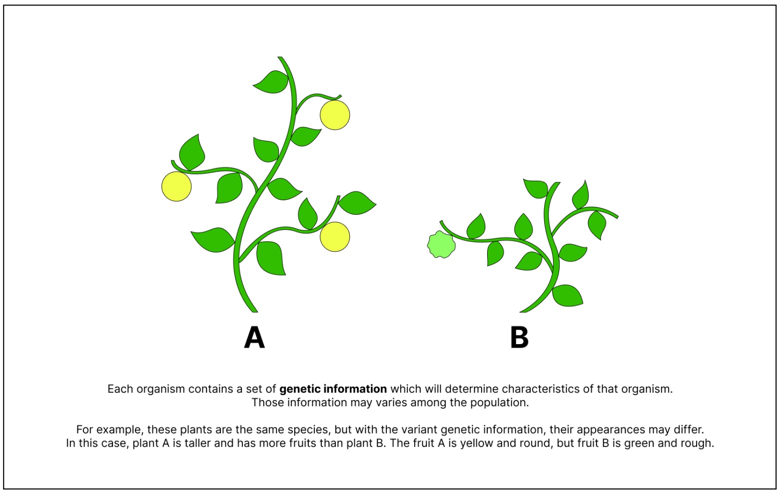


Figure B.7: Learning material 1 out of 4 of the topic: The Genetic

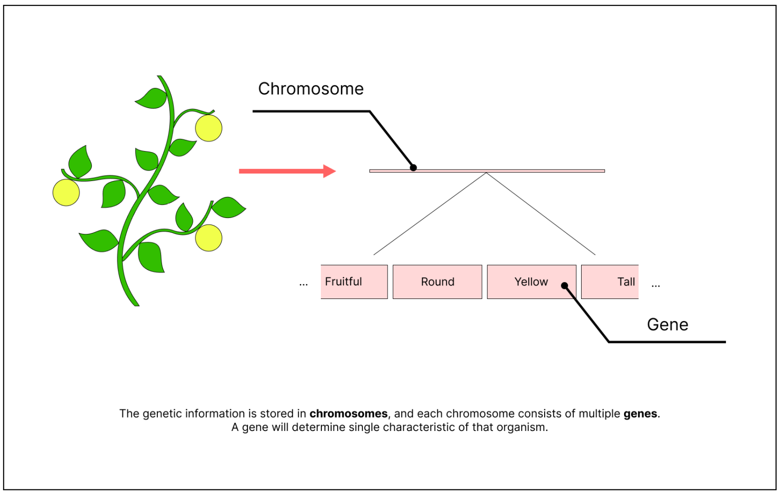


Figure B.8: Learning material 2 out of 4 of the topic: The Genetic

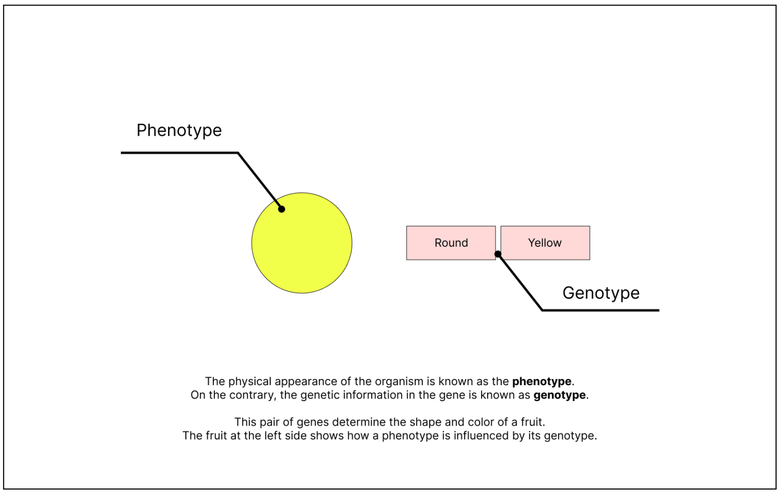


Figure B.9: Learning material 3 out of 4 of the topic: The Genetic

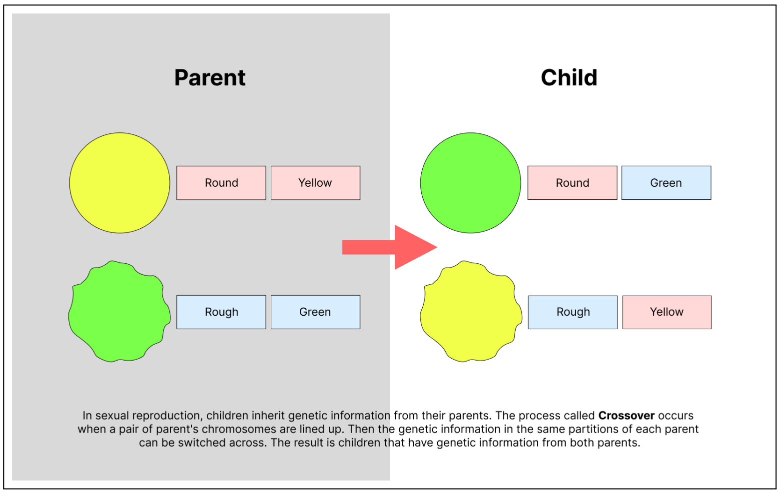


Figure B.10: Learning material 4 out of 4 of the topic: The Genetic

- Genetic Algorithm

1. The Flow of Genetic Algorithm

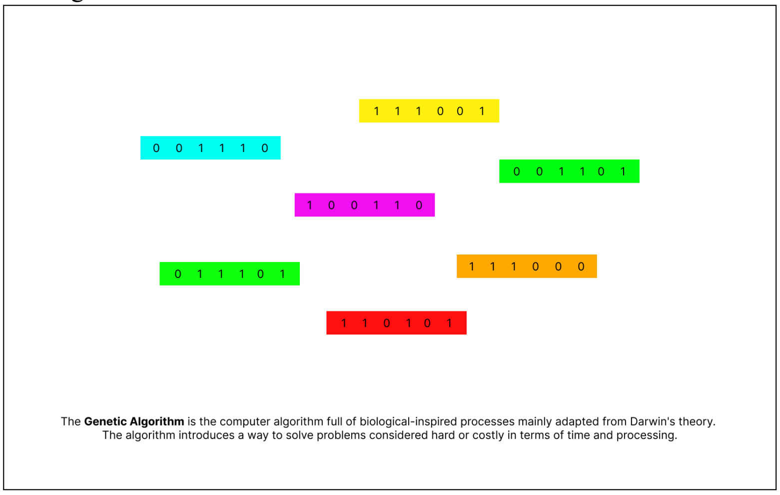


Figure B.11: Learning material 1 out of 9 of the topic: The Flow of Genetic Algorithm

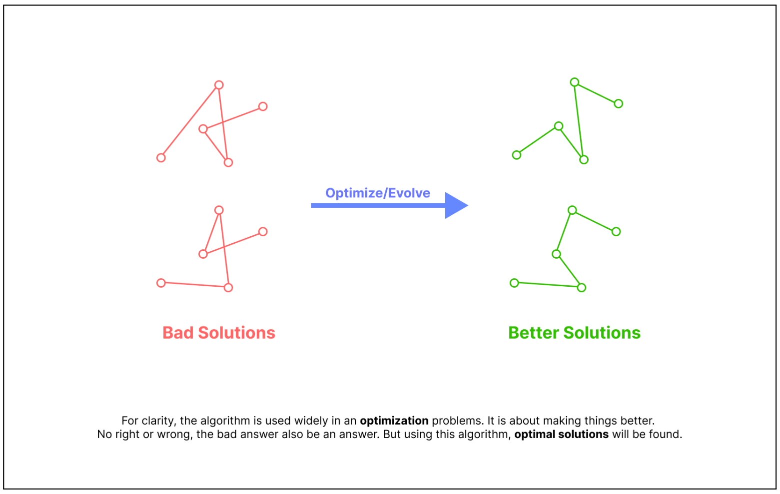


Figure B.12: Learning material 2 out of 9 of the topic: The Flow of Genetic Algorithm

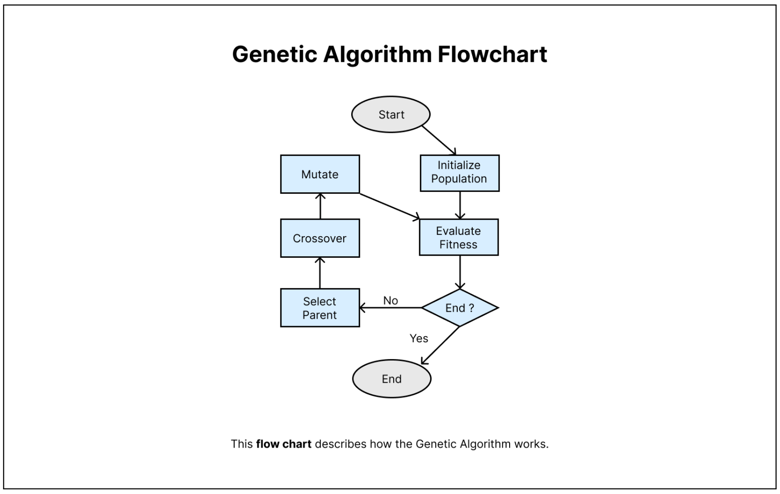


Figure B.13: Learning material 3 out of 9 of the topic: The Flow of Genetic Algorithm

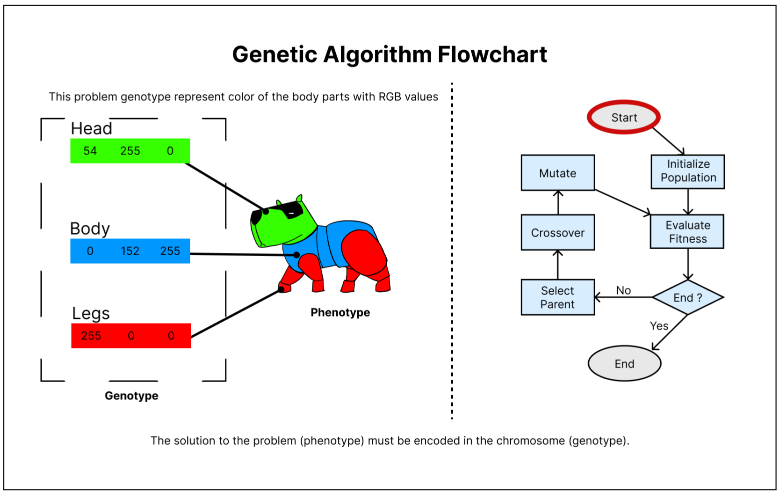


Figure B.14: Learning material 4 out of 9 of the topic: The Flow of Genetic Algorithm

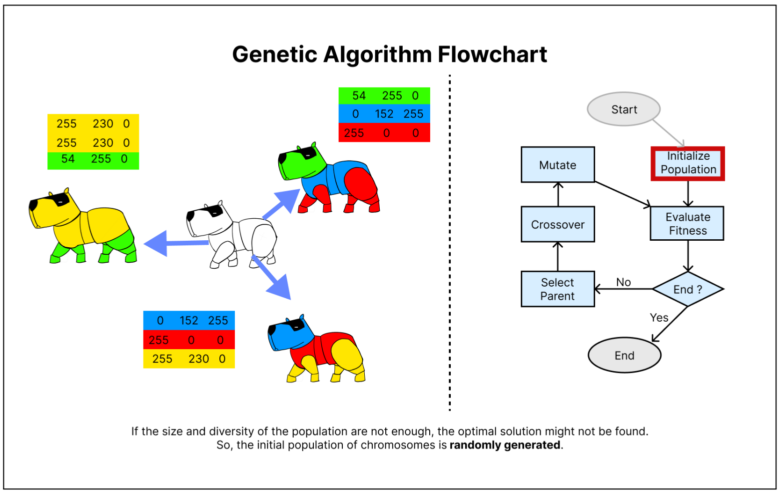


Figure B.15: Learning material 5 out of 9 of the topic: The Flow of Genetic Algorithm

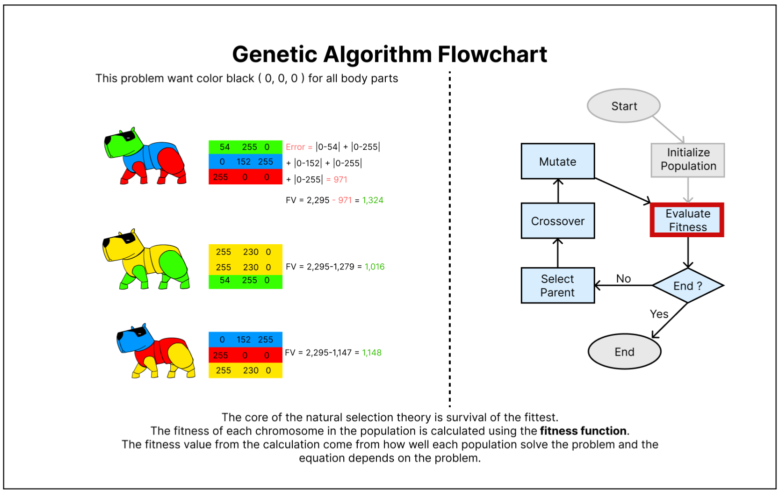


Figure B.16: Learning material 6 out of 9 of the topic: The Flow of Genetic Algorithm

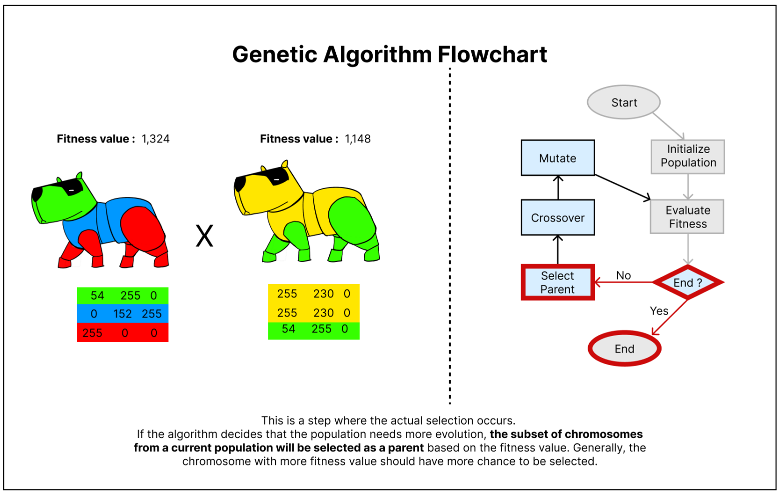


Figure B.17: Learning material 7 out of 9 of the topic: The Flow of Genetic Algorithm

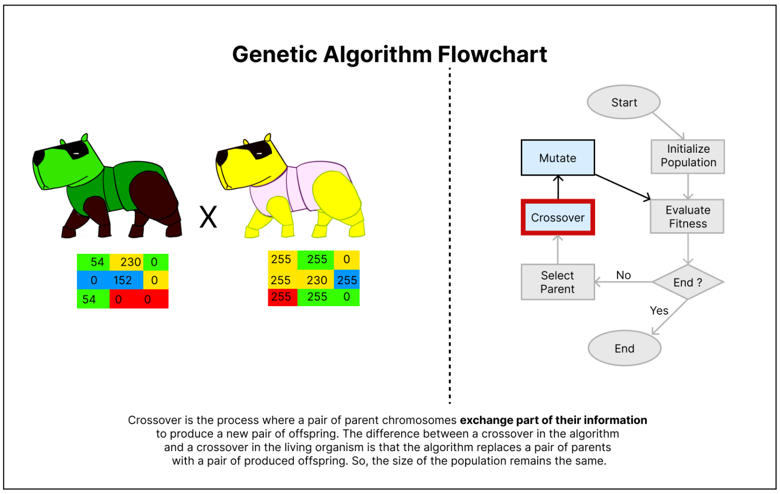


Figure B.18: Learning material 8 out of 9 of the topic: The Flow of Genetic Algorithm

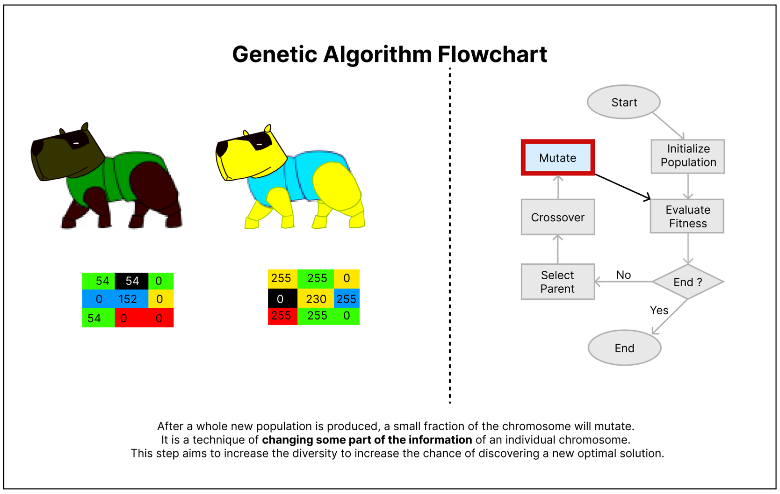


Figure B.19: Learning material 9 out of 9 of the topic: The Flow of Genetic Algorithm

2. Parent Selection

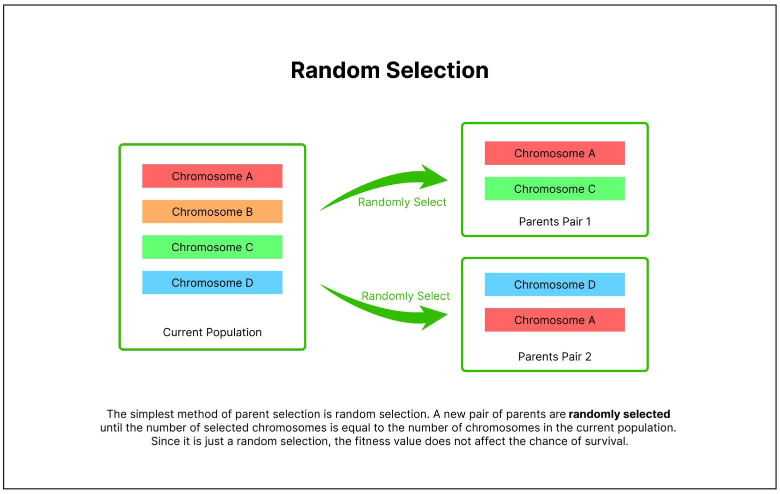


Figure B.20: Learning material of the topic: Random Selection

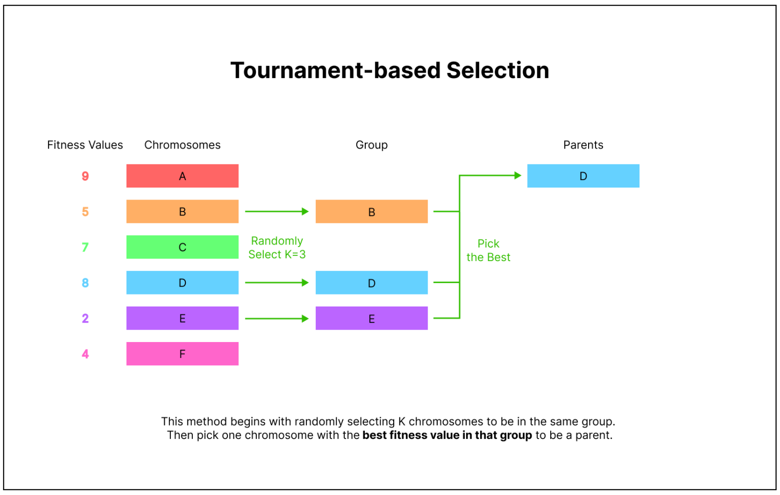


Figure B.21: Learning material 1 out of 2 of the topic: Tournament-based Selection

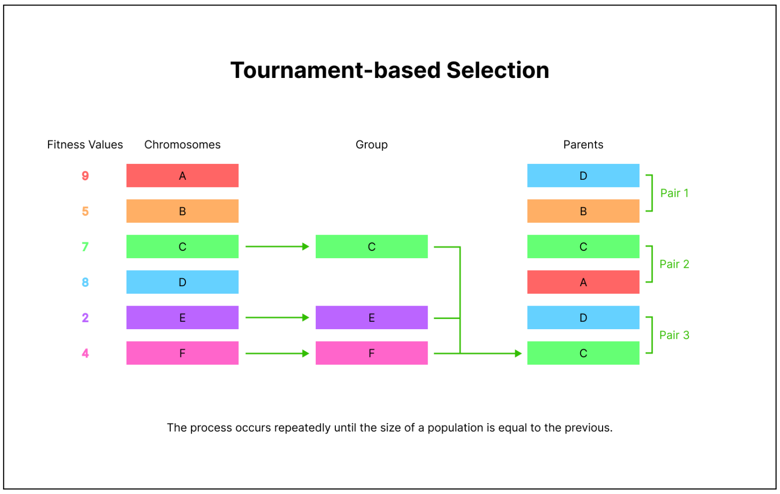


Figure B.22: Learning material 2 out of 2 of the topic: Tournament-based Selection

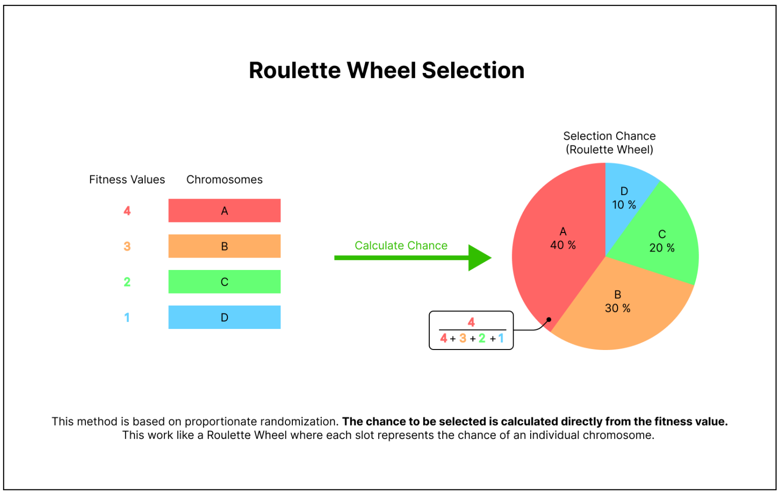


Figure B.23: Learning material 1 out of 2 of the topic: Roulette Wheel Selection

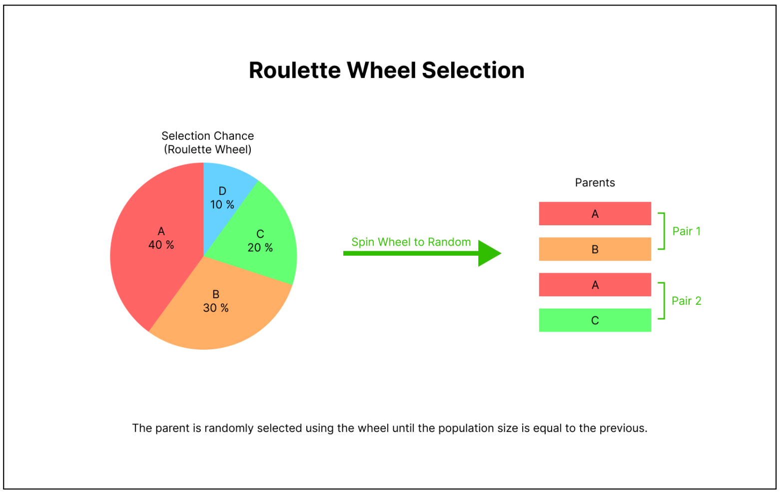


Figure B.24: Learning material 2 out of 2 of the topic: Roulette Wheel Selection

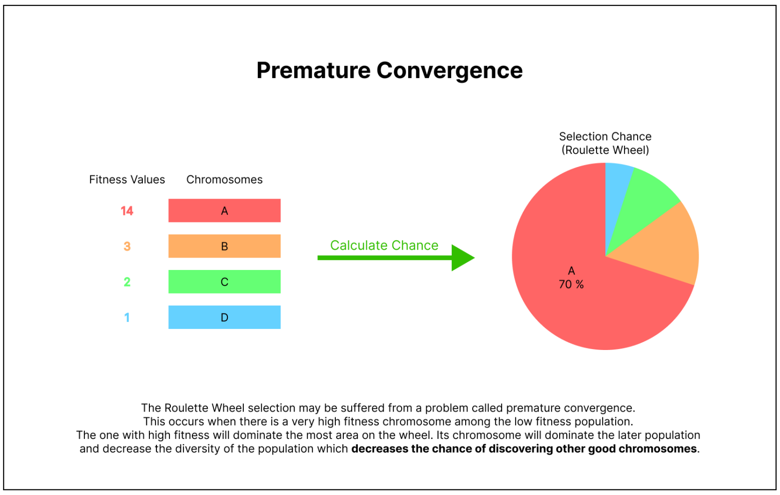


Figure B.25: Learning material of the topic: Premature Convergence

Figure B.26: Learning material of the topic: Rank-based Selection

3. Crossover

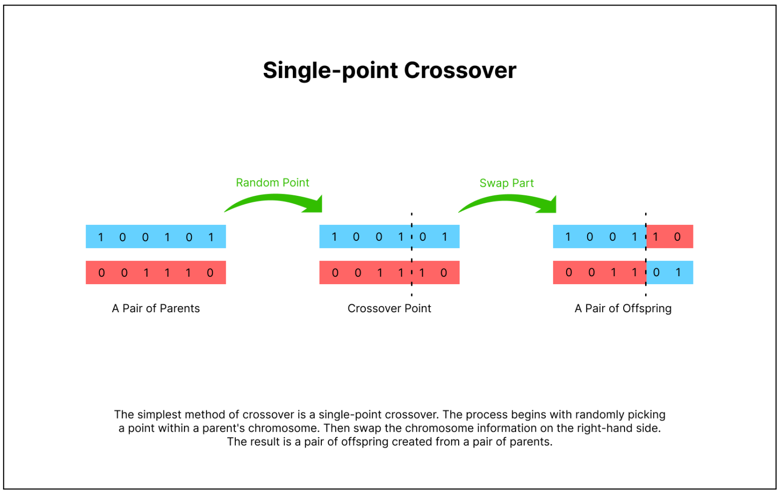


Figure B.27: Learning material of the topic: Single-point Crossover

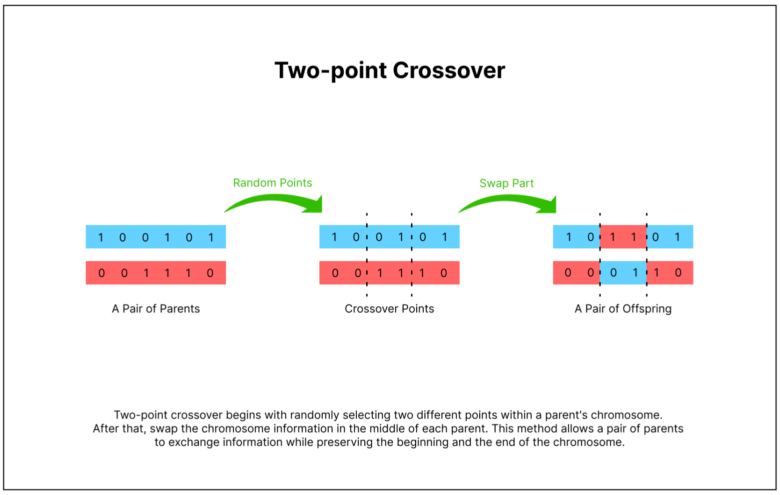


Figure B.28: Learning material of the topic: Two-point Crossover

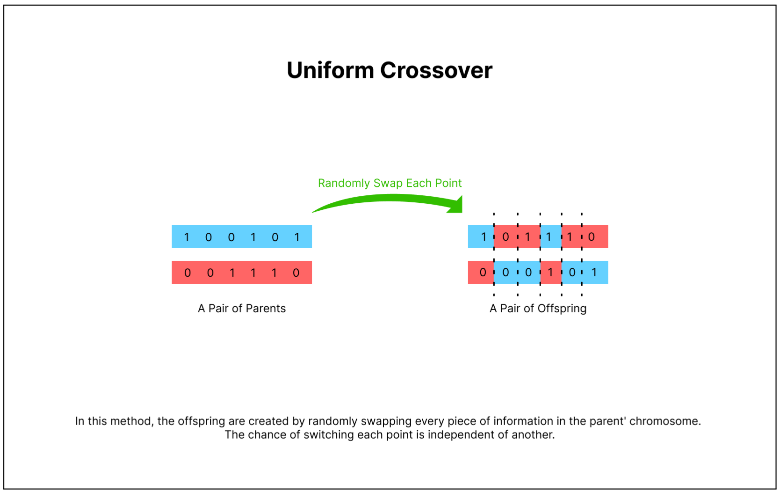


Figure B.29: Learning material of the topic: Uniform Crossover

4. Improvement Techniques

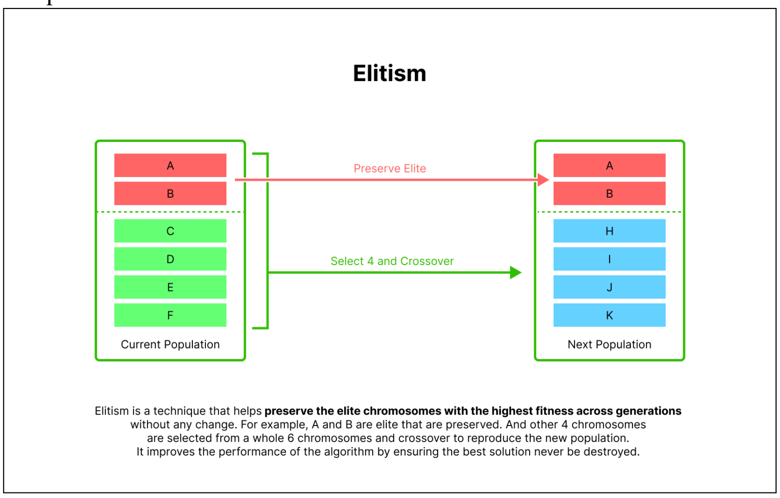


Figure B.30: Learning material of the topic: Elitism

APPENDIX C
PUZZLES

Puzzle Storyboard

This appendix shows the storyboard of the puzzle in a manner of the sequence of user interface (UI). The demonstration puzzle and problem-solving puzzle are included. Each type of puzzle can also be divided into the parent selection and the crossover. Note that the puzzle related to the real-world problem of encoding and decoding is in the progress of revision due to the change in the project's scope.

- Selection Demonstration

1. Roulette Wheel Selection

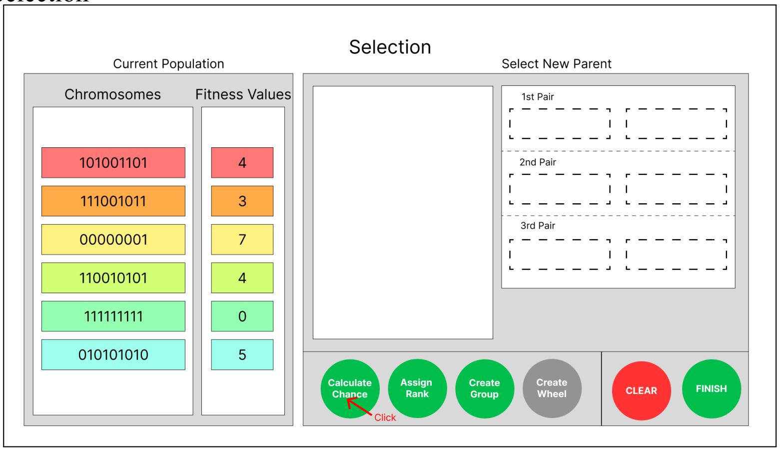


Figure C.1: Demonstration Puzzle of Roulette Wheel Selection 1 out of 6

The players click calculate chance to calculate percent of probability proportional to the current population fitness.

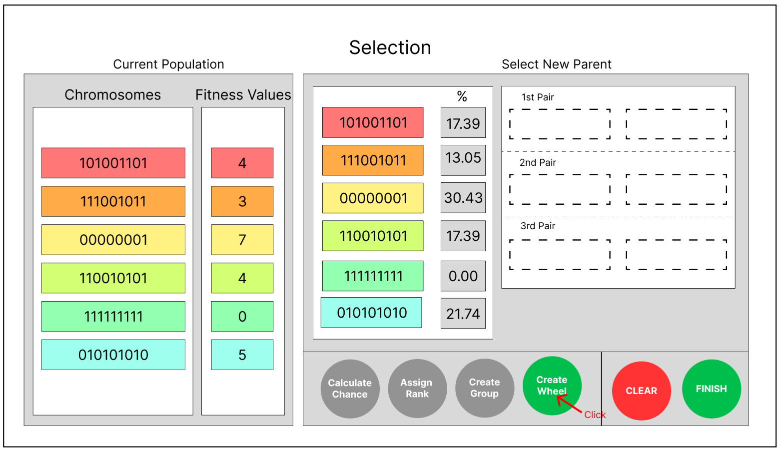


Figure C.2: Demonstration Puzzle of Roulette Wheel Selection 2 out of 6

The players click Create Wheel to make a pie graph.

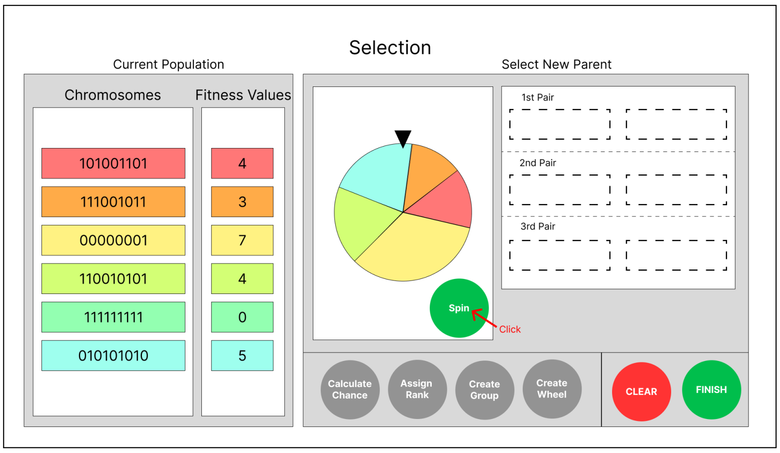


Figure C.3: Demonstration Puzzle of Roulette Wheel Selection 3 out of 6

The players click Spin to spin the wheel.

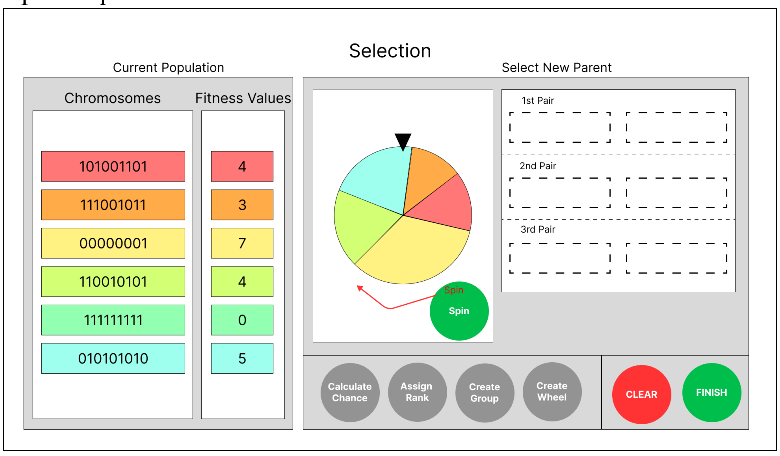


Figure C.4: Demonstration Puzzle of Roulette Wheel Selection 4 out of 6

The wheel spins until it stops.

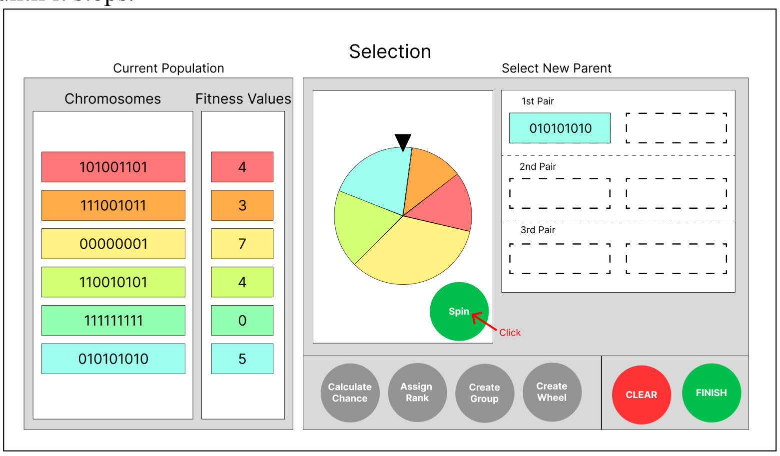


Figure C.5: Demonstration Puzzle of Roulette Wheel Selection 5 out of 6

When the wheel stops spinning, the pointed chromosome in the wheel will be selected. The players click spin to spin the wheel again.

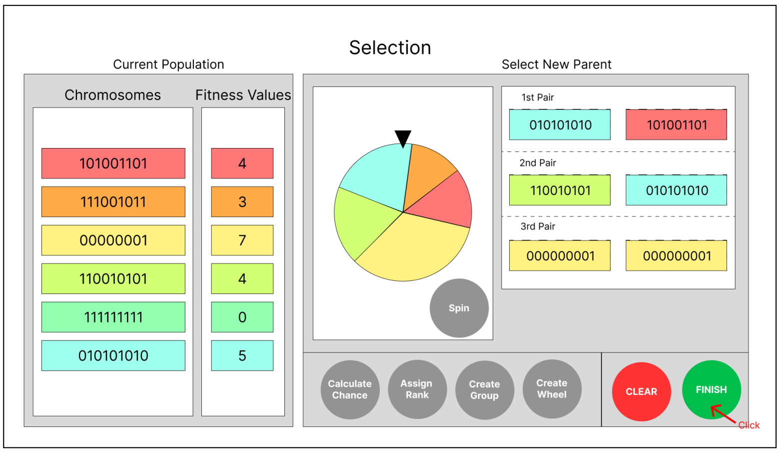


Figure C.6: Demonstration Puzzle of Roulette Wheel Selection 6 out of 6

The players click spin until all pairs of new parents are complete then click finish to send the answer.

2. Tournament-based Selection

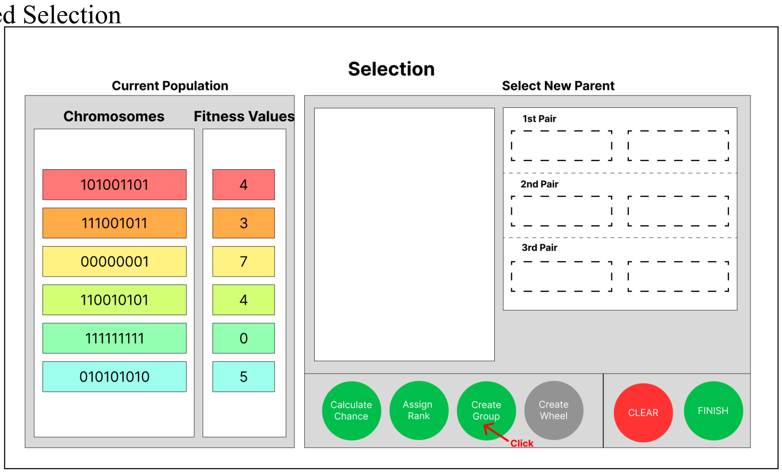


Figure C.7: Demonstration Puzzle of Tournament-based Selection 1 out of 4

The players click Create Group to randomly pick 3 chromosomes out of the current population.

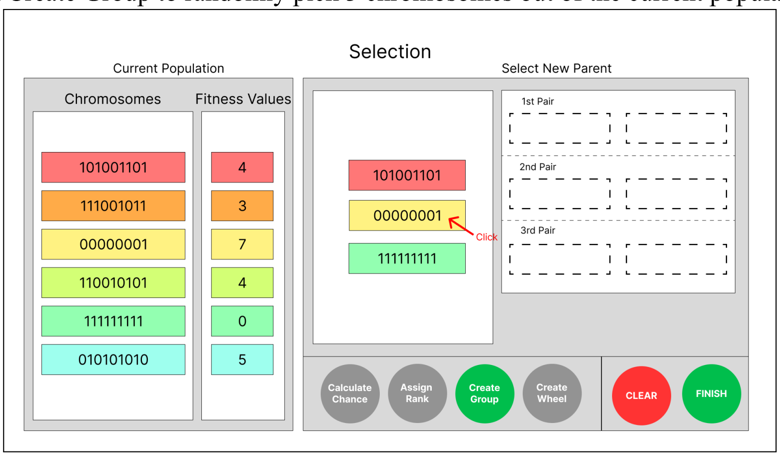


Figure C.8: Demonstration Puzzle of Tournament-based Selection 2 out of 4

The players click the chromosome with the highest fitness values as a selected chromosome.

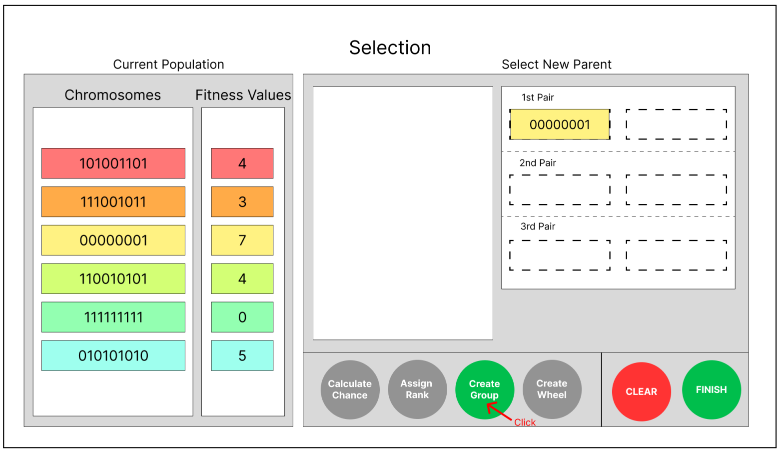


Figure C.9: Demonstration Puzzle of Tournament-based Selection 3 out of 4

The players click Create Group again to randomly pick another 3 chromosomes out of the current population.

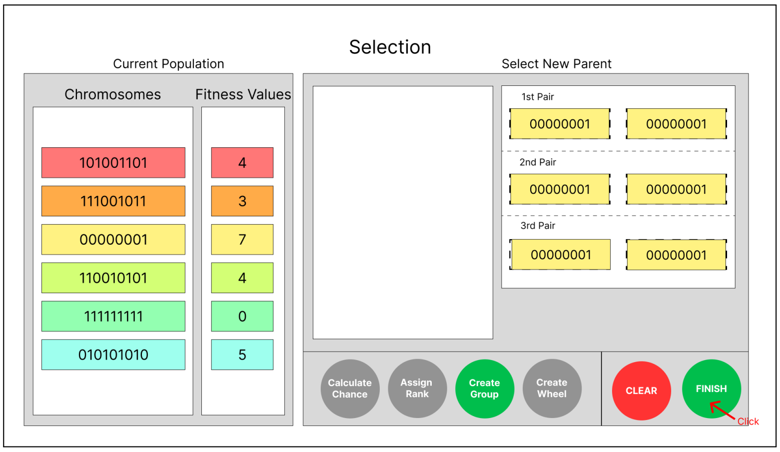


Figure C.10: Demonstration Puzzle of Tournament-based Selection 4 out of 4

The players repeat the action until all pairs of new parents are complete then click finish to send the answer.

3. Rank-based Selection

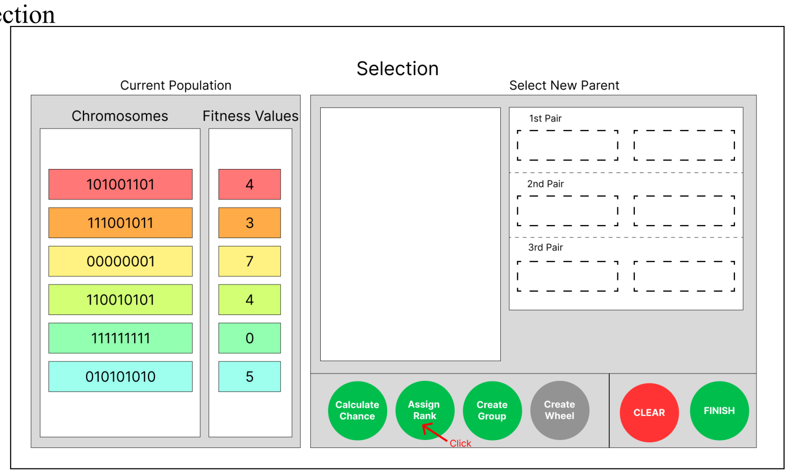


Figure C.11: Demonstration Puzzle of Rank-based Selection 1 out of 11

The players click assign rank to assign the rank to the current population.

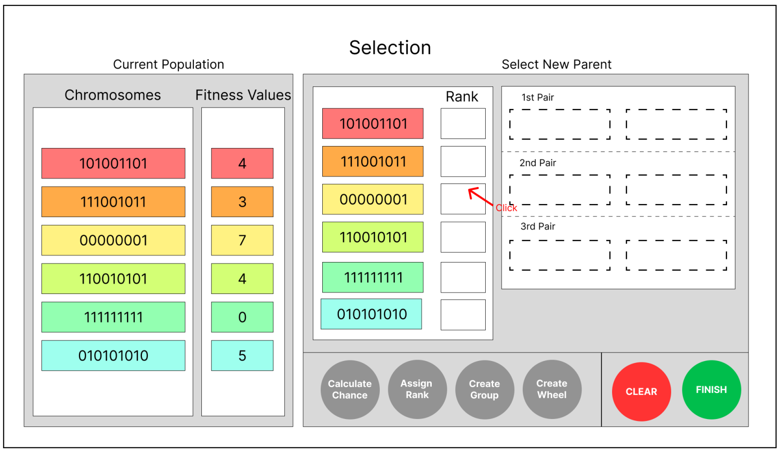


Figure C.12: Demonstration Puzzle of Rank-based Selection 2 out of 11

The players click on the rank block to assign the number of rank.

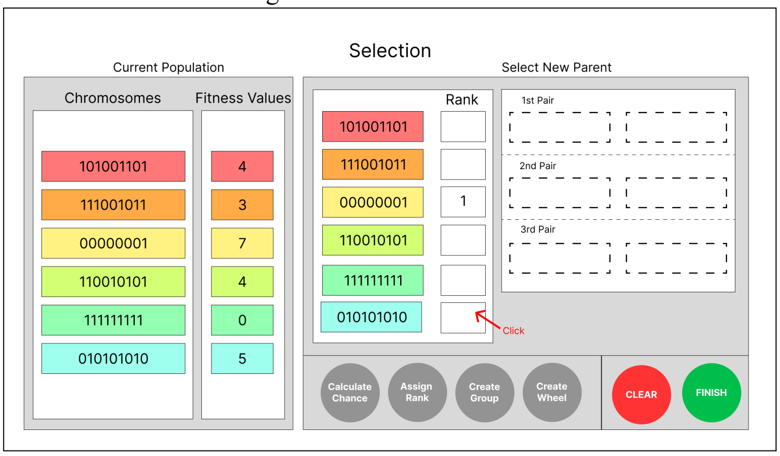


Figure C.13: Demonstration Puzzle of Rank-based Selection 3 out of 11

The players click on the other rank block to assign the next number of rank.

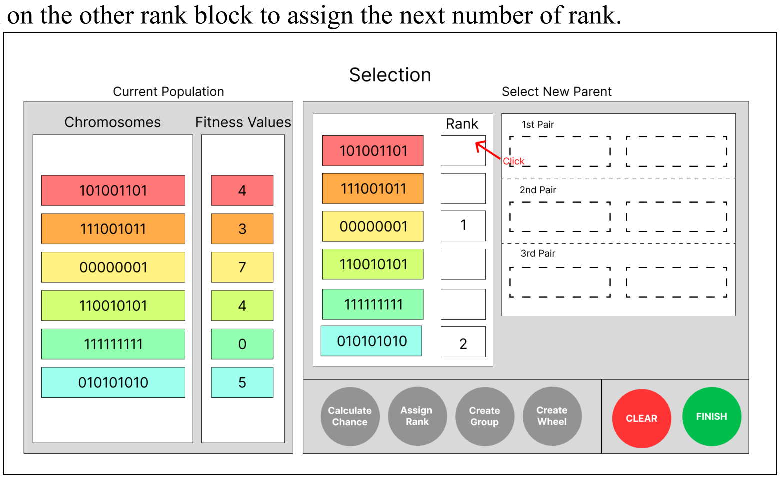


Figure C.14: Demonstration Puzzle of Rank-based Selection 4 out of 11

The players click on the other rank block to assign the next number of rank until every block is filled.

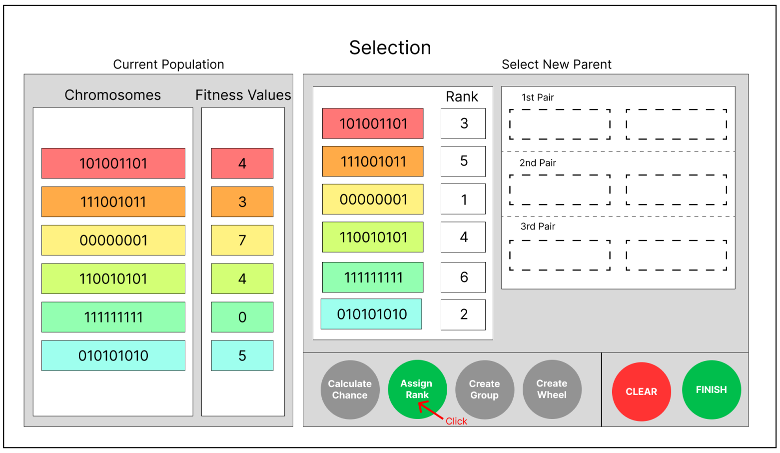


Figure C.15: Demonstration Puzzle of Rank-based Selection 5 out of 11

The players click assign rank again after filled all the rank block to confirm the filled rank.

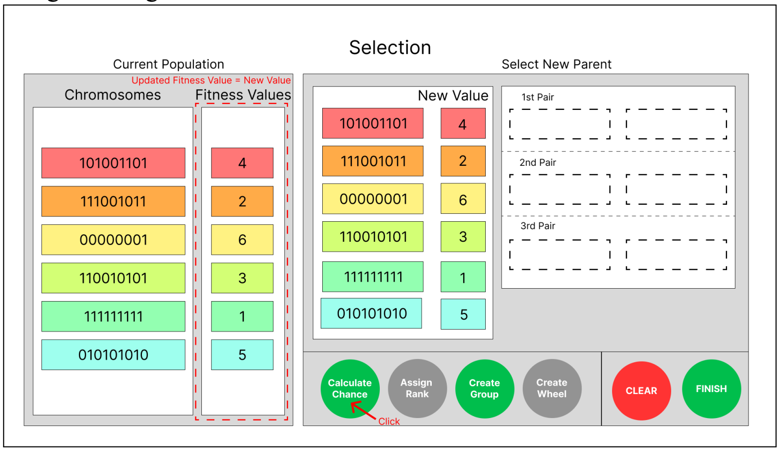


Figure C.16: Demonstration Puzzle of Rank-based Selection 6 out of 11

The puzzle will create new fitness values and revise the original value to the new one. The players click Calculate Chance to calculate percent of probability proportional to the current population fitness.

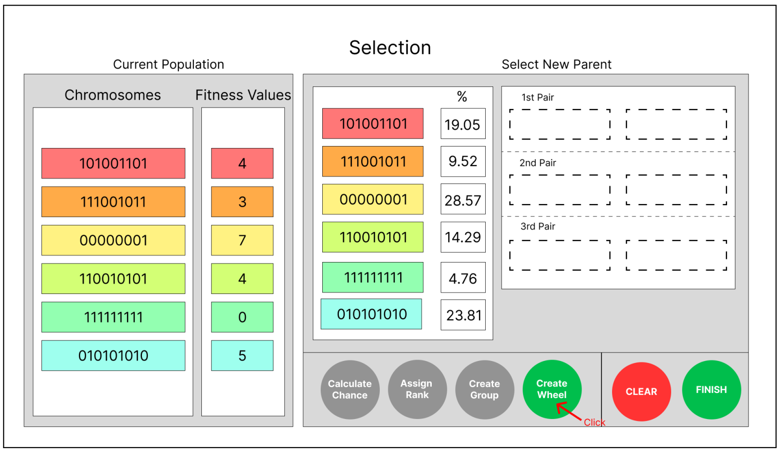


Figure C.17: Demonstration Puzzle of Rank-based Selection 7 out of 11

The players click Create Wheel to make a pie graph.

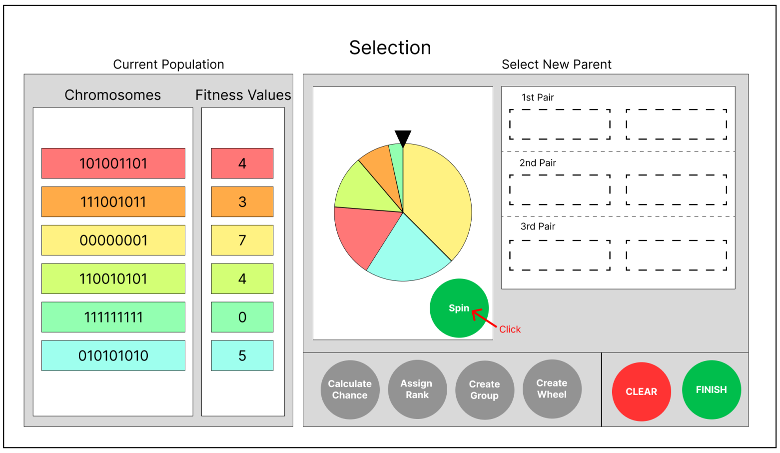


Figure C.18: Demonstration Puzzle of Rank-based Selection 8 out of 11

The players click Spin to spin the wheel.

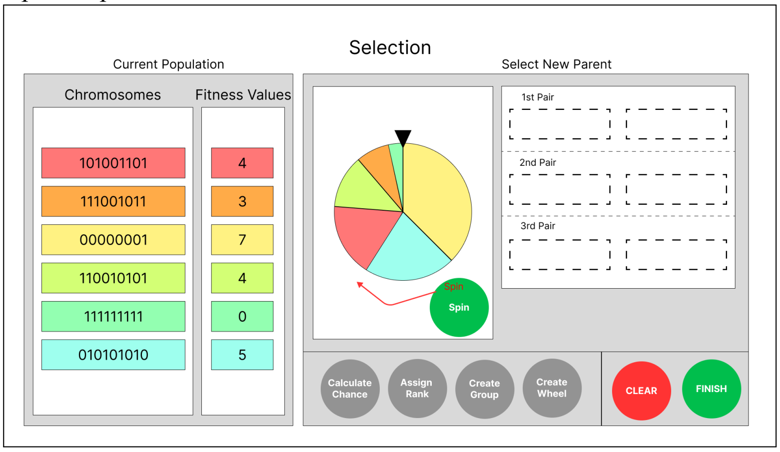


Figure C.19: Demonstration Puzzle of Rank-based Selection 9 out of 11

The wheel spins until it stops.

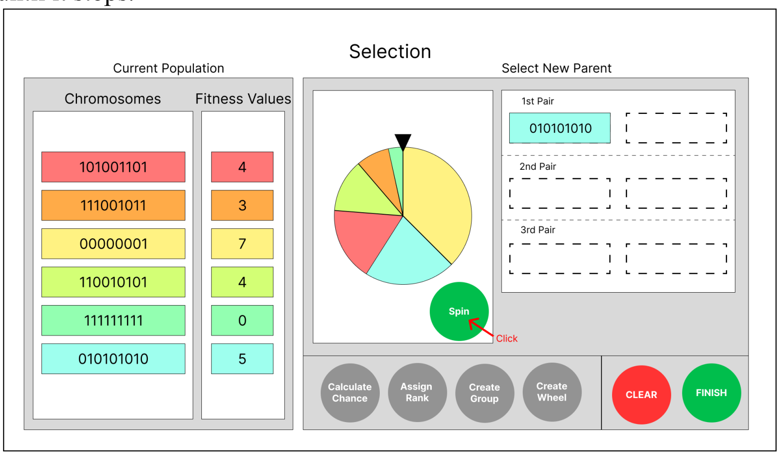


Figure C.20: Demonstration Puzzle of Rank-based Selection 10 out of 11

When the wheel stops spinning, the pointed chromosome in the wheel will be selected. The players click spin to spin the wheel again.

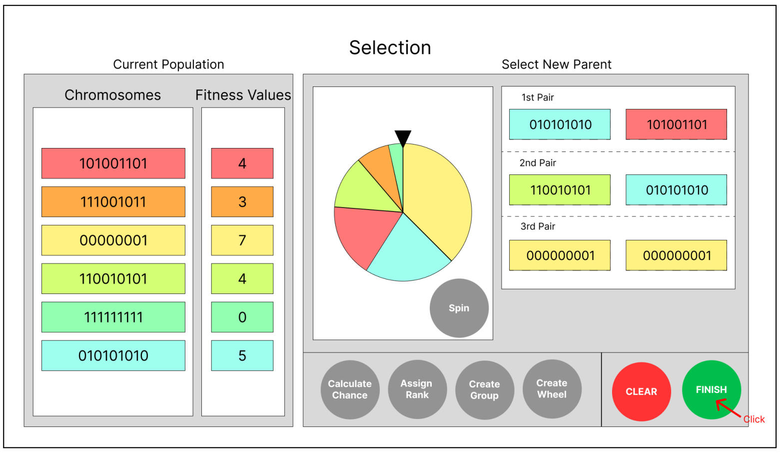


Figure C.21: Demonstration Puzzle of Rank-based Selection 11 out of 11

The players click spin until all pairs of new parents are complete then click finish to send the answer.

- Crossover Demonstration

1. Single-point Crossover

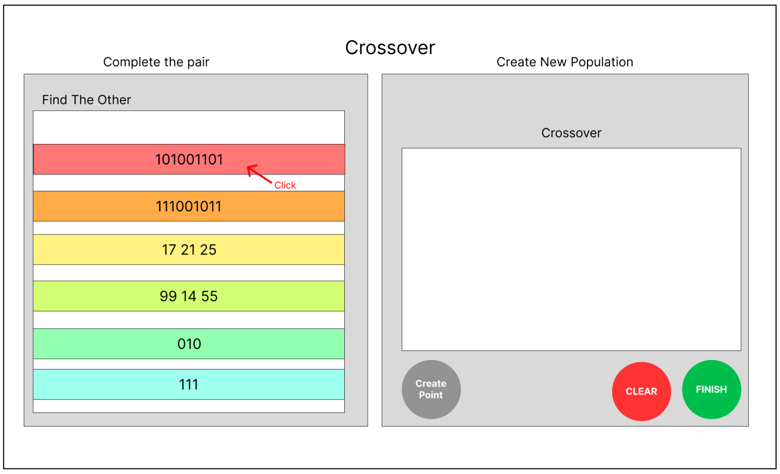


Figure C.22: Demonstration Puzzle of Single-point Crossover 1 out of 5

The players click the chromosome they want to use.

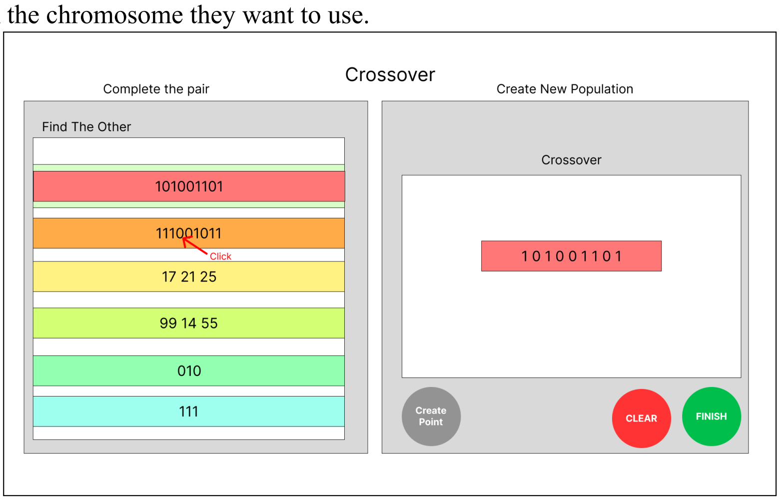


Figure C.23: Demonstration Puzzle of Single-point Crossover 2 out of 5

The players click the other chromosome with the same type of the first picked chromosome to complete the pair.

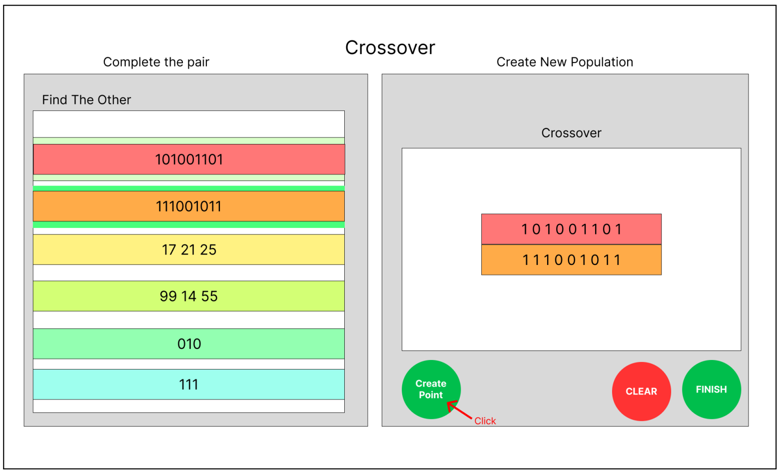


Figure C.24: Demonstration Puzzle of Single-point Crossover 3 out of 5

The players click create point to create a random crossover point on the pair.

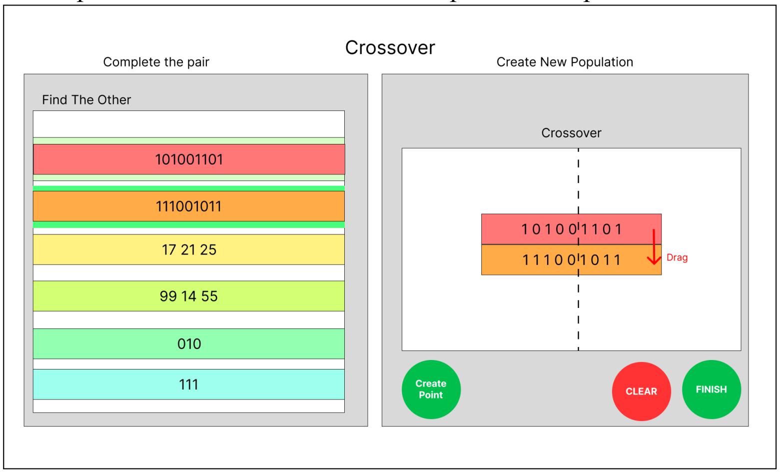


Figure C.25: Demonstration Puzzle of Single-point Crossover 4 out of 5

The players drag the right side of the top chromosome to swap with the bottom chromosome.

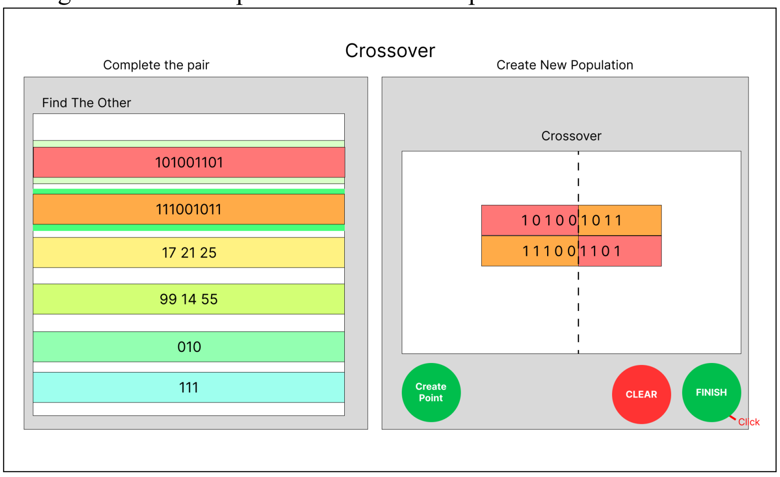


Figure C.26: Demonstration Puzzle of Single-point Crossover 5 out of 5

The players click finish to send the answer.

2. Two-point Crossover

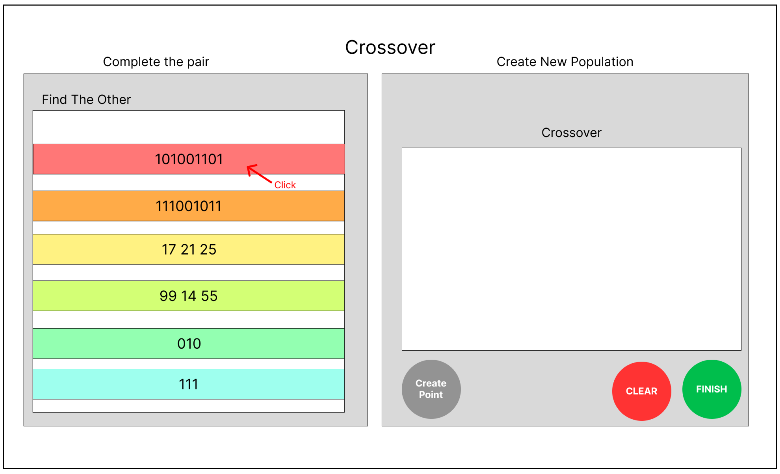


Figure C.27: Demonstration Puzzle of Two-point Crossover 1 out of 6

The players click the chromosome they want to use.

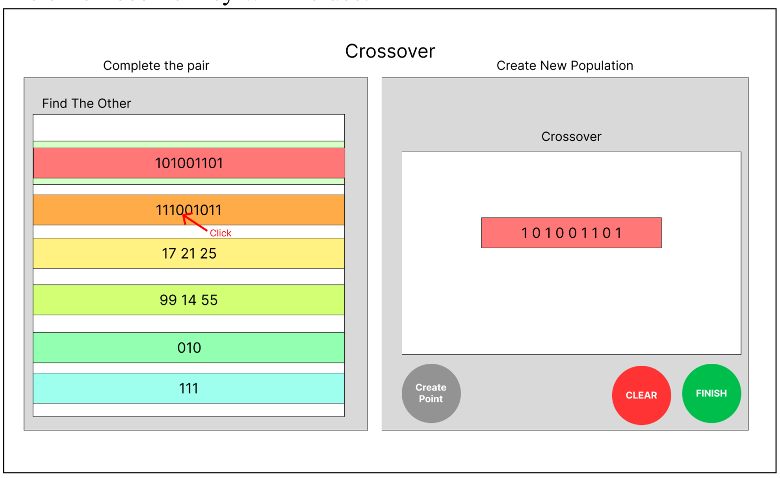


Figure C.28: Demonstration Puzzle of Two-point Crossover 2 out of 6

The players click the other chromosome with the same type of the first picked chromosome to complete the pair.

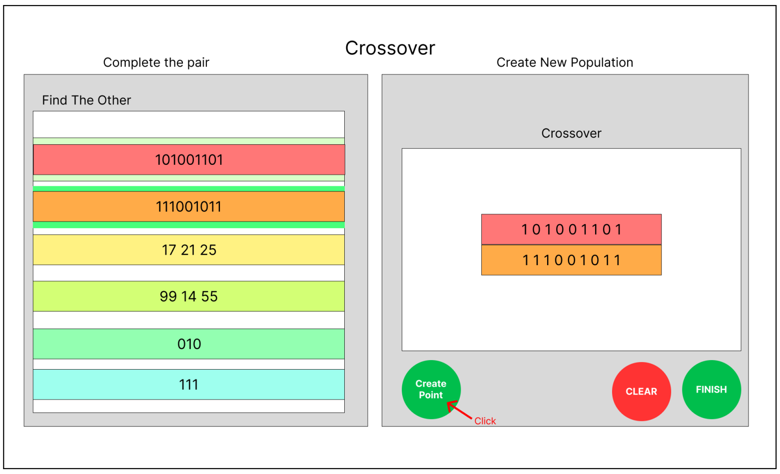


Figure C.29: Demonstration Puzzle of Two-point Crossover 3 out of 6

The players click create point to create a random crossover point on the pair.

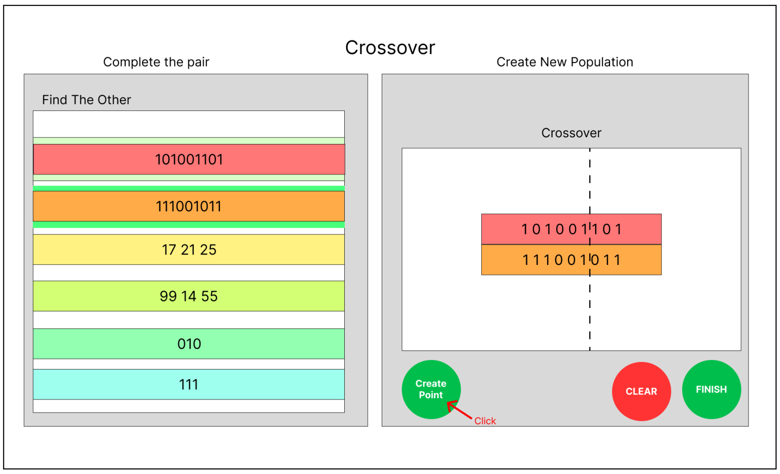


Figure C.30: Demonstration Puzzle of Two-point Crossover 4 out of 6

The players click create point again to create a second random crossover point on the pair.

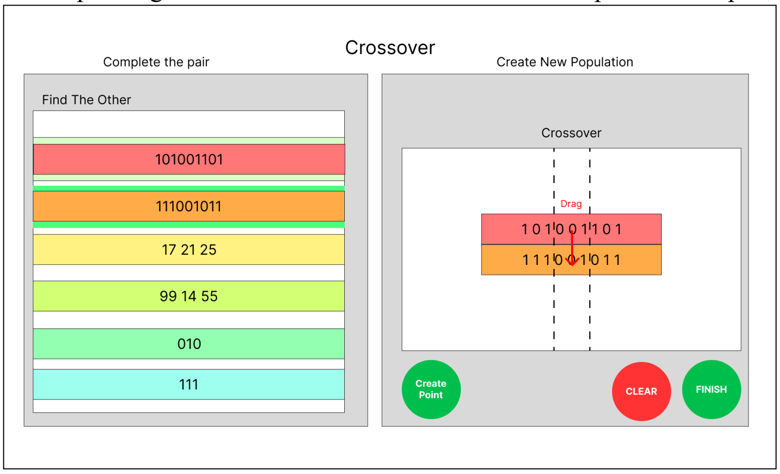


Figure C.31: Demonstration Puzzle of Two-point Crossover 5 out of 6

The players drag the middle part of the top chromosome to swap with the bottom chromosome.

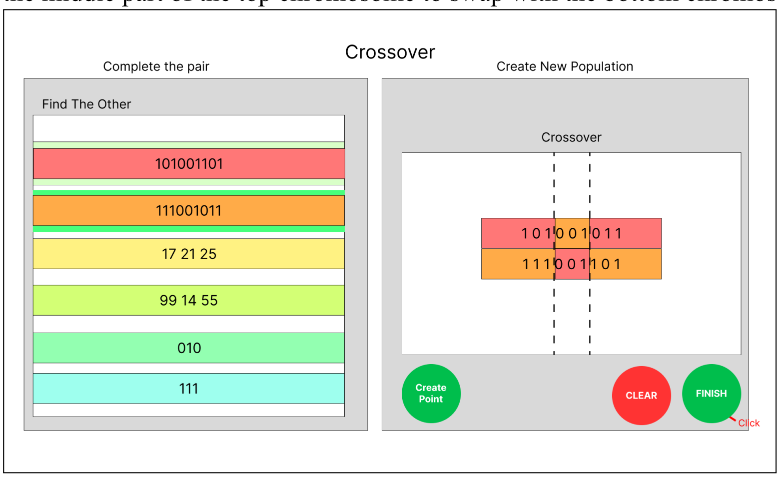


Figure C.32: Demonstration Puzzle of Two-point Crossover 6 out of 6

The players click finish to send the answer.

- Selection Problem-solving

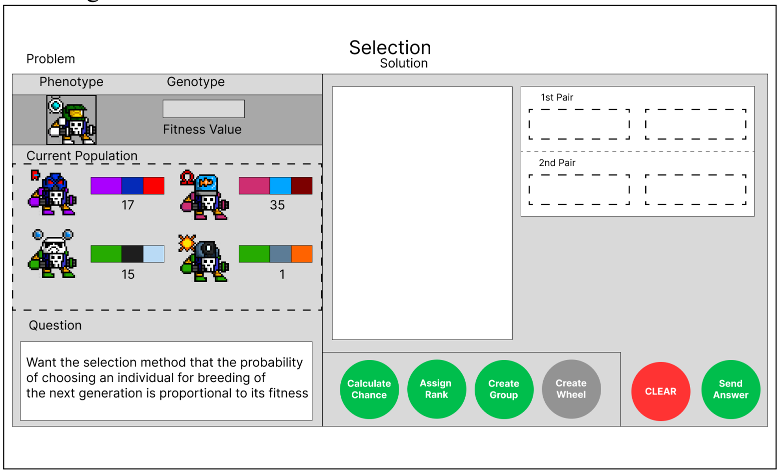


Figure C.33: Problem-solving Puzzle of Roulette Wheel Selection

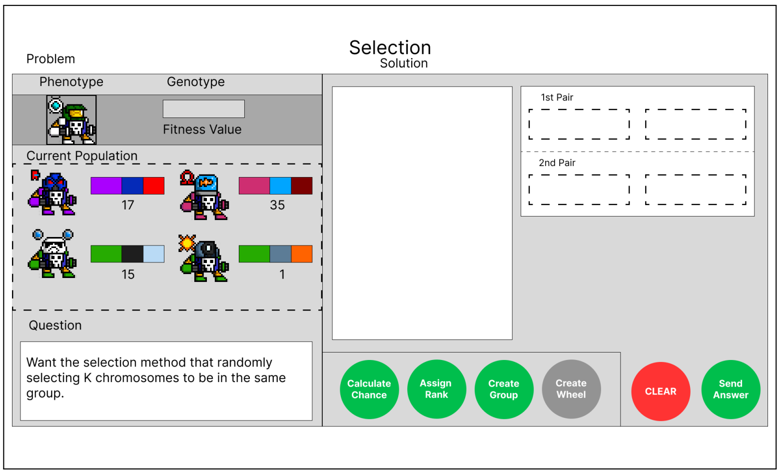


Figure C.34: Problem-solving Puzzle of Tournament-based Selection

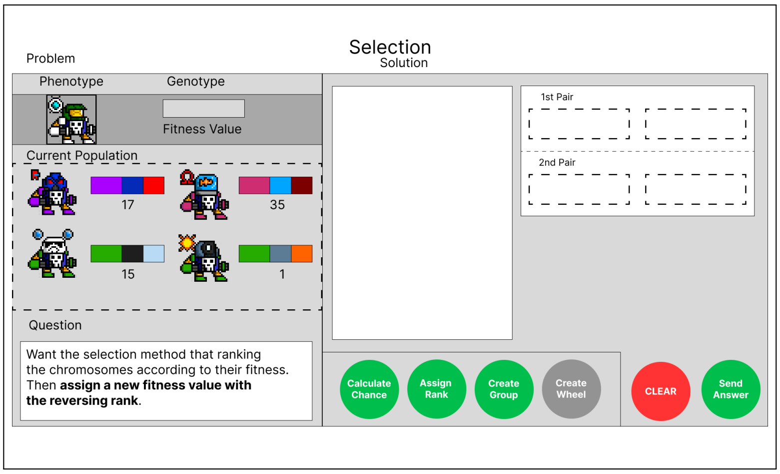


Figure C.35: Problem-solving Puzzle of Rank-based Selection

- Crossover Problem-solving

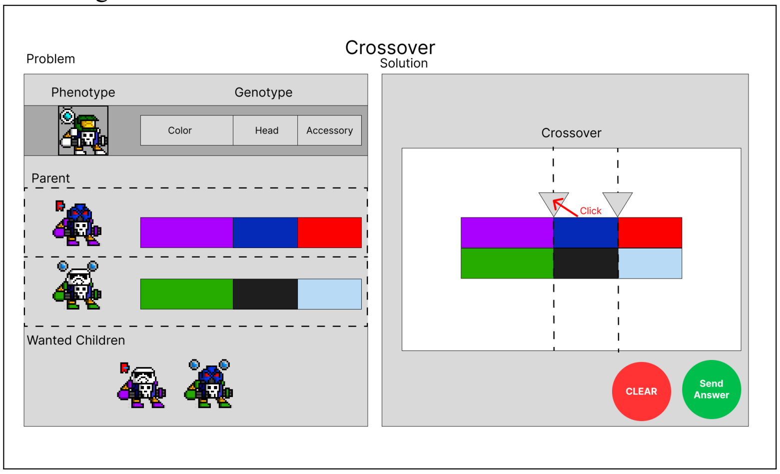


Figure C.36: Problem-solving Puzzle of Crossover 1 out of 4

The players click create point to create a crossover point on the pair.

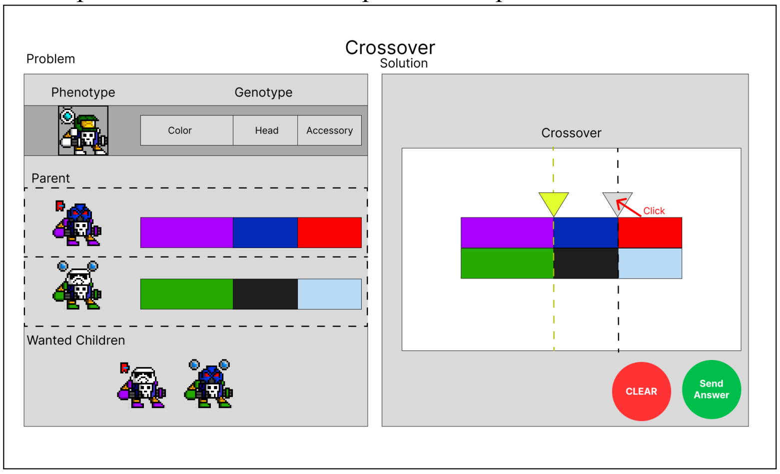


Figure C.37: Problem-solving Puzzle of Crossover 2 out of 4

The players click create point to create a second crossover point on the pair.

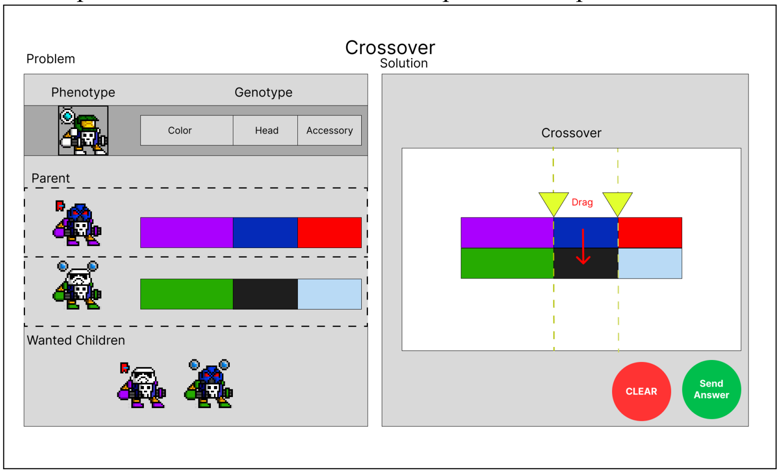


Figure C.38: Problem-solving Puzzle of Crossover 3 out of 4

The players drag the middle part of the top chromosome to swap with the bottom chromosome.

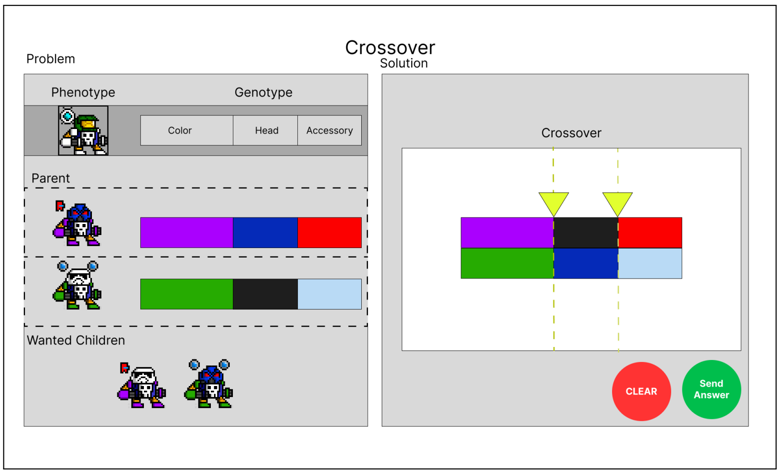


Figure C.39: Problem-solving Puzzle of Crossover 4 out of 4

The players click Send Answer to send the answer.

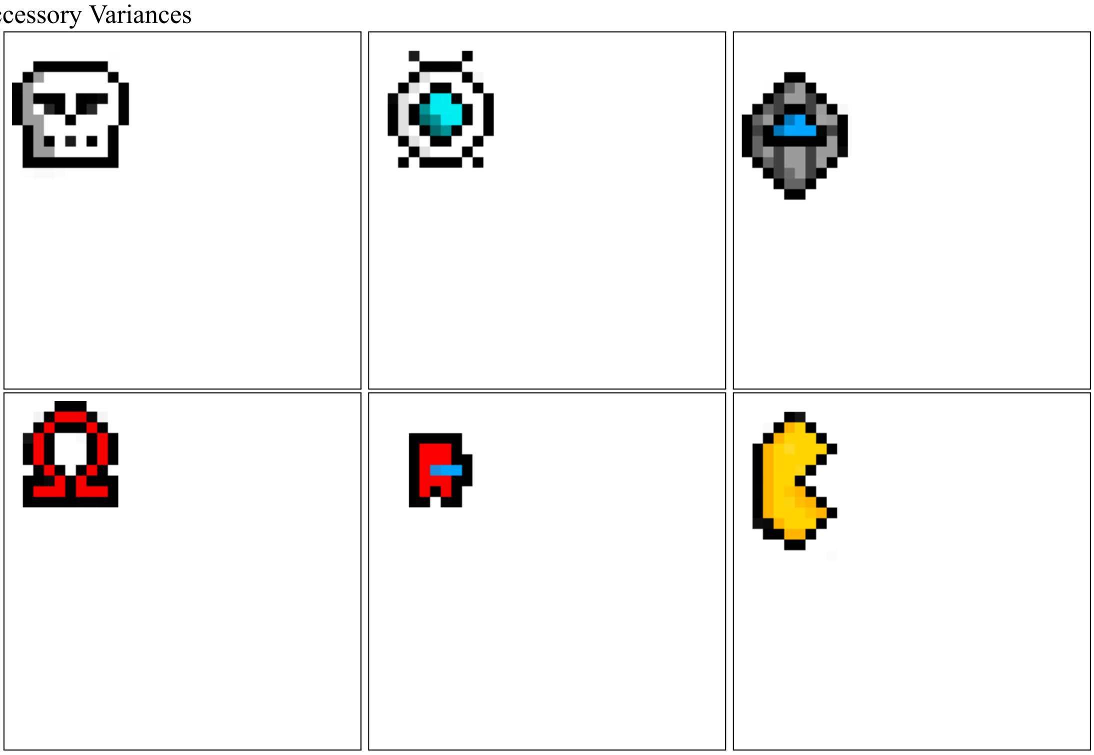
APPENDIX D
CHARACTER SPRITES

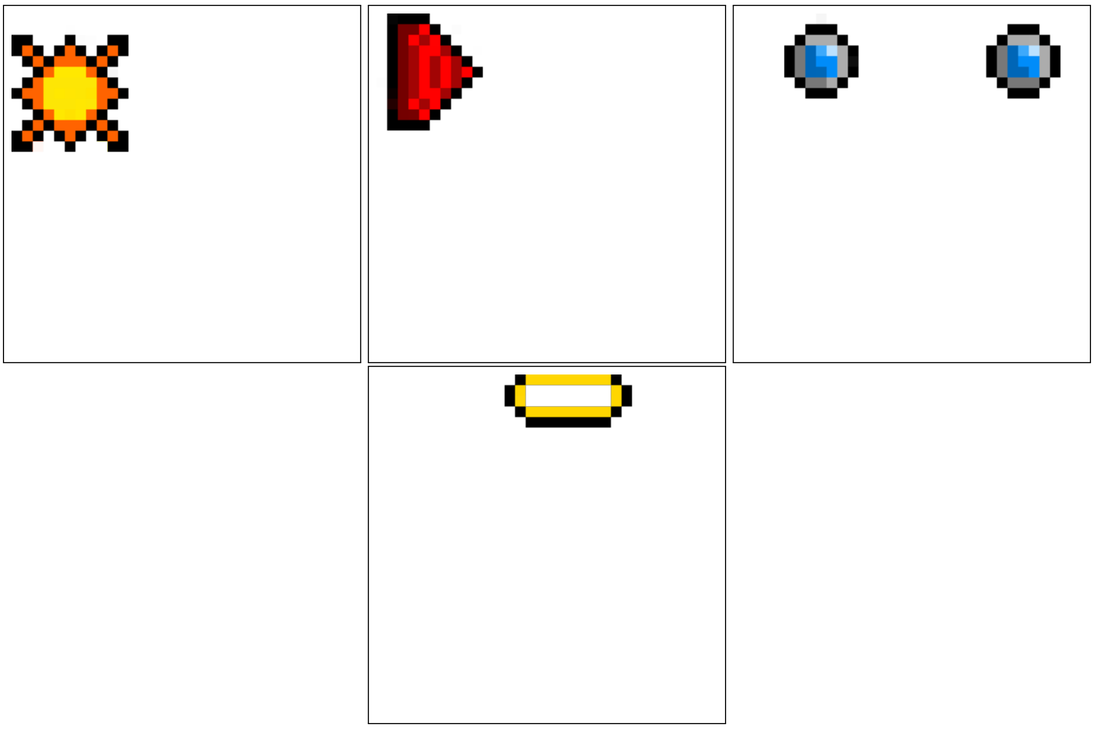
Character Sprites

Sprites of every monster in this game consist of three parts, head, body, and accessory. Currently, there are twenty head variances and ten accessory variances. The body of a monster will stay the same, but its arms and legs can change colors.

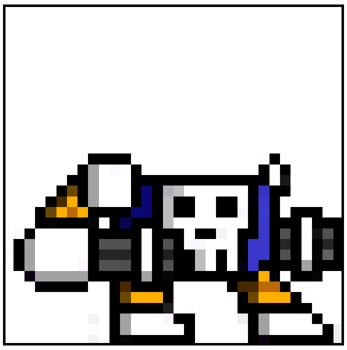
1. Head Variances

2. Accessory Variances





3. Body



4. Assembling Examples