



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

.....
(Asst.Prof. Suthathip Maneewongvatana, Ph.D.)

Committee Member

.....
(Jaturon Harnsomburana, Ph.D.)

Committee Member

Copyright reserved

Project Title	Project G - Educational Game for Learning Genetic Algorithm
Credits	3
Member(s)	Mr. Thanapat Mahiskhamin Mr. Pojnarin Nanta Mr. Sirapop Apananda
Project Advisor	Assoc.Prof. Natasha Dejdumrong, D.Tech.Sci.
Co-advisor	Taweechai Nuntawisuttiwong, Ph.D.
Program	Bachelor of Engineering
Field of Study	Computer Engineering
Department	Computer Engineering
Faculty	Engineering
Academic Year	2022

Abstract

Nowadays, education is not explicitly conducted in a classroom anymore. Everybody can gain new knowledge about anything at any time they want. Various media are also adjusted to fit diverse target groups; one of the most popular formats is the educational game.

Project G - Educational Game for Learning Genetic Algorithm was built with the intention of providing players knowledge about the Genetic Algorithm and the Knapsack Problem through the game, as it can be both entertaining and academic at the same time. It can also evaluate players with various methods. Its training methods through puzzles will give players more understanding of those topics.

In this game, players will roleplay as mech company owners, using the Genetic Algorithm in the mech and weapon production. Players have to learn and use the knowledge to manage the facilities. Above that, there are more activities for players, such as fixing facilities, producing mechs according to customers' requests, and using mechs to battle and earn rewards.

Keywords: Computer Game / Genetic Algorithm / Educational / Breeding Simulation / Resource Management / Puzzle

หัวข้อปริญญาบัณฑิต	Project G - Educational Game for Learning Genetic Algorithm
หน่วยกิต	3
ผู้เขียน	นายธนาวัชร์ มหาดิษฐ์มิน
	นายพจนรินทร์ นันตะ
	นายสิรภพ อปานนท์
อาจารย์ที่ปรึกษา	รศ.ดร. ณัฐชา เดชาธรรม ดร. ทวีชัย นันทวิสุทธิจิวงศ์
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมคอมพิวเตอร์
คณะ	วิศวกรรมศาสตร์
ปีการศึกษา	2565

บทคัดย่อ

ในปัจจุบัน การเรียนรู้นั้นไม่ถูกจำกัดอยู่ภายในห้องเรียนอีกต่อไป ทุกคนสามารถหาความรู้ได้อย่างรวดเร็วและง่ายดายกว่าเดิม สื้อต่าง ๆ สามารถเข้าถึงได้เพียงเชื่อมต่ออินเทอร์เน็ต ยิ่งไปกว่านั้น สื่อการเรียนรู้เองก็ได้มีการพัฒนาให้มีความหลากหลายให้เหมาะสมกับกลุ่มเป้าหมายต่าง ๆ และหนึ่งในรูปแบบที่ถูกจับตามองก็คือเกมเพื่อการศึกษานั่นเอง

Project G - Educational Game for Learning Genetic Algorithm นั้นถูกจัดทำขึ้นโดยมีจุดประสงค์เพื่อมอบความรู้เชิงวิชาการเกี่ยวกับอัลกอริธึมเชิงพันธุกรรมและปัญหาถุกระสอบให้กับผู้ที่สนใจผ่านรูปแบบเกมซึ่งเป็นสื่อที่สามารถตอบทั้งความรู้และความสนุกให้กับผู้เล่น พัฒนาทักษะการวัดผลการเรียนรู้อีกด้วย แต่เดียวกันนี้ยังคงมีวิธีการสอนและบททวนผ่านการแก้ปริศนาซึ่งสามารถช่วยให้ผู้เล่นเข้าใจหลักการของอัลกอริธึมได้ดียิ่งขึ้น

ตัวเกมจะให้ผู้เล่นได้สวมบทบาทเป็นเจ้าของโรงงานพัฒนาหุ่นยนต์ซึ่งมีวิธีการพัฒนาหุ่นยนต์และอาชีวศึกษาอัลกอริธึมเชิงพันธุกรรม โดยผู้เล่นจะต้องเรียนรู้หลักการทำงานของมันและใช้ความรู้ที่มีประกอบการบริหารโรงงานได้ ซึ่งมีกิจกรรมต่าง ๆ ให้ผู้เล่นได้ทำ เช่นการทำบัญชีงาน การผลิตหุ่นยนต์ให้ตรงตามความต้องการของลูกค้า หรือการนำหุ่นยนต์ไปโรงงานไปต่อสู้เพื่อชิงเงินรางวัล

คำสำคัญ: เกมคอมพิวเตอร์ / อัลกอริธึมเชิงพันธุกรรม / การศึกษา / การจำลองการสมมต์ / การจัดการทรัพยากร / ปริศนา

ACKNOWLEDGMENTS

First, we would like to thank our advisors, Assoc.Prof.Dr. Natasha Dejbumrong and Dr. Taweechai Nuntawisuttiwong, for giving us various advice and sharing their knowledge about game development and other crucial things to make this project happen.

Next, we sincerely appreciate the help of every committee for commenting and setting our project on the right path. There were multiple issues that we overlooked in the project, and we would not notice them without the wit of the committees.

We are truly grateful for our 92 play-testers sacrificing their own time to test our game and submit numerous helpful feedback from the perspectives of the player that we developers might ignore to improve our game. In addition to those mentioned earlier, we are thankful to ChatGPT for providing us with many helpful and creative opinions, and supporting us in diverse fields throughout the project. Arknights is thanked for giving us ideas and inspiration about how the developers care about their game and players.

Finally, we thank the capybaras for being such cute creatures and providing us with the motivation to start and finish this project. This gratitude extends to the natural selection and evolution that make capybaras exist on this planet. We genuinely appreciate that.

CONTENTS

	PAGE
ABSTRACT	ii
THAI ABSTRACT	iii
ACKNOWLEDGMENTS	iv
CONTENTS	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER	
1. INTRODUCTION	1
1.1 Background	1
1.2 Objectives	1
1.3 Scope of Work	1
1.4 Project Schedule	3
1.4.1 Development Activities	3
1.4.2 Product Backlog	3
1.4.3 Draft Schedule	4
2. BACKGROUND THEORY AND RELATED WORK	5
2.1 Game Design Theory	5
2.1.1 Natural Funativity	5
2.1.2 Maslow's Hierarchy of Needs	5
2.1.3 Eight Kinds of "Fun"	7
2.2 Education Design Theory	7
2.2.1 Bloom's Taxonomy	8
2.2.2 Outcome-based Education	9
2.2.3 ADDIE Model	10
2.3 Genetic Algorithm	11
2.3.1 Population Initialization	11
2.3.2 Fitness Evaluation	11
2.3.3 Genetic Operations	11
2.3.4 Knapsack Problem	17
2.4 Usability Testing	19
2.5 Languages and Technologies	19
2.6 Related Works	20
2.6.1 Similar Games Genres	20
2.6.2 Similar Games Example	20
3. METHODOLOGY AND DESIGN	24
3.1 Analysis	24
3.1.1 Problem Statement	24
3.1.2 Target Audience	24
3.2 Design	24
3.2.1 Outcome-based Education (OBE) Design	24
3.2.2 Game Design Theory Application	27
3.2.3 Game Overview	28
3.2.4 Gameplay and Mechanics	31

3.3 Development	39
3.3.1 Learning material	39
3.3.2 Puzzle structure	39
3.3.3 Prototyping and usability testing	43
4. IMPLEMENTATION RESULTS	44
4.1 Usability Testing Result	44
4.2 Implementation Result	44
4.3 User Satisfaction Evaluation Result	55
5. CONCLUSIONS	60
5.1 Work Process	60
5.2 Problems and Solution	60
5.3 Future Works	62
5.4 Conclusion	62
REFERENCES	64
APPENDIX	66
A Storyboard	67
B Learning Material	72
C Puzzles	91
D Character Sprites	106
E Usability Test Responses	113
F Farm and Factory Unlock Requirements	117
G User Satisfaction Survey Responses	119

LIST OF TABLES

TABLE	PAGE
1.1 Product Backlog	3
2.1 The Comparison Between Selection Methods.	13
2.2 The Comparison Between Mutation Methods	17
3.1 Rubric for Assessing the Learner	25
3.1 Rubric for Assessing the Learner	26
3.2 The Category, Topic, and Content in Research Lab System	33
3.3 Activity to Achieve the Learning Level of each Criteria	41
3.4 Assessment of the Activity for Criteria	42
3.5 The Questionnaire for Usability Testing	43
4.1 The Summary Response to Usability Questionnaire	44
4.2 The Summary Response of User Satisfaction Survey to the Overall System of the Game	56
4.2 The Summary Response of User Satisfaction Survey to the Overall System of the Game	57
F.1 Monster Farm Unlock Requirements	117
F.2 Weapon Factory Unlock Requirements	117

LIST OF FIGURES

FIGURE	PAGE
1.1 Draft Schedule of the Project	4
2.1 Maslow's Hierarchy of Needs	6
2.2 Bloom's Taxonomy with Action Verbs	8
2.3 Triangle of Effective Learning	9
2.4 The phrases of the ADDIE Model	10
2.5 Flow Chart of Genetic Algorithm	11
2.6 Tournament-based Selection	12
2.7 Roulette Wheel Selection	13
2.8 Single-point Crossover	14
2.9 Two-point Crossover	14
2.10 Uniform Crossover	15
2.11 Mapping section of PMX	15
2.12 Mapping results in PMX	16
2.13 The final result of PMX	16
2.14 Bit Flip Mutation	16
2.15 Example of Standard Knapsack Problem Setting	18
2.16 Example of Chromosome Representation of the Standard Knapsack Solution	18
2.17 Game Example: Dragon City	21
2.18 Game Example: Princess Connect! Re:Dive (Story Section)	21
2.19 Game Example: Princess Connect! Re:Dive (Arena Section)	22
2.20 Game Example : Omega (Code Writing Section)	22
2.21 Game Example : Omega (Simulation Section)	22
2.22 Game Example: Super Auto Pets (Preparation Phase)	23
2.23 Game Example: Super Auto Pets (Battle Phase)	23
3.1 Example Material for Learning Natural Selection	27
3.2 Game Flow Diagram	29
3.3 Main Game Page Interface	30
3.4 Example of Interface of the Puzzle Gameplay	30
3.5 City Interface	31
3.6 Overall Game State Diagram	32
3.7 Chromosome Representation of the Monster	34
3.8 Breeding Farm Interface	35
3.9 Interface Design of the Battle in the Arena	35
3.10 Interface of Questboard for Accept Quest Scene	36
3.11 Interface of Quest Submission Scene	36
3.12 Interface of the Shop Page	37
3.13 Interface of the Calendar System	37
3.14 Use Cases Diagram	38
3.15 Learning Material for Single-point Crossover.	39
3.16 The Example of Question Scene	40
3.17 The Example of Demonstration Puzzle	40
3.18 The Example of Problem Solving Puzzle	40
4.1 Title Screenshot	45
4.2 Main Hub Screenshot (Overview)	45
4.3 Main Hub Screenshot (Fixing facilities)	46
4.4 Main Hub Screenshot (Quest Overlay)	46

4.5 Farm Screenshot	47
4.6 Factory Screenshot	47
4.7 Habitat Screenshot	48
4.8 Research Lab Screenshot	48
4.9 Cutscene Screenshot (Dialogue)	49
4.10 Cutscene Screenshot (Choice)	49
4.11 Hall of Intelligence Screenshot	50
4.12 Puzzle Screenshot (Selection)	50
4.13 Puzzle Screenshot (Crossover)	51
4.14 Puzzle Screenshot (Knapsack)	51
4.15 Hall of Capybara Screenshot	52
4.16 Calendar Screenshot	52
4.17 City Screenshot	53
4.18 Questboard Screenshot (Overview)	53
4.19 Questboard Screenshot (Detail)	54
4.20 Shop Screenshot	54
4.21 Arena Screenshot (Preparation)	55
4.22 Arena Screenshot (Battle)	55
4.23 Example of Screenshot of the Hall Of Intelligence from the tester	56
4.24 Level of knowledge about topics in Genetic Algorithm before playing the game	58
4.25 Level of knowledge about topics in Genetic Algorithm after playing the game	58
5.1 Project Burndown Chart	61
A.1 Storyboard part 1 out of 4	67
A.2 Storyboard part 2 out of 4	68
A.3 Storyboard part 3 out of 4	69
A.4 Storyboard part 4 out of 4	70
B.1 Learning material 1 out of 6 of the topic: On the Origin of Species	72
B.2 Learning material 2 out of 6 of the topic: On the Origin of Species	72
B.3 Learning material 3 out of 6 of the topic: On the Origin of Species	73
B.4 Learning material 4 out of 6 of the topic: On the Origin of Species	73
B.5 Learning material 5 out of 6 of the topic: On the Origin of Species	74
B.6 Learning material 6 out of 6 of the topic: On the Origin of Species	74
B.7 Learning material 1 out of 4 of the topic: The Genetic	74
B.8 Learning material 2 out of 4 of the topic: The Genetic	75
B.9 Learning material 3 out of 4 of the topic: The Genetic	75
B.10 Learning material 4 out of 4 of the topic: The Genetic	75
B.11 Learning material 1 out of 9 of the topic: The Flow of Genetic Algorithm	76
B.12 Learning material 2 out of 9 of the topic: The Flow of Genetic Algorithm	76
B.13 Learning material 3 out of 9 of the topic: The Flow of Genetic Algorithm	76
B.14 Learning material 4 out of 9 of the topic: The Flow of Genetic Algorithm	77
B.15 Learning material 5 out of 9 of the topic: The Flow of Genetic Algorithm	77
B.16 Learning material 6 out of 9 of the topic: The Flow of Genetic Algorithm	77
B.17 Learning material 7 out of 9 of the topic: The Flow of Genetic Algorithm	78
B.18 Learning material 8 out of 9 of the topic: The Flow of Genetic Algorithm	78
B.19 Learning material 9 out of 9 of the topic: The Flow of Genetic Algorithm	78
B.20 Learning material of the topic: Random Selection	79
B.21 Learning material 1 out of 2 of the topic: Tournament-based Selection	79
B.22 Learning material 2 out of 2 of the topic: Tournament-based Selection	79
B.23 Learning material 1 out of 2 of the topic: Roulette Wheel Selection	80
B.24 Learning material 2 out of 2 of the topic: Roulette Wheel Selection	80
B.25 Learning material of the topic: Premature Convergence	81

B.26 Learning material of the topic: Rank-based Selection	81
B.27 Learning material of the topic: Single-point Crossover	81
B.28 Learning material of the topic: Two-point Crossover	82
B.29 Learning material of the topic: Uniform Crossover	82
B.30 Learning material of the topic: Elitism	82
B.31 Learning material of the topic: Bit Flip Mutation	83
B.32 Learning material of the topic: Flip Bit Mutation	83
B.33 Learning material of the topic: Boundary Mutation	83
B.34 Learning material 1 out of 5 of the topic: Standard Knapsack Problem	84
B.35 Learning material 2 out of 5 of the topic: Standard Knapsack Problem	84
B.36 Learning material 3 out of 5 of the topic: Standard Knapsack Problem	84
B.37 Learning material 4 out of 5 of the topic: Standard Knapsack Problem	85
B.38 Learning material 5 out of 5 of the topic: Standard Knapsack Problem	85
B.39 Learning material 1 out of 5 of the topic: Multidimensional Knapsack Problem	85
B.40 Learning material 2 out of 5 of the topic: Multidimensional Knapsack Problem	86
B.41 Learning material 3 out of 5 of the topic: Multidimensional Knapsack Problem	86
B.42 Learning material 4 out of 5 of the topic: Multidimensional Knapsack Problem	87
B.43 Learning material 5 out of 5 of the topic: Multidimensional Knapsack Problem	87
B.44 Learning material 1 out of 5 of the topic: Multiple Knapsack Problem	87
B.45 Learning material 2 out of 5 of the topic: Multiple Knapsack Problem	88
B.46 Learning material 3 out of 5 of the topic: Multiple Knapsack Problem	88
B.47 Learning material 4 out of 5 of the topic: Multiple Knapsack Problem	89
B.48 Learning material 5 out of 5 of the topic: Multiple Knapsack Problem	89
C.1 Demonstration Puzzle of Roulette Wheel Selection 1 out of 6	91
C.2 Demonstration Puzzle of Roulette Wheel Selection 2 out of 6	91
C.3 Demonstration Puzzle of Roulette Wheel Selection 3 out of 6	92
C.4 Demonstration Puzzle of Roulette Wheel Selection 4 out of 6	92
C.5 Demonstration Puzzle of Roulette Wheel Selection 5 out of 6	92
C.6 Demonstration Puzzle of Roulette Wheel Selection 6 out of 6	93
C.7 Demonstration Puzzle of Tournament-based Selection 1 out of 4	93
C.8 Demonstration Puzzle of Tournament-based Selection 2 out of 4	93
C.9 Demonstration Puzzle of Tournament-based Selection 3 out of 4	94
C.10 Demonstration Puzzle of Tournament-based Selection 4 out of 4	94
C.11 Demonstration Puzzle of Rank-based Selection 1 out of 11	94
C.12 Demonstration Puzzle of Rank-based Selection 2 out of 11	95
C.13 Demonstration Puzzle of Rank-based Selection 3 out of 11	95
C.14 Demonstration Puzzle of Rank-based Selection 4 out of 11	95
C.15 Demonstration Puzzle of Rank-based Selection 5 out of 11	96
C.16 Demonstration Puzzle of Rank-based Selection 6 out of 11	96
C.17 Demonstration Puzzle of Rank-based Selection 7 out of 11	96
C.18 Demonstration Puzzle of Rank-based Selection 8 out of 11	97
C.19 Demonstration Puzzle of Rank-based Selection 9 out of 11	97
C.20 Demonstration Puzzle of Rank-based Selection 10 out of 11	97
C.21 Demonstration Puzzle of Rank-based Selection 11 out of 11	98
C.22 Demonstration Puzzle of Single-point Crossover 1 out of 5	98
C.23 Demonstration Puzzle of Single-point Crossover 2 out of 5	98
C.24 Demonstration Puzzle of Single-point Crossover 3 out of 5	99
C.25 Demonstration Puzzle of Single-point Crossover 4 out of 5	99
C.26 Demonstration Puzzle of Single-point Crossover 5 out of 5	99
C.27 Demonstration Puzzle of Two-point Crossover 1 out of 6	100
C.28 Demonstration Puzzle of Two-point Crossover 2 out of 6	100

C.29 Demonstration Puzzle of Two-point Crossover 3 out of 6	100
C.30 Demonstration Puzzle of Two-point Crossover 4 out of 6	101
C.31 Demonstration Puzzle of Two-point Crossover 5 out of 6	101
C.32 Demonstration Puzzle of Two-point Crossover 6 out of 6	101
C.33 Problem-solving Puzzle of Roulette Wheel Selection	102
C.34 Problem-solving Puzzle of Tournament-based Selection	102
C.35 Problem-solving Puzzle of Rank-based Selection	102
C.36 Problem-solving Puzzle of Crossover 1 out of 4	103
C.37 Problem-solving Puzzle of Crossover 2 out of 4	103
C.38 Problem-solving Puzzle of Crossover 3 out of 4	103
C.39 Problem-solving Puzzle of Crossover 4 out of 4	104
E.1 Response to the usability question 1	113
E.2 Response to the usability question 2	113
E.3 Response to the usability question 3	113
E.4 Response to the usability question 4	114
E.5 Response to the usability question 4 after rearrangement	115
E.6 Response to the usability question 5	115
E.7 Response to the usability question 6	115
G.1 Response to the satisfaction question 2	119
G.2 Response to the satisfaction question 4	119
G.3 Response to the satisfaction question 5	120
G.4 Response to the satisfaction question 7	120
G.5 Response to the satisfaction question 8	120
G.6 Response to the satisfaction question 9	121
G.7 Response to the satisfaction question 10	121
G.8 Response to the satisfaction question 11	121
G.9 Response to the satisfaction question 12	122
G.10 Response to the satisfaction question 13	122
G.11 Response to the satisfaction question 14	123
G.12 Response to the satisfaction question 15	123
G.13 Response to the satisfaction question 16	124
G.14 Response to the satisfaction question 17	124
G.15 Response to the satisfaction question 18	125
G.16 Response to the satisfaction question 19	125
G.17 Response to the satisfaction question 20	125
G.18 Response to the satisfaction question 21	126
G.19 Response to the satisfaction question 22	126
G.20 Response to the satisfaction question 23	126
G.21 Response to the satisfaction question 24	127
G.22 Response to the satisfaction question 25	127
G.23 Response to the satisfaction question 26	128
G.24 Response to the satisfaction question 27	128
G.25 Response to the satisfaction question 28	128
G.26 Response to the satisfaction question 29	129
G.27 Response to the satisfaction question 30	129
G.28 Response to the satisfaction question 31	129
G.29 Response to the satisfaction question 32	130
G.30 Response to the satisfaction question 33	130
G.31 Response to the satisfaction question 34	130
G.32 Response to the satisfaction question 35	131
G.33 Response to the satisfaction question 36	131

G.34Response to the satisfaction question 37	131
G.35Response to the satisfaction question 38	132
G.36Response to the satisfaction question 39	132
G.37Response to the satisfaction question 40	132
G.38Response to the satisfaction question 41	133
G.39Response to the satisfaction question 42	133
G.40Response to the satisfaction question 43	133
G.41Response to the satisfaction question 44	134
G.42Response to the satisfaction question 45	134

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 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. However, some content in this field is hard to study, for example, the Genetic Algorithm. Integrating knowledge about the Genetic Algorithm with computer games provides friendlier ways to access and understand the Genetic Algorithm.

An article suggests there is currently no academic game teaching about the Genetic Algorithm [2]. After some consideration, research, planning, and designing, our group proposed an educational game to educate the players, simulate how the Genetic Algorithm works, and present them to the audiences in an interactive media with the story and incentive to play. To help and motivate them to have more understanding of the said topic, we use puzzle and breeding simulator genres with the design of outcome-based education for teaching, evaluation, and education. We also use role-playing and auto-battler genres with a game design theory to motivate the players and let them have fun while playing the game. Using games as a medium for education encourages players to have the motivation and focus by giving engaging gameplay and pleasing visual and audio aesthetics to the players [3].

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

Using the Knapsack problem as a weapon creation. 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

The place where players can spend their money to buy more monsters or weapons to use in their farms and factories.

ii. Time progression

The system allows players to renew the content in quest, arena and skip past the times used for the breeding process.

iii. Money

The currency for buying more resources for the farms and factories, fixing the broken parts and breeding monsters or weapons to make a new generation.

iv. Arena

The system for players to earn money by using their monsters and weapons to fight the randomly generated enemies.

v. Quest

The system for players to earn money by sending their monsters or weapons that match the requested monsters. The amount of money that players earn depends on the grading of how similar between the sent monsters and the requested monsters.

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 Knapsack Problem
 - ii. Multidimensional Knapsack Problem
 - iii. Multiple 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.	36	1
2	Players can learn about the Genetic Algorithm.	36	2
3	Players can apply knowledge to solve real-world problems.	32	3
4	Players can breed their monsters.	20	4
5	Players have mandatory goals for playing a game.	12	5
6	Players have other activities to spend their time during the breeding phase.	32	6
7	Developers integrate all the systems.	20	7
8	Developers add more aesthetics to the game.	32	8
9	Developers assure quality.	20	9
Total		240	

1.4.3 Draft Schedule

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

Sprint	August				September				October				November				December				January				February				March				April				May			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
Sprint 1																																								
Sprint 2																																								
Sprint 3																																								
Sprint 4																																								
Sprint 5																																								
Sprint 6																																								
Sprint 7																																								
Sprint 8																																								
Sprint 9																																								

Figure 1.1: Draft Schedule of the Project

The goal of the Sprints in the first semester focus on the design. The first Sprint is for research and designing the educational content. The second and the third Sprint are for designing the game systems. And the fourth, which is the last Sprint of the first semester focuses on finalizing the design and document.

The Sprints in the second semester focus on the implementation. The fifth Sprint is designated for prototyping. The sixth and the seventh Sprint are for developing each system in the game. The eighth Sprint is planned for integrating the game. And the ninth Sprint which is supposed to be the last is for testing the full game and finalizing the document 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 theories involving game design, education design, and engineering content like the Genetic Algorithm. Then, we will describe the tools and technologies required to develop the game. Finally, we will discuss the works 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 [4], all fun derives from practicing survival and social skills, which consist of 3 categories.

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. A game with intense and fast action like a first-person shooter (FPS) is a good example of a game introducing this type of fun. For playing such type of game, the player needs to observe the environment using their eyes, continuously control their in-game character using their hands, and act fast to fight the enemy to win the game.

2. Social Fun

The fun of interacting with others. The interaction can be conversations, competitions, cooperation, exchanges, etc. A multiplayer game like a massively multiplayer online role-playing game (MMORPG) is a good example of a game with social fun. In this type of game, the players continuously interact with other players in many ways through conversation, trading, or even competition. These activities sure give players a sense of social fun.

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. The puzzle game, a game in which the players must continuously use their reasoning, planning, or problems solving skills to progress the game, is a perfect example of this type of fun.

All the types of fun do not need to be all in a game to make it a fun game. But mixing more of these aspects of fun will help the game tend to be more fun and popular.

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 [5], representing each level in the Figure 2.1 below.

1. Physiological Needs

It is the most essential need for every human to survive, and it is mandatory to fulfill this kind of need

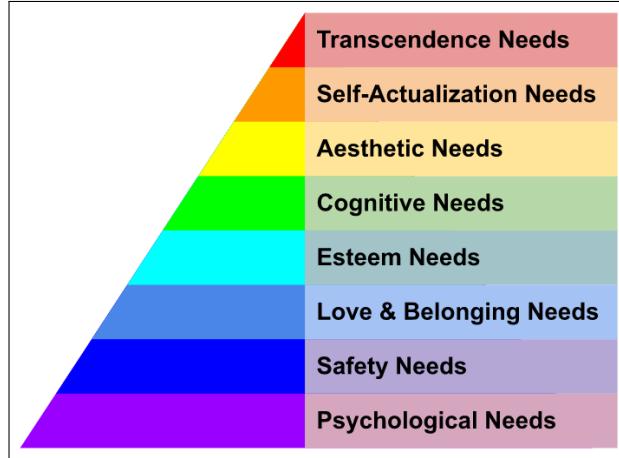


Figure 2.1: Maslow's Hierarchy of Needs

before going further, such as air, water, food, clothing, etc. This level of need may be introduced as a health point system in a game that the player must maintain to survive.

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. This level of need can be used as resources or items that the player could gather to make their character stronger and more stable.

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. The game applying this needs as a social system like co-op, guild, or multiplayer mode.

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. The achievement system of the game is a system directly derived from this level of needs.

5. Cognitive Needs

The need for learning, and gaining intelligence. This includes skill, knowledge, experience, understanding, curiosity, exploration, etc. The in-game character with a skill tree or an experience system is the application of this need.

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. The beautiful graphic of the game is also the aesthetic that fulfills this need of humans.

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. Creating an immersive game that drives the player to be in a way they have not experienced before is an example of fulfilling this need through the game.

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. The fantasy gameplay in which players can use magic; the famous player who helps other players in the game. These are examples of this need in the aspect of the 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 [6].

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 to ensure that our game will surely educate the player. 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 six levels of learning [7]. These levels indicate the depth of learning and are used widely in the field of education. 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.

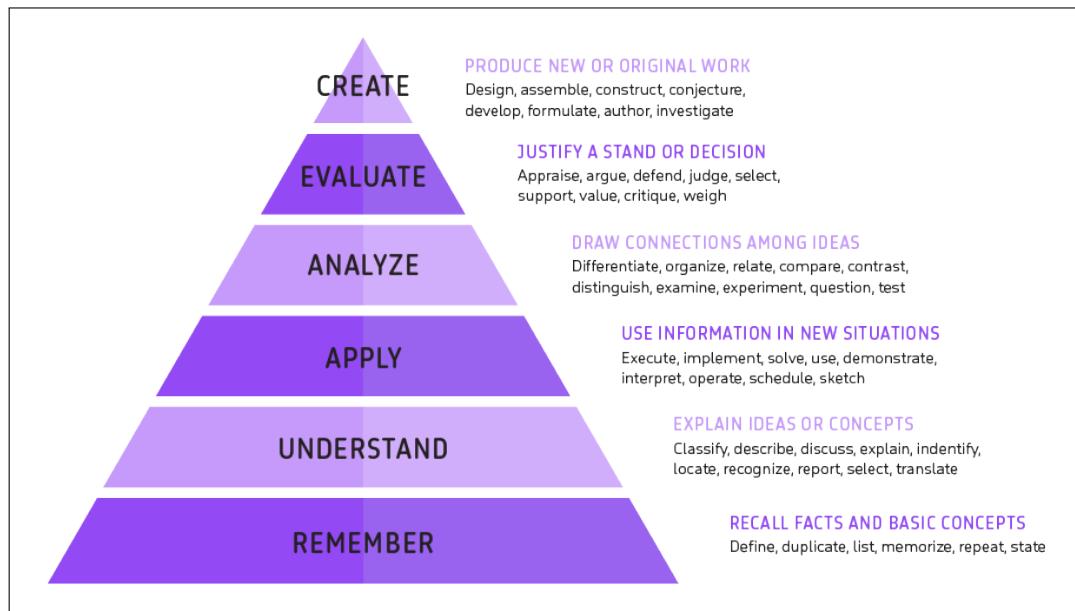


Figure 2.2: Bloom's Taxonomy with Action Verbs [7]

Figure 2.2 shows all the learning levels and their action verbs. Each of them will be described below.

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.

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 outcome [8]. The curriculum is designed backward, starting from the outcome instead of what will be taught.

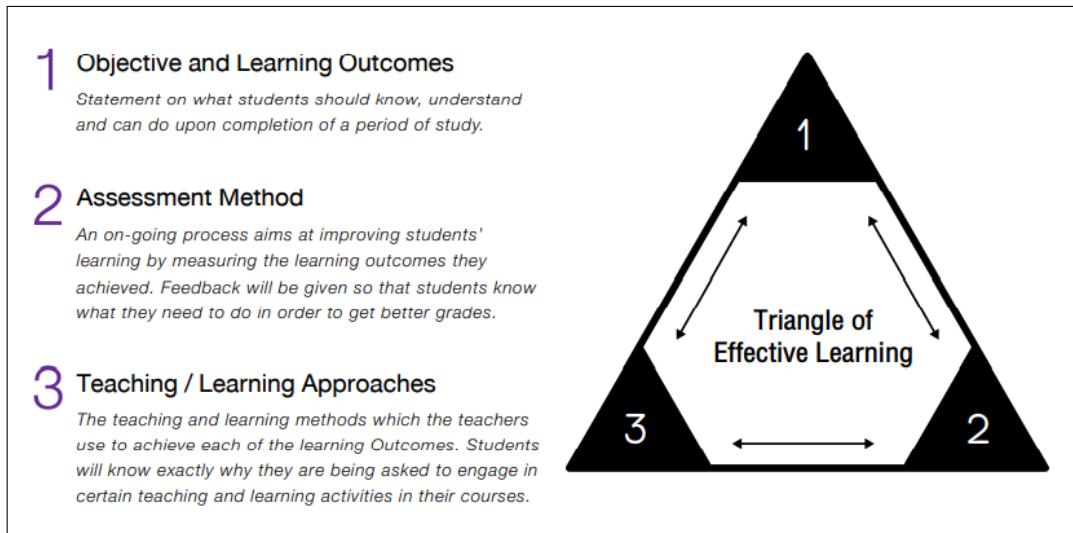


Figure 2.3: Triangle of Effective Learning [8]

According to the triangle of Effective Learning in Figure 2.3, the first step of designing is setting the learning outcome, which are skills 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 [9], 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 for one outcome to consist of only one benchmark. A good rubric should be observable and measurable.

Finally, the teaching and learning activities are designed based on learning outcomes and assessments. This includes not only the materials but also the activities. We adopt this concept in our educational game design 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 [10]. 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.

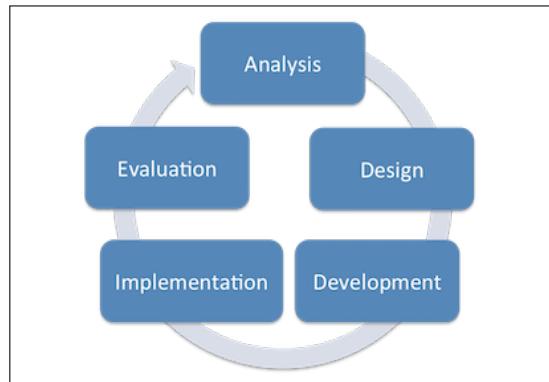


Figure 2.4: The phrases of the ADDIE Model [10]

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. Those contents can be used later for the pilot test, a small-scale educational simulation used for gathering data about the feasibility of the project [11]. Such a test is applied to the group of target students to collect the necessary feedback for content revising.

4. Implementation

This step focuses on preparing the people involved in education and implementing the actual learning resources, such as content, material, and tool 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 two methods of evaluation are formative evaluation and summative evaluation. Formative evaluation is the type of evaluation that is conducted at every stage of the ADDIE model and focuses on ongoing project revision. Summative evaluation occurs after the implementation, aiming to assert the learner's outcome and the effectiveness of the educational program once the course is completed.

2.3 Genetic Algorithm

The Genetic Algorithm is the computer algorithm adapted from the Darwinian theory of Natural Selection [12]. The algorithm introduces an easy-to-understand way to solve complicated problems since it's a technique with very little mathematics. Even deadlocked problems, the complex problems with conflicting conditions that require simultaneous solutions previously considered difficult can be solved with Genetic Algorithm [13].

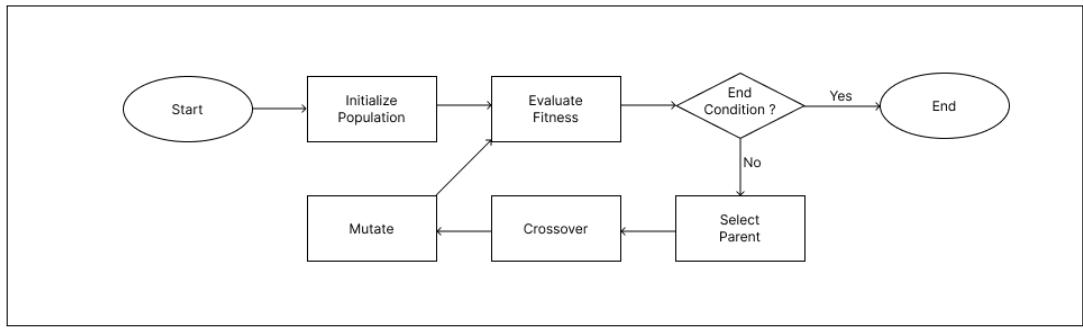


Figure 2.5: Flow Chart of Genetic Algorithm

The algorithm begins with encoding the possible solutions to the problem into a data structure called a chromosome. Such data structure usually is an array of bit values, the value that can be only 0 or 1. Other forms of data structure can be similarly used, for example, the matrix of the integer. Once the chromosome population is generated, they will be repeatedly evolved through the biological-inspired processes as shown in Figure 2.5. Commonly, the Genetic Algorithm is used to generate the optimal solutions, the best solution among a vast number of generated possible solutions, for optimization and search problems. The detail of each process in the algorithm is described as follows.

2.3.1 Population Initialization

Usually, the population initialization will be done via randomizing the set of valid possible solution chromosomes, the chromosomes that satisfy all the constraints. 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.

2.3.2 Fitness Evaluation

The fitness of a chromosome will be evaluated using specific methods, including fitness function and constraint. In a more general context, the fitness function is an objective function of an optimization problem which is a function that calculates the value we wish to maximize or minimize. In the context of the Genetic Algorithm, the fitness function is used to calculate the fitness value that indicates how close to the optimum solution of the chromosome is. If any chromosome violates the constraint, it is considered an invalid solution, and the fitness value of that chromosome will be zero.

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 during the parent selection process. The selected parent will inherit part of its chromosome to the offspring in the next generation through the crossover process. Meanwhile, the chromosome with lower fitness is likely to be destroyed.

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

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, 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.

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. For example in Figure 2.6, the chromosomes are first picked as a group of 3, and the chromosome “A” is selected from the group due to its highest fitness value.

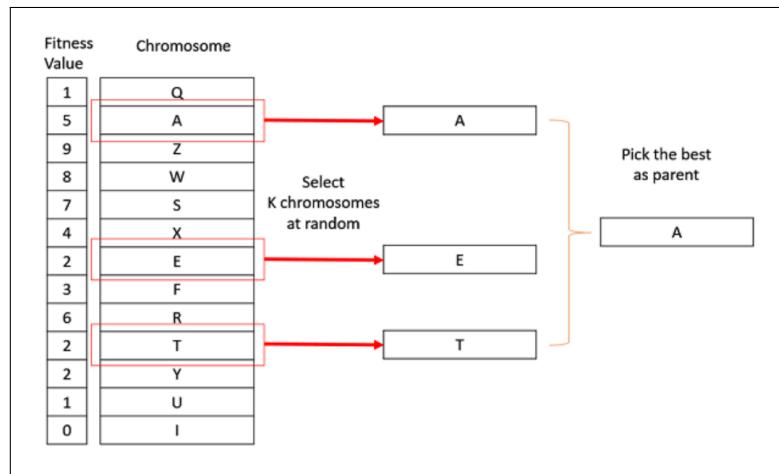


Figure 2.6: Tournament-based Selection [12]

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 being selected proportionate to its fitness. The higher fitness of the chromosome, the larger space on the wheel.

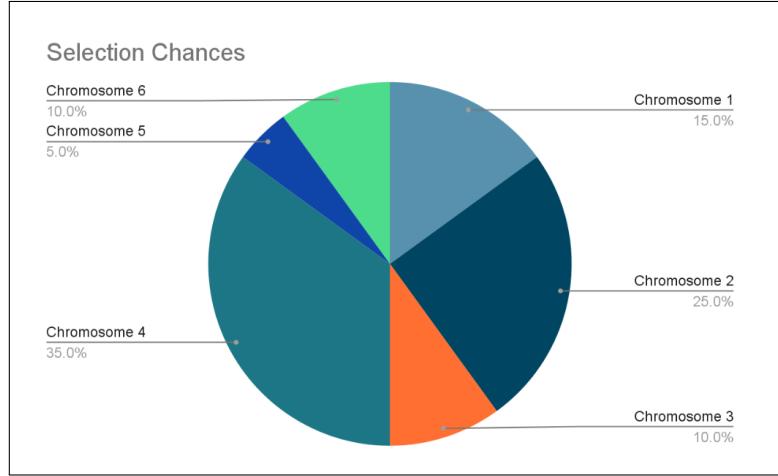


Figure 2.7: Roulette Wheel Selection

The selection chance for each chromosome is calculated using its fitness value divided by the total fitness value of all chromosomes in the population. In Figure 2.7, given that chromosome 4 has a fitness value of 7 and the total fitness value among the population is 20. So, the selection chance of chromosome 4 will be 7 divided by 20 which is equal to 0.35, or 35 percent.

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. This can be done by reassigning the fitness value of the chromosome using the reversed rank. The worst chromosome will have fitness 1, the second worst will have fitness 2, and so on [14]. The best chromosome will have the highest fitness equal to the number of chromosomes in the population.

Table 2.1: The Comparison Between Selection Methods.

Selection Method	Advantage	Disadvantage
Random	It is easy to understand and implement.	There is no use in the fitness value; the good solution might be accidentally destroyed.
Tournament-based	It can be implemented efficiently since no extra calculation like a sorting is required.	If the tournament group is large, the weak chromosome will have a smaller chance to be selected.
Roulette Wheel	The chromosome with higher fitness has more chance to be selected.	It's not fair because the worst chromosome will barely be selected.
Rank-based	All chromosomes have a similar chance to be selected which means it preserves diversity.	Computationally expensive due to the sorting process. The best solution is found slower since the difference between each chromosome is small.

All the parent selection methods have their advantages and disadvantages. The comparison between these methods [14] is discussed in Table 2.1.

5. Elitism

Elitism is a method that helps preserve the elites, 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. However, increasing the ratio of the elites too much hurt the algorithm due to decreasing in population diversity. From the experiment, the algorithm with a higher reliability requirement needs a higher population size but lower elitism rate [15]. So, the elitism rate is preferred to be a small percentage value around 5% to 10% [16].

- Crossover

The crossover is the process where a pair of selected 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 and k-point crossover, uniform crossover, and partially-mapped crossover (PMX).

1. Single-point Crossover

The process of this type of crossover is illustrated in Figure 2.8. The process begins with randomly selecting a single point on both parents' chromosomes as a crossover point. Then, the gene information from the right 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. This is one of the most simple crossover methods and it is also one of the lowest computational cost methods.

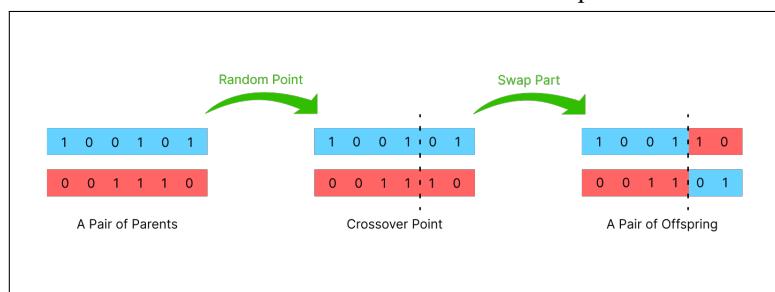


Figure 2.8: Single-point Crossover

2. Two-point and K-point Crossover

The process of the two-point crossover and the k-point crossover is the same. The only difference between them is the number of crossover points. Figure 2.9 shows the processes of the two-point crossover. Two points from both parents' chromosomes are randomly selected to be crossover points, dividing the chromosome into 3 sections. Then, the gene information in the second section (from left to right) which lay between these two points and is swapped between the parents to create two new offspring. Note that the first and the third sections of the chromosomes which are odd-numbered sections stay still.

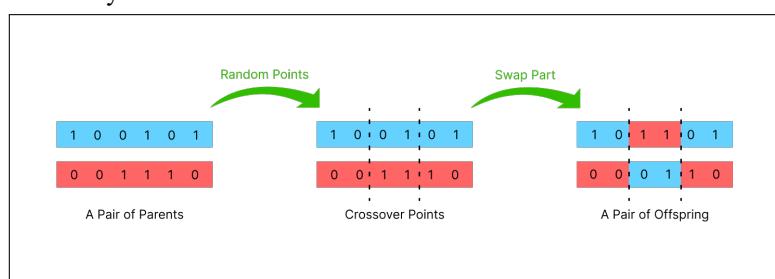


Figure 2.9: Two-point Crossover

For the k -point crossover, the k crossover points are randomized, divide the chromosomes into $k + 1$ sections. Same as the two-point crossover, the odd-numbered sections stay still. In the other hand, the even-numbered sections swapping between parents. This type of crossover help the chromosome be able to exchange its subset of information for multiple times.

3. Uniform Crossover

In this type, as shown in Figure 2.10, the offspring are created by randomly swapping every piece of information in the parent's chromosome. The chance of switching each point is independent of another. This type of crossover allows the offspring to inherit from both parents equally and independently.

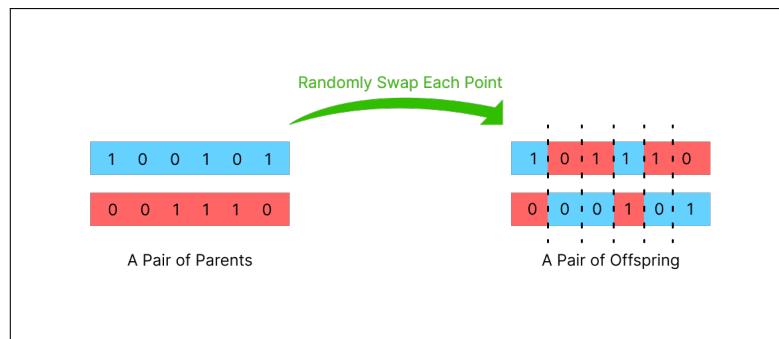


Figure 2.10: Uniform Crossover

4. Partially-Mapped Crossover (PMX)

Whereas other previous methods cannot be implemented on the chromosome that can contain a duplicate value, the PMX can. To be accurate, PMX is a crossover method that only is used with the chromosome with such a condition. The method begins with randomly selecting two crossover points and swapping the chromosome information in the middle section. The swapped section is also called the mapping section since each information point on the parent is mapped with the corresponding point on another parent. Figure 2.11 shows an example of such mapping.

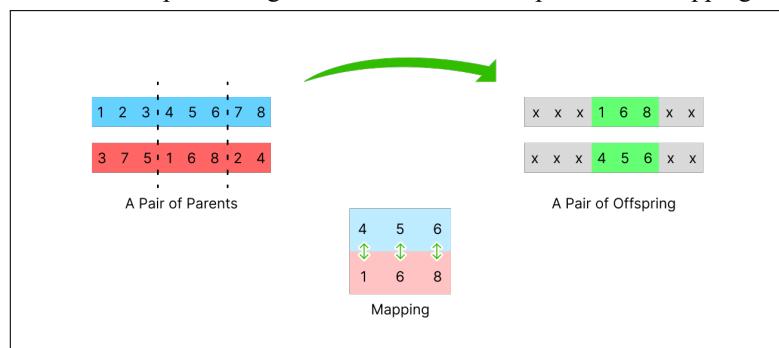


Figure 2.11: Mapping section of PMX

Then the information from its parent in the same position transfer to the offspring. If such information already exists in the offspring, that information will be changed into the corresponding value in the mapping until it turns out to be information that never exists in the offspring. In Figure 2.12, the value 1 already exists in the offspring. So, it is changed into its mapping value which is 4.

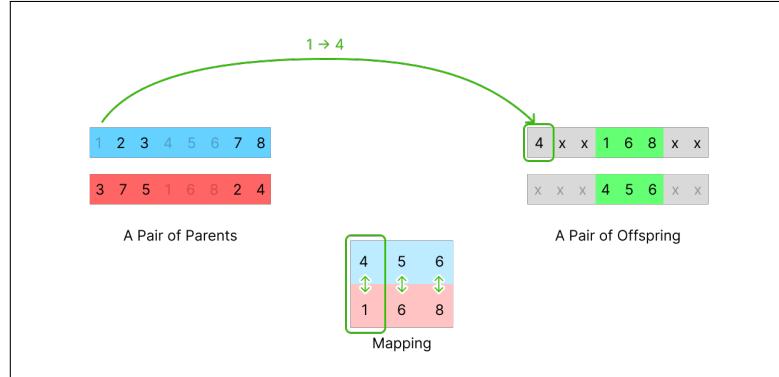


Figure 2.12: Mapping results in PMX

The rest of the information in both offspring is transferred in the same way. Figure 2.13 shows the final result of the PMX operation.

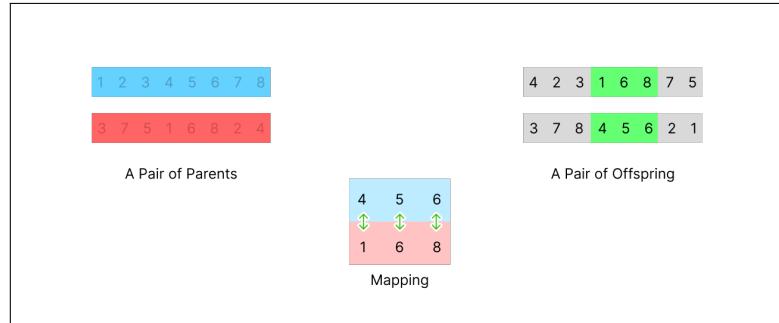


Figure 2.13: The final result of PMX

- 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. But just like Elitism, setting the mutation rate too high makes no good to the algorithm. If we use a high mutation rate, the algorithm would act as a random algorithm that does not utilize the advantage of using other genetic operations. So, the mutation rate is preferred to be a low value similar to the elitism rate. There are different types of basic mutation including bit flip mutation, flip bit mutation, and boundary mutation.

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 2.14, the highlighted bit is changed from 0 to 1.



Figure 2.14: Bit Flip Mutation [12]

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.

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.

Table 2.2: The Comparison Between Mutation Methods

Mutation Method	Amount of Changed value	Suitable Type of Chromosome
Bit Flip	Randomized individual value within a chromosome	Binary bit string
Flip Bit	All information in randomized chromosome	Binary bit string
Boundary	Randomized individual value within a chromosome	Integer or float

Each mutation method suits a different type of chromosome and changes a different amount of value. Table 2.2 shows a brief comparison of each method.

2.3.4 Knapsack Problem

The real-world problem involving our project is the knapsack problem and its variant, it's one of the most popular optimization problems that can be solved using Genetic Algorithm [17]. 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 [18]. 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.

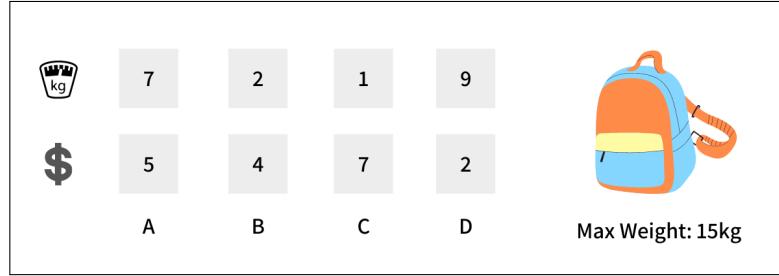


Figure 2.15: Example of Standard Knapsack Problem Setting [18]

Figure 2.15 shows the example of Standard MKP. There are four items which are A, B, C, and D. Each item has a weight of 7, 2, 1, and 9; and has a value of 5, 4, 7, and 2 respectively. The constraint is that the knapsack can hold a maximum total weight equal to 15.

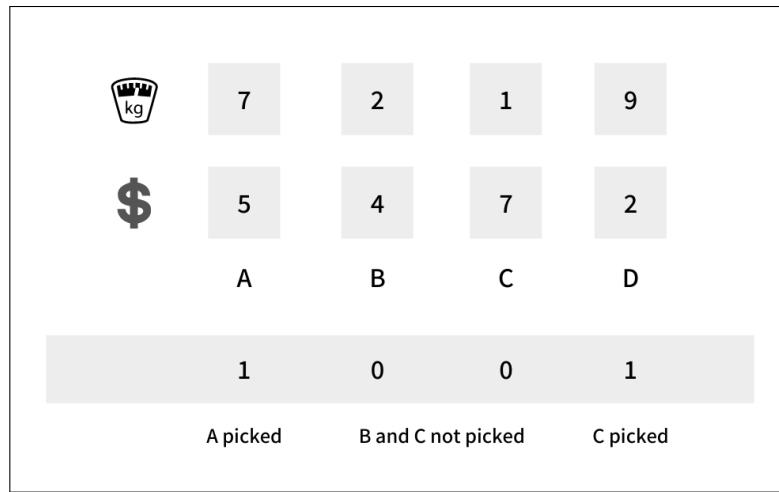


Figure 2.16: Example of Chromosome Representation of the Standard Knapsack Solution [18]

Since there are four items, the binary bit string chromosome with a length of four can be used as an encoded representation of the solution. Figure 2.16 demonstrates the chromosome with the value of 1, 0, 0, and 1. By decoding these values, we get the solution which is picking items A and D into a knapsack. Using encoding and decoding this way, we can represent any solution using the binary bit string; and we can finally use a Genetic Algorithm to solve such a problem.

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.

2.4 Usability Testing

Usability testing aims to understand how users interact with the product and find issues in existing products [19]. The information gathered from the test can be used to improve the design or current development. In general, the usability testing questions are divided into 3 sets which are screening, in-test and post-test.

The screening question indicates whether the participants represent the target group. The data from this type of question includes both demographic and personal experiences. The in-test question aims to find insight into user behaviors such as the reason behind the user's certain actions. The insight from this section usually be generated during the user's interaction with the product. The post-test question aims to gather the user's additional feedback. The opinion gathered from this section can be useful for product improvement.

With the reason for the benefit-cost ratio, the recommended test user size is 5 [20]. With this size of the test user, almost all usability problems will be found while the effort of performing the test is not too high.

2.5 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.

1. C#

C# (pronounced "See Sharp") is one of 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.

2. Unity

Unity is a popular free game engine used to create both 3D and 2D games for multiple platforms; in 2020, 61% of surveyed developers decided to use Unity as their preferred game engine [21], and the number increase by 31% in 2021 [22]. 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. Compared to the similarly popular game engine, Unreal Engine [23], Unity is considered more lightweight and beginner-friendly, with various platform integrations and dedicated tools for 2D game development [24]. Apart from said features and our previous experience with this game engine, we also chose Unity due to its popularity, which makes the community pretty huge and helps when we run into trouble while developing games.

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 has the ability to design all kinds of graphics, including websites, mobile app interfaces, prototyping designs, etc. The fact that Figma is available on multiple platforms containing various sets of designing tools available to use and collaboration features for team projects [25] makes it to be one of the most widely-used graphic tools.

6. Aseprite

Aseprite is a pixel art tool for creating pixel art images and 2D animations. It offers various tools for creating 2D art, such as shading, pixel-perfect strokes, tiled mode, and filled and contour options. Additionally, it provides animation tools, including a real-time animation preview feature that helps users create pixel art or animation. It allows users to create multiple layers of art, separating the usage of layers for still images and frames for animations. This makes it easier to have art with multiple layers and create animations by specifying a specific layer in each frame.

2.6 Related Works

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

2.6.1 Similar Games Genres

Our game can be considered as various genres including visual novels, puzzles, and education.

1. Visual Novel

A Visual Novel (VN) is a game genre that mainly focuses on story narration through text with the help of static arts, sound effects, and soundtracks, similar to a graphic novel as per the name suggested [26]. Players can interact with the game by clicking or tapping to progress the story; this system is called On-Click Progression. Furthermore, the game may provide players with extra interactions, such as dialogue choices or actions that may affect the story.

2. Puzzle Game

A puzzle game is a problem that has a correct answer for players to discover [27]. Players must understand the concepts and features of the game and apply those rules with one's logical thinking to reach the goal. A good puzzle game should be entertaining and have an appropriate difficulty level for its target group.

3. Educational Game

An educational game is a type of game that integrates academic and training factors into its gameplay, story narration, and other elements [28]. It requires players to use various skills such as problem-solving, strategizing, and other higher-order thinking skills in order to achieve goals and receive rewards.

2.6.2 Similar Games Example

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.

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. We decided to integrate their theme of collecting monsters and putting them on a farm into our game.

Due to its multiplayer-based nature, players can communicate and group up with other players as a guild to share information and passion about the game with each other, enjoying their given social fun. They can also participate in competitions between players to climb to higher ranks, fulfilling their esteem needs. There are countless dragons for players to discover and collect, giving them discovery fun.



Figure 2.17: 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. We carefully extracted some of the arena elements into our game, more details will be covered in the next chapter.

Similar to the previous game, this game has multiplayer features like the Guild system and PVP system for players to gain esteem needs, social fun, and fellowship fun. It also has tons of refined characters to collect. Moreover, there are multiple engaging voice-over stories with animations displayed as a visual novel during the story telling section, completing players' narrative fun.



Figure 2.18: Game Example: Princess Connect! Re:Dive (Story Section)
Source: Cygames Inc. (2023). Princess Connect! Re:Dive. Screenshot by Author.



Figure 2.19: Game Example: Princess Connect! Re:Dive (Arena Section)
Source: Cygames Inc. (2023). Princess Connect! Re:Dive. Screenshot by Author.

3. Omega

Omega is a puzzle game developed by ORIGIN Systems, Inc in 1989 [29]. 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 2.20: Game Example : Omega (Code Writing Section) [29]



Figure 2.21: Game Example : Omega (Simulation Section) [29]

4. Super Auto Pets

Super Auto Pets is a free-to-play auto-battler created by Team Wood Games, publicly released on

Steam on 24 September 2021, before later released on mobile platforms. Players prepare their teams by buying new pets from a randomized pool and purchasing food to enhance their pets in the preparation phase as in Figure 2.22. Then they can send their pets to battle with other players in the battle phase as in Figure 2.23. These pets battle automatically, as both pets on the front row will attack each other simultaneously; if one's health point reduces to zero, it will faint and be out of combat. After the impact concludes, the most frontal pets strike again until a team, or both, has no pets left standing. Same as Princess Connect! Re Dive, we also extracted some of the battle features into our game.

The game contains varieties of pets with different kinds of stats and skills for players to use in their team, letting players discover ways to synergize their pets and food the best within given limited resources, satisfying discovery fun. The game does not require players to make decisions quickly, and they can exit the game anytime during the preparation phase, thus making the game considered casual and able to satisfy players' submission fun.



Figure 2.22: Game Example: Super Auto Pets (Preparation Phase)

Source: [Super Auto Pets on Steam](#)

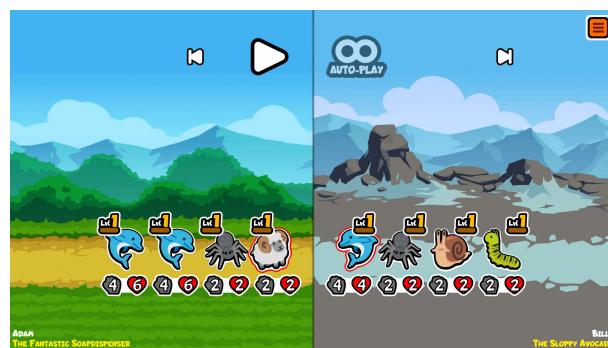


Figure 2.23: Game Example: Super Auto Pets (Battle Phase)

Source: [Super Auto Pets on Steam](#)

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

People who want to learn about Genetic Algorithms, especially university students who are studying in the relevant faculties such as computer engineering. 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 [8]. 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 [7] 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, Multidimensional, and Multiple Knapsack Problem.	Able to demonstrate the chromosome encoding and decoding of the Standard and Multidimensional 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 and Multidimensional Knapsack Problem.	Able to solve the chromosome encoding and decoding of a 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 [30]. 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.

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 result of a puzzle that they must complete with the correct processes, generating the result corresponding to the expected result.
3. The evidence involving problem-solving is a result of a puzzle similar to the demonstration, but the puzzle in this level gives more choice to the player. So, the player must decide what type of method to perform from the given description through the puzzle's element selection. Then the player should perform the method like in the demonstration.

The evidence involving explaining and demonstrating for assessment levels 1 to 3 will be collected from the test after completing the lesson in the research lab section of the game and All the types of

evidence of every level will be collected more from the result of the puzzles used for fixing the facilities when they are broken.

- 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.

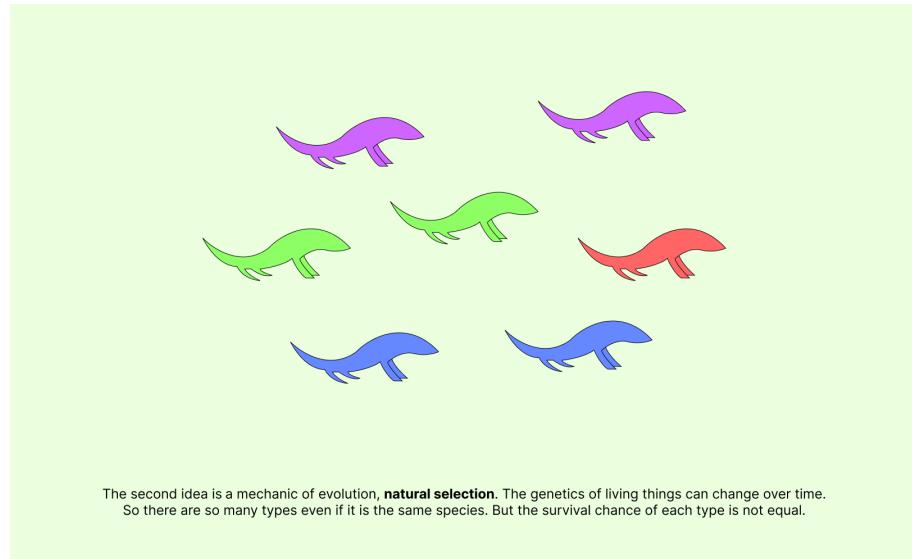


Figure 3.1: Example Material for Learning Natural Selection

Figure 3.1 is an example of the material we have already prepared for Natural Selection, one of the basic biology concepts the Genetic Algorithm is based on. The material in the figure includes the image on the center and description about natural selection at the bottom. The full learning materials are in Appendix.

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

The game we designed mainly focuses on Mental Fun categories of Natural Funativity by Noah Falstein. 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.

The game we designed caters to the needs of the players following Maslow's Hierarchy of Needs to make players feel fulfilled and have answers to their needs when playing the game. The first need is Physiological

Needs where the players must obtain resources to breed stronger monsters to survive the battle inside the arena. Next is the Safety Needs where the players can improve the status of their monsters and weapons by creating new generations. For the Esteem Needs of the players we design the game to assign rank and rewards for completing quests based on the quality of the requested item to help fulfill the players when they get high rank and reward when completing the quest or win the battle in the battle arena. For the Cognitive Needs, The game was designed to teach players about genetic algorithms and how to solve the problem with this knowledge. The game's Aesthetic Needs are met through its music and visually appealing rewards with higher rankings and more powerful weapons resulting in more elaborate and appealing rewards. Lastly, the game design caters to the Self-Actualization Needs of the players by allowing them to manage their farm, factories, money, and resources according to their preferred playstyle and choose how to spend their resources and how to obtain them by themselves. The game design's hierarchy of needs is well-thought-out, providing players with an immersive and satisfying experience.

The game we design aims to provide different kinds of fun that cater to the players' preferences. The first kind of fun is Sensation, where the game's sci-fi and fantasy theme aesthetic stimulates the senses through visuals, character design, music, and sound effects from the players' interactions. The second kind of fun is Fantasy, where the game's story, world, and characters help the players feel immersed in the game and give them a role to play with objectives to accomplish in the story. The third kind of fun is Challenge, where players must complete quests and battle opponents in the arena to obtain the resources to upgrade their farms and factories. As the game progresses, the quests and opponents become more difficult, providing a sense of accomplishment upon completion. Lastly, the game design provides Expression, where the players can freely manage their money and resources, and choose the combination of weapons and monsters for the battle arena that suits their playstyle. The game design's various kinds of fun make the game engaging and enjoyable for all types of players.

3.2.3 Game Overview

As we mentioned, we decided to develop the game using an approach in which Computer Assisted Instruction (CAI) is designed first, then adding the game systems to transform it into Digital Educational Games (DEG). The main reason for adding the game systems is to increase the external learning motivation of the player so that the player has more will for playing and learning in the game. The game overview will describe how we will create the CAI and what kind of game systems we will add.

- Game Concept

Our game will focus on three main aspects: obtaining skills, applying skills, and other activities. The player will be played as one participates in the global-guardian project, the project for creating a powerful combat force used for future fighting with the alien. To do that, the player has to research through visual novel gameplay and validate their skills using the puzzles. Then, applying their genetic algorithm skills to weapon and monster breeding. Lastly, the player can perform other activities that don't directly involve the genetic algorithm but may be related. Battles using their bred weapon and monster is an examples.

For the first aspect, obtaining skills, there is a set of main systems involved. The player has to research via the cutscene explaining the subject. The research result (explanation for that subject) is kept in the research lab for review in the future. After that, the player is required to pass a skill check using the puzzles. And the skill-checking results are all kept in the hall of intelligence, a place a game progress is shown and can be translated into the learning outcome evaluation further.

For the second aspect, applying skills, there is a set of primary supporting systems involved. The usual

visual novel and puzzle gameplay that focuses on education may be boredom since the learning motivation may not be enough. To drive the motivation of the player, we design supporting systems that the player should use their learned skills to create a weapon using the weapon factory and breed the monster using monster farm. To build more facilities and unlock more weapons, the player is required to complete certain research. Since the factory and farm break down after some time passes, the player must solve the same puzzle as skill validating to fix these facilities.

For the last aspect, other activities, there is a set of secondary supporting systems involved. The battle arena is a system where the player can battle with the Non-Player Character (NPC) using their bred weapon and monster. There is also a side quest where the player is required to deliver a monster with some specific appearance. Another system is a random event where a random specie of Capybara walk by facilities, and the player can capture it by rapidly clicking on it. All these systems are completely optional and don't really affect education. But they can give the player great motivation to keep playing the game.

- Game Flow Summary

As an educational game, we conduct the game flow linearly to help the player get the basic before going into more advanced subjects. Figure 3.2 shows the game flow diagram in terms of progression and game systems. The player must achieve the condition at the arrow tail before being able to achieve the head of an arrow. For example, the player must obtain skill 1 to finish main quest 1.

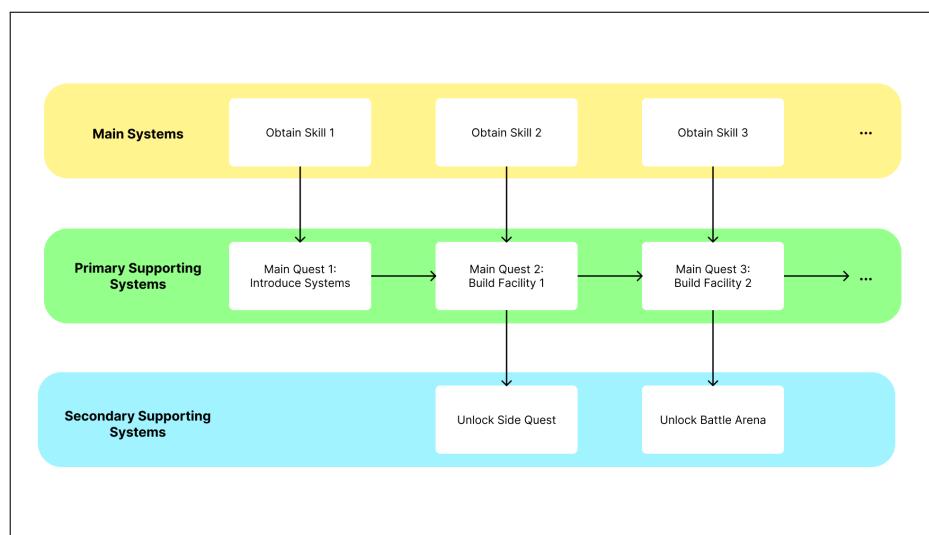


Figure 3.2: Game Flow Diagram

The game begins with the introduction to the main systems, the Research Lab and the Hall of Intelligence. After completing the early learning and getting the early skill, such as the ability to explain basic biology, the main quest will conduct the player to build more facilities that require a higher level of skill corresponding to the diagram shown in Figure 3.2. Figure 3.3 shows the main game page interface where the player can access any facility and all other systems.



Figure 3.3: Main Game Page Interface

After players unlock and build the facility, they can use that facility in the way their want. The facilities have a chance that their durability goes down after breeding. A decrease in durability results in a decrease in breeding speed and the breeding is completely stopped when all the durability run out. 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 3.4 shows an example of the interface of the puzzle the player needs to solve to fix the facility. The full gameplay of all the puzzle material can be found in Appendix.

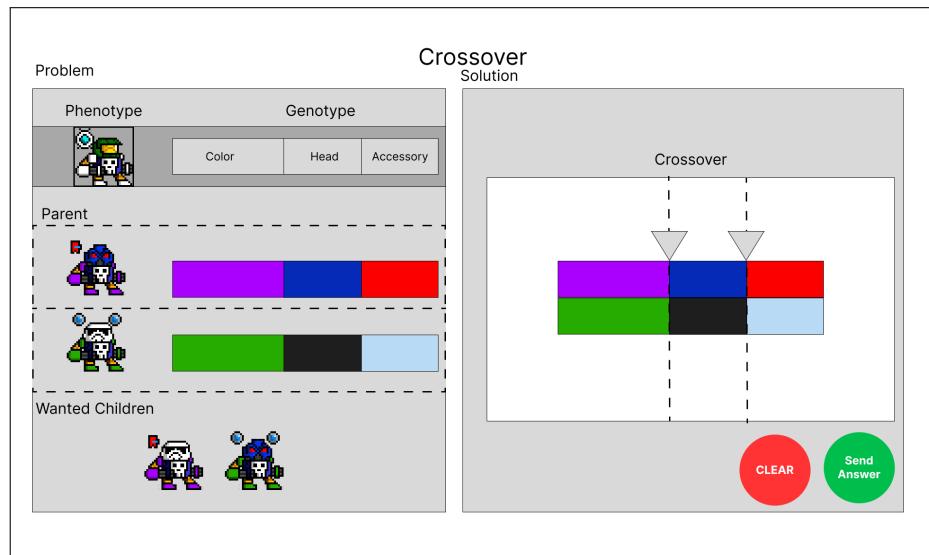


Figure 3.4: Example of Interface of the Puzzle Gameplay

Even if the main quest conducts the player continually unlocks the new facility which requires new research, the player also can enter the city freely to access other activities. Figure 3.5 shows the city interface where the player can go to the quest board and receive a side quest, buy a new monster from a shop, and battle in the arena using their own monster.

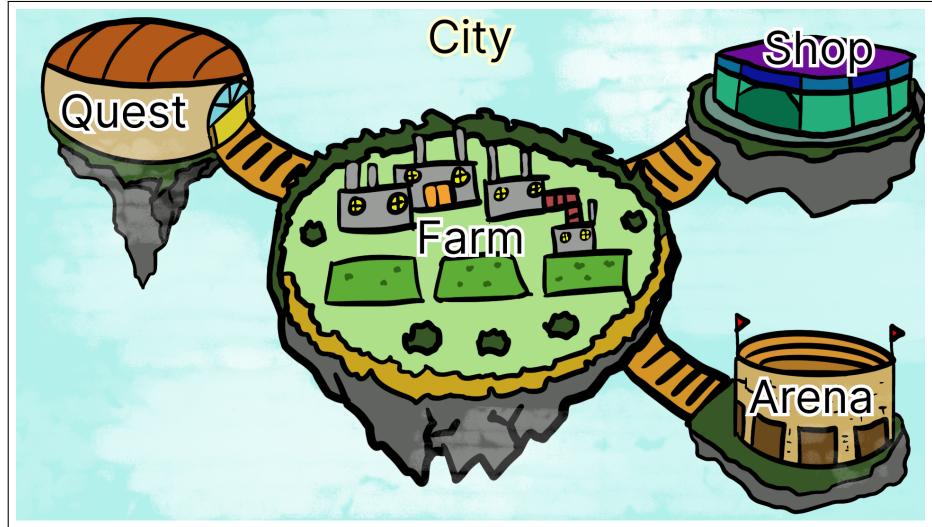


Figure 3.5: City Interface

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 unlock all the factories. Since the player is required to pass a skill-checking before unlocking a new factory, the player will be considered as completing the game as well as the learning outcomes upon completing the last main quest. And the player is allowed to continue playing in other systems freely without any restriction.

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 will be played as a company owner who participates in Project-G, a global-guardian project for creating a powerful combat force for future fighting with the alien. 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 passing a skill check for certain subjects about genetic algorithms, having enough resources, and building prior facilities.

Building new 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 a monster from their company, resulting in a reduction in the overall population, which can be solved by buying new ones from the shop.

After all the factories are unlocked, the game story ends since all combat power against aliens is already prepared, and the player is evaluated as completing learning as well. The minimum requirement for unlocking all factories is passing a skill check for performance level 3 for all criteria

described in Table 3.1, which is determined as a minimum level to pass the education. The player also be able to play the game furthermore without any restrictions.

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.

- System Details

The game systems can be described as states of the game. The states of the game and its transition are shown in Figure 3.6 below.

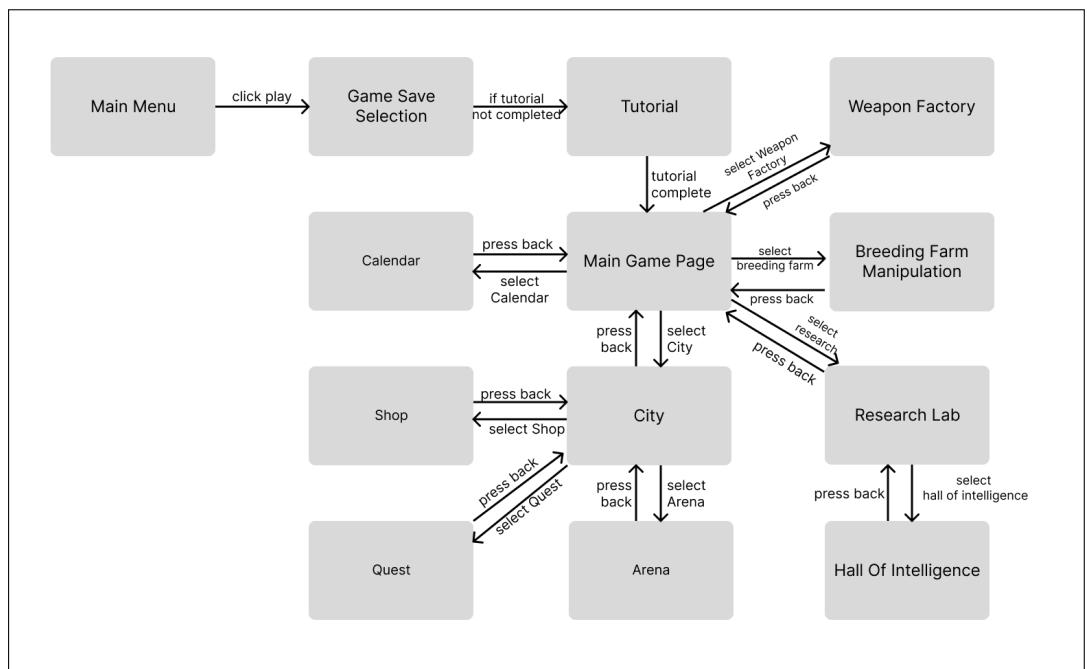


Figure 3.6: Overall Game State Diagram

- Main Systems

These features are used for teaching and testing the players' knowledge based on the assessment criteria and level described in Table 3.1 mentioned earlier.

1. Cutscence

This system is designed to guide players in performing specific tasks to learn how to play the game and progress through its story. Cutscenes serve another purpose by teaching players and testing their knowledge of the lessons that were taught. Cutscenes can prompt players for answers through dialogue choices and evaluate their responses to determine whether they pass the test or not.

2. Research Lab

This system allows players to learn about the Genetic Algorithm and how to solve real-world

problems using it. After finishing the learning on a specific subject through the cutscene, the summary of the content will be kept in the research lab for review in the future. 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 Knapsack	The problem setting, goal, chromosome encoding, and fitness evaluation of the Standard Knapsack Problem will be described.
	Multidimensional Knapsack	The problem setting, goal, chromosome encoding, and fitness evaluation of the Multidimensional Knapsack Problem will be described.
	Multiple Knapsack	The problem setting, goal, chromosome encoding, and fitness evaluation of the Multiple Knapsack Problem will be described.

3. Hall of Intelligence

This system is an extension of the Research Lab mentioned earlier. The Hall of Intelligence provides a platform for players to test their knowledge acquired from the learning materials in the Research Lab and demonstrate their learning progress. Additionally, the Hall of Intelligence tracks players' game duration, current quest status, and the number of times they have played the puzzles or tests.

4. Assessment Methods

This is a system of tests that players are required to pass to keep progressing through the game. The assessment methods for the players based on the performance descriptors outlined in Table 3.1 can be categorized as multiple-choice questions and puzzles.

Multiple-choice questions utilize the cutscene system mentioned earlier to evaluate how well players can explain and classify their knowledge. The test consists of a series of multiple-choice questions, where players need to choose from dialogue choices to provide their answers. The total scores are calculated based on all the answers given, and if players achieve passing marks, they successfully clear the test. However, if they do not meet the passing marks, they will fail the test.

The puzzles test is designed to assess players' knowledge acquired from the lessons by allowing them to demonstrate the step from the lesson or solve the problem related to each lesson's topic. Each topic has its own platform with specific puzzles. The puzzle platform consists of Parent Selection Puzzle, Crossover Puzzle, and Knapsack Formulation Puzzle.

– Primary Supporting Systems

The systems include a monster farm and weapon factory where the player should use a portion of knowledge to play with. These systems are used to control the game progression using main quests since they are gameplay added to directly reinforce the will for learning of the player.

1. Monster Farm

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 as shown in Figure 3.7, each section representing each aspect of a monster including an appearance and battle status. The appearance is their head type, body color, and accessory type. The battle status includes attack, defense, health points, and speed.

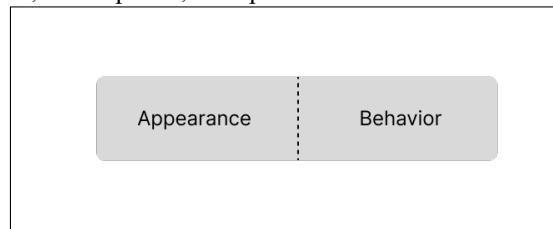


Figure 3.7: Chromosome Representation of the Monster

Players can adjust several parameters involved in the algorithm, such as the parent selection and crossover method. In addition to the basic operations of the Genetic Algorithm, players can utilize elitism, allowing them to preserve their favorite monster's chromosome for the next generation. The number of monsters on a farm and the number of breeding generations affect the amount of currency consumed which introduces the management gameplay into a game. Such currency is acquired from the main quest completion and other activities like a battle in the arena or capturing Capybara. Figure 3.8 shows the interface of the breeding farm. The players can configure the generic algorithm of the farm in the setting on the right side and set the number of generations of breeding before starting the process by paying the to-

tal money. The player can look at the monster set in the habitat or look at the monster's detail.

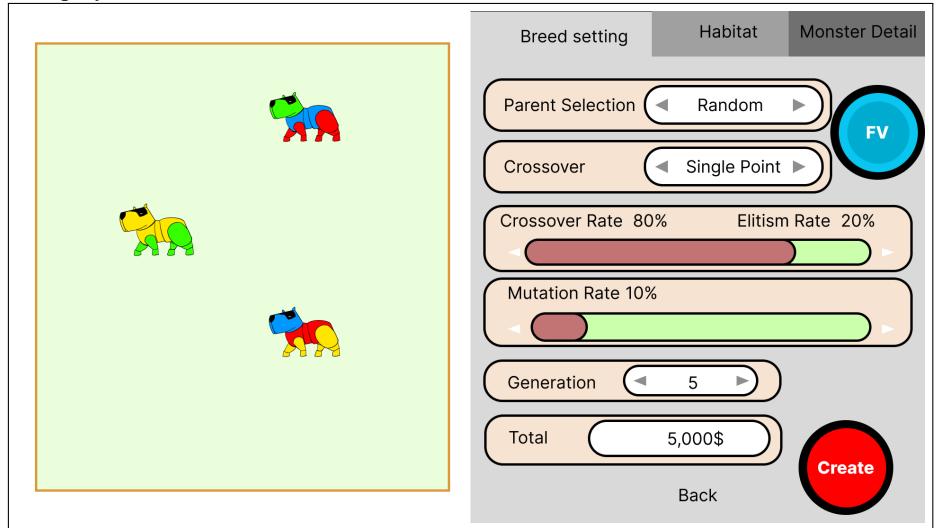


Figure 3.8: Breeding Farm Interface

2. Weapon Factory

This system allows players to test their skills in solving real-world problems using the Genetic Algorithm. The weapons obtained from the factory are the result of attempts to solve the knapsack problem. The fitness values of each weapon's solution determine its effectiveness and appearance. These weapons can enhance the battle status of monsters, making them stronger in combat within the arena, and provide unique bonus effects specific to each factory's weapon type.

– Secondary Supporting Systems

This is a set of systems that don't require any knowledge or skill to play with. The systems include a battle arena, side quest, and shop, which involve using their weapon and monsters in some way. The Capybara system will randomly spawn a Capybara for the player to capture. Except for the calendar system, which is used to progress the in-game activities, all the systems reward players with the resource used for breeding.

1. Battle Arena

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.

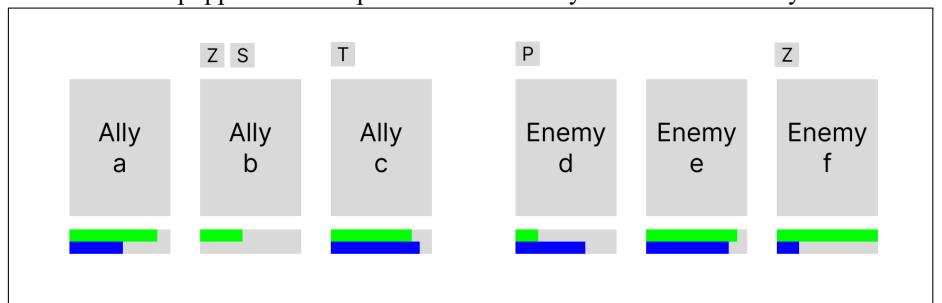


Figure 3.9: Interface Design of the Battle in the Arena

In Figure 3.9, the players' monster team is on the left and the enemy team is on the right side of the scene. 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.

2. Side Quest

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 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. The designed interface of the side quest system is in Figure 3.10 and 3.11.

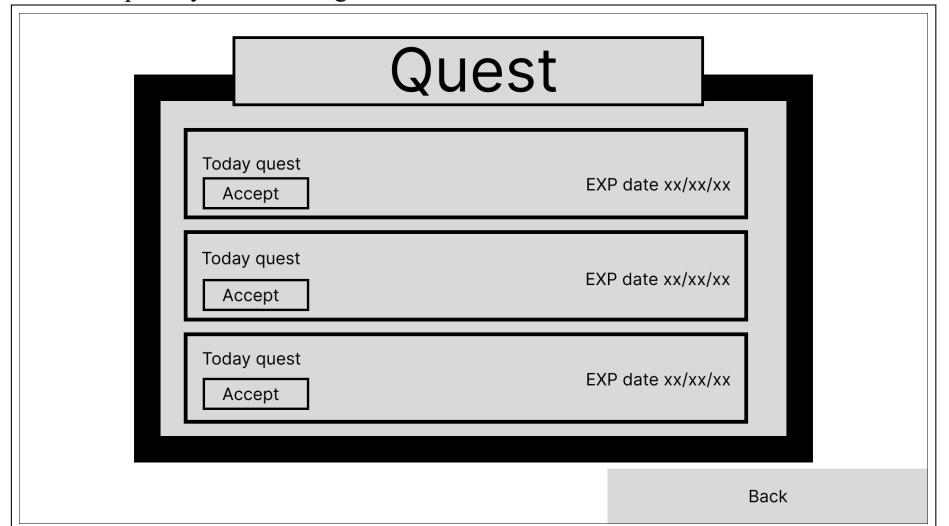


Figure 3.10: Interface of Questboard for Accept Quest Scene

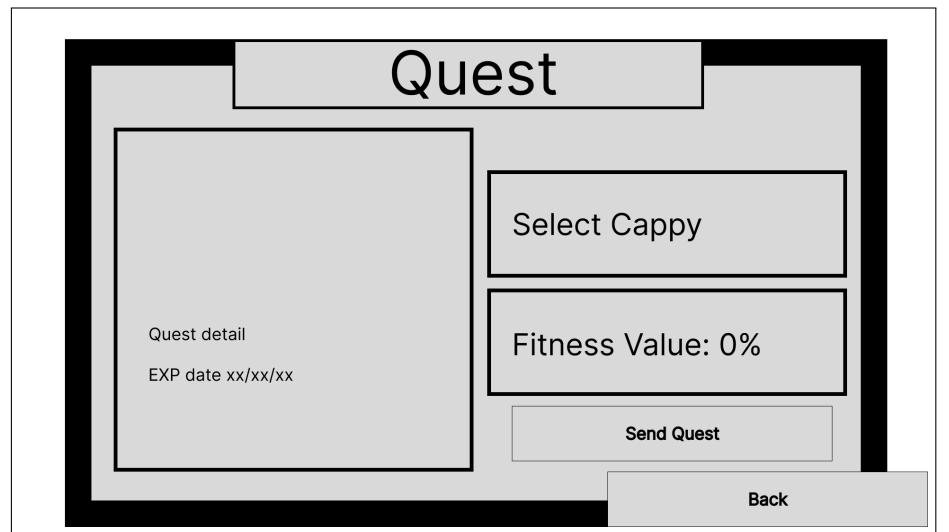


Figure 3.11: Interface of Quest Submission Scene

In Figure 3.10, the players can choose the quest they want to accept. The quest board will provide details of the quest including the appearance of the requested monster and time limit of the quest. If the players accept and in Figure 3.11, the players can submit their monster to complete the quest.

3. Shop

After submitting quests, the population of monsters 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. The example of the designed interface of the shop page is shown as Figure 3.12.

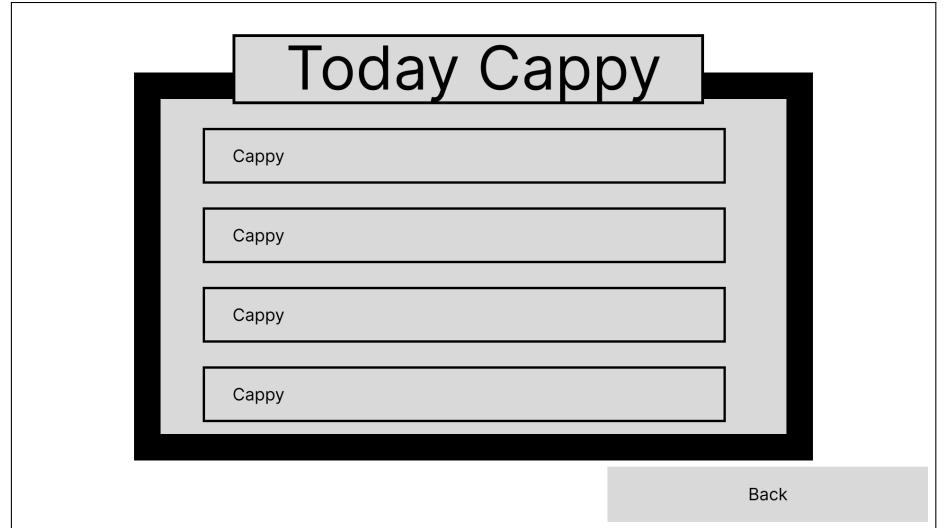


Figure 3.12: Interface of the Shop Page

4. Capybara

This is a random event that a random species of Capybara walk by the facility on the main page. It's an addition that represents the variation in genetics using a variety in the appearance of each species. Different species hold a different number of clicks required for capture and a different amount of reward. The player can rapidly click on Capybara to capture it, and the player will get a reward upon successful capture.

5. Calendar

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. The designed interface of the calendar system is in Figures 3.13.

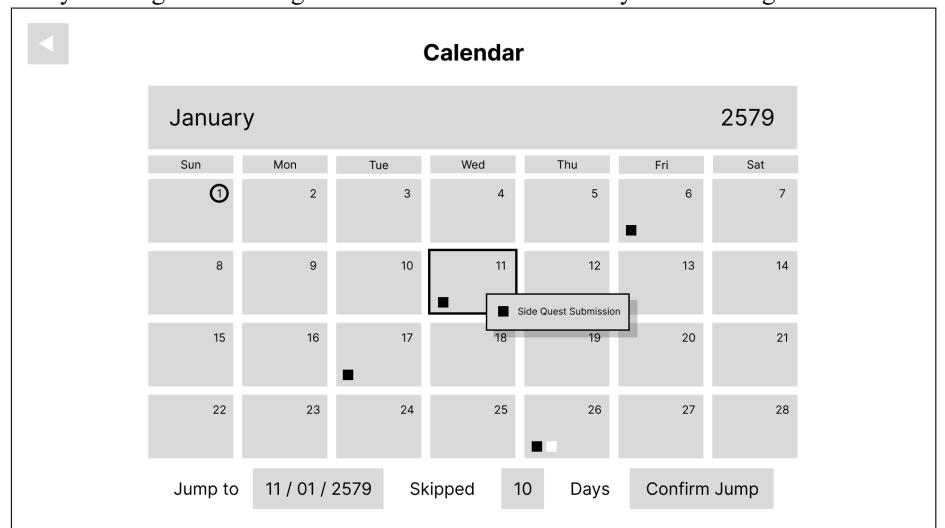


Figure 3.13: Interface of the Calendar System

- 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

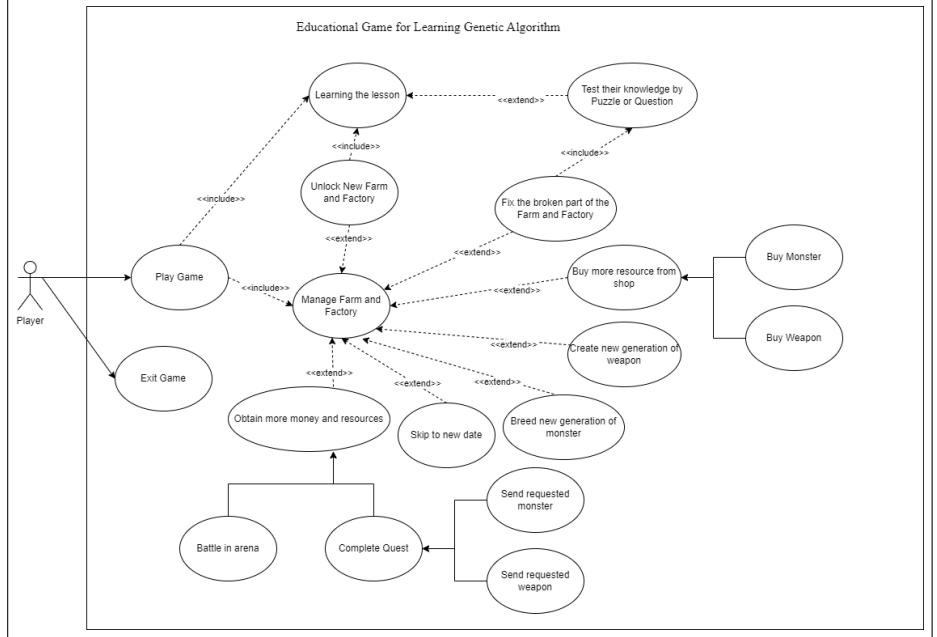


Figure 3.14: Use Cases Diagram

Figure 3.14 shows the actions the player can perform in our game. 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 unlocking new research and facility, breeding weapons and monsters, and purchasing new monsters. Players can earn money by completing main and side quests, winning the battle in the arena, and capturing Capybara. The unlocking fee for new research is set to be very low compared to other situations to prevent the situation that money be an obstacle to learning.

- 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.

3.3 Development

This phase of work aims to develop the learning materials and the learning activities. Such activities in the context of the educational game are precisely the puzzle gameplay that we will use as learning and assessment activities. The following content will describe the learning material, the structure of the puzzle, and prototyping and usability testing.

3.3.1 Learning material

From the expected outcome, we break it down and list the topic of the learning content as we mentioned in the research lab system. For each topic, the image and description text is created. Figure 3.15 is an example of the learning material we created for the single-point crossover. Please note that the material will be adapted to suit the game dialogue later, and all the learning materials are provided in the Appendix.

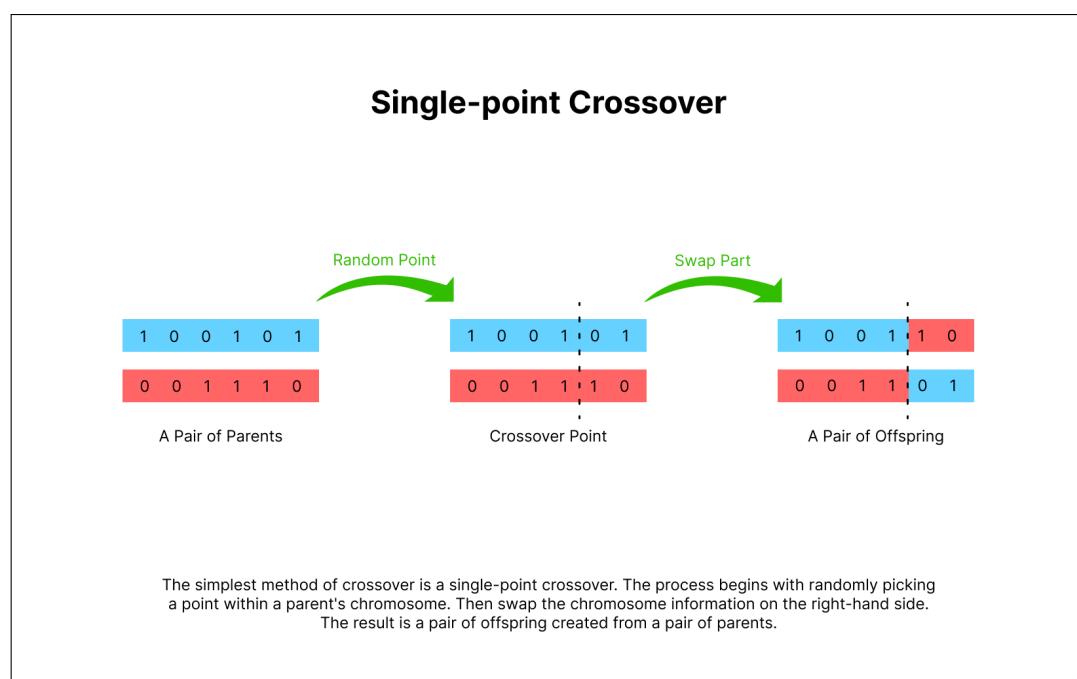


Figure 3.15: Learning Material for Single-point Crossover.

3.3.2 Puzzle structure

The puzzle will be used as an assessment activity for the demonstration and solving level of learning of the criteria in the rubric Table 3.1. These puzzles not only will be used as an assessment but also will be used as an exercise that helps the player maintain and verify their knowledge.

The players will encounter puzzles in 2 different situations. The first situation is right after the players finish learning the lesson from the research lab. The players are required to solve the puzzle to complete the lesson and unlock the new corresponding facility. This will be treated as the first assessment. Another situation is when the farms or factories break down. Players will have to solve the puzzle to fix and continue to use it. This can be interpreted as an exercise and also be calculated as a re-assessment.

The puzzle the player will encounter can be classified into 3 types as follows:

1. Questions

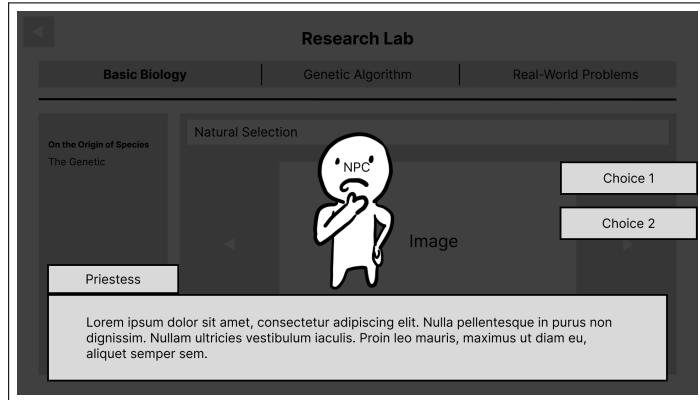


Figure 3.16: The Example of Question Scene

In Figure 3.16, the players will have to choose the answer from the question. This type of test is for testing the players on how they explain, classify, or identify the lessons they have learned.

2. Demonstration Puzzle

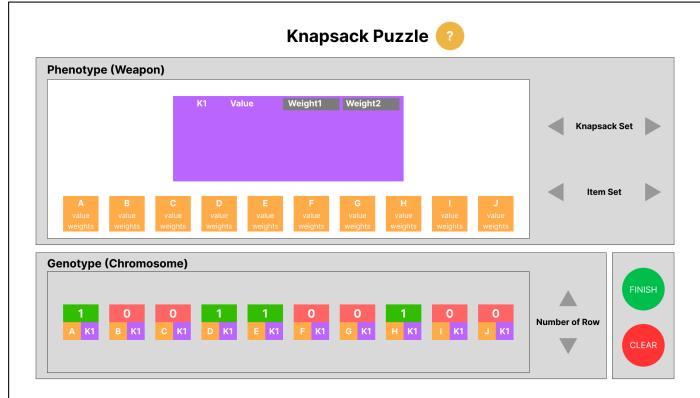


Figure 3.17: The Example of Demonstration Puzzle

This type of puzzle is an open space where the player can operate the objects on the screen to show off their skill. The result of the puzzle will be used as evidence to evaluate the players' learning level for such criteria. The example of a knapsack decoding demonstration puzzle is shown in Figure 3.17, the player is required to drag and drop the item into the knapsack above to make the phenotype match the given genotype below.

3. Problem Solving

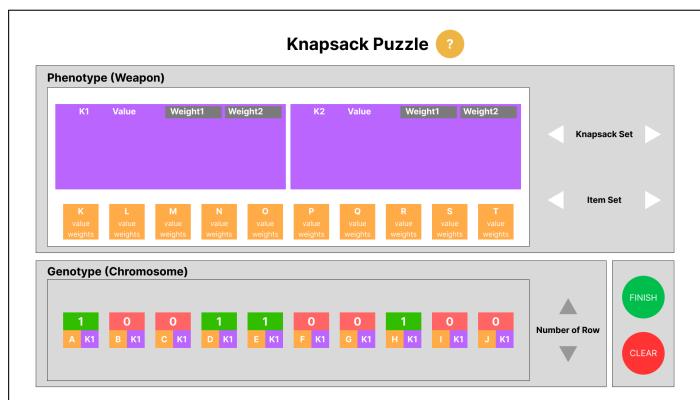


Figure 3.18: The Example of Problem Solving Puzzle

To solve the problem, the learner must choose the method of how they can solve the problem and then perform that method to solve it [30]. This type of puzzle is very similar to the demonstration puzzle with the extra step of choosing the method. The example of a knapsack solving puzzle is shown in Figure 3.18, the extra step of choosing the method is represented as a knapsack and items set selection. The player must click on the button on the right-hand side to choose the correct knapsack and item set. Then perform the drag and drop procedure just like what they must do in the demonstration puzzle to this problem-solving puzzle.

The learning and assessment activity that involves the demonstration and problem-solving level of learning includes the knapsack problem formulation, the parent selection, and the crossover. The detail of the action the player should perform in these puzzles to pass the assessment is described in Table 3.3 below.

Table 3.3: Activity to Achieve the Learning Level of each Criteria

Rubric Criteria (Topic)	Activity to Achieve Demonstration Level	Activity to Achieve Problem-solving Level
Knapsack Problem Formulation	<ul style="list-style-type: none"> - For problem decoding, drag and drop the item(s) into the knapsack(s) to make the phenotype match the given genotype. - For problem encoding, point and click the bitstring to make the genotype matches the given phenotype. 	<ul style="list-style-type: none"> - For problem decoding, choose the correct item(s) and knapsack(s) set. Then perform the action like the demonstration level. - For problem encoding, choose the correct bitstring format. Then perform the action like the demonstration level.
Parent Selection	For the given parent selection method and operation buttons, point and click the operation buttons in the correct order until the number of selected parent meet the number of the current population.	Give the player the description rather than the exact name of the parent selection method with all operation buttons (unnecessary operation buttons included). The player must choose what selection method the description means and perform the action using only the necessary operation.
Crossover	<ul style="list-style-type: none"> For the given crossover type and the set of chromosomes, the player must pick any pair of chromosomes with the same bitstring format. Then create a random crossover point(s) to suit said crossover type, and drag and drop the proper section of chromosome to complete the crossover. 	For the given set of parent chromosomes and the wanted child chromosome, the player must pick the proper pair of parents. Then select the necessary crossover point(s), and drag and drop the proper section of the chromosome to make them match the wanted child chromosome.

From the described in-game assessment activities we will use in the game, the assessment result will be kept in the Hall of Intelligence using various types of qualification pieces. Such pieces include copper, silver, and gold. The assessment of activity for the criteria in the rubric Table 3.1 is outlined in Table ?? below. The learning performance involving explaining, classifying, and identifying is highlighted and represented in-game using a copper piece. The demonstration level is highlighted and represented in-game using a silver piece. And the gold piece is used for the solving level.

Table 3.4: Assessment of the Activity for Criteria

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, Multidimensional and Multiple Knapsack Problem.	**Able to demonstrate the chromosome encoding and decoding of the Standard and Multidimensional 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 and Multidimensional Knapsack Problem.	Able to solve the chromosome encoding and decoding of a Multiple Knapsack Problem.
The learner is able to solve the parent chromosome selection problem in the Genetic Algorithms.	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.
The learner is able to solve the chromosome crossover problem in the Genetic Algorithms.	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 techniques including Elitism, Bit Flip Mutation, Flip Bit Mutation, and Boundary Mutation.		

Note: *Orange highlighted = Copper Jigsaw piece, **Gray highlighted = Silver Jigsaw piece, ***Yellow highlighted = Gold Jigsaw Piece

3.3.3 Prototyping and usability testing

For typical education design using the ADDIE model, the developed contents may be used in the education simulation to test the feasibility of the project. As an educational game, we decide to conduct usability testing instead since it's more suitable for the game and we can't directly perform the learning simulation.

Since we haven't developed the actual working software yet in this phase, we choose to develop the simple wireframe prototype, a series of mock-up screens simulating the working software, for testing purposes. Then we record the prototype as a video together with the narration of how things work.

After the prototyping is done, we decide to conduct the test on a small group of the target user which is the student that currently studies in the field of computer engineering. The data we want from this test is mainly opinions about our current design. So, we sent the video of the prototype to the tester and request them to answer the post-test question about their opinion on the design. The six questions in total with the possible answer will be shown in Table 3.5.

Table 3.5: The Questionnaire for Usability Testing

Question	Possible Answer
Have you ever studied Genetic Algorithms?	Yes/No
What do you think about this design?	Score from 1 (Terrible) to 5 (Excellent)
What do you think about the user interface?	Score from 1 (Hard to Use) to 5 (Easy to Use)
What do you think about the complexity of the gameplay?	Score from 1 (Too Complicated) to 5 (Too Simple)
If you notice any issue that makes this game terrible, what is it?	Short text
If you could change one thing in this game, what would it be and why?	Short text

CHAPTER 4 IMPLEMENTATION RESULTS

In this chapter, according to the ADDIE model, we will introduce our work in the implementation phase and evaluation phase of our project. This includes the result of usability testing, the software implementation result, and the user satisfaction evaluation result.

4.1 Usability Testing Result

We conduct usability testing using the information we already mentioned in the development phase in chapter 3. The test takes place in the first week of February 2023, which is in the first sprint of semester 2. The summary of the responses from 13 testers will be described.

Table 4.1: The Summary Response to Usability Questionnaire

Question	Response
Have you ever studied Genetic Algorithms?	8 Yes, 5 No
What do you think about this design?	An average score is equal to 4.0 (Close to excellent)
What do you think about the user interface?	An average score is equal to 4.2 (Close to easy to use)
What do you think about the complexity of the gameplay?	An average score is equal to 2.2 (Close to too complicated)
If you notice any issue that makes this game terrible, what is it?	Optional short text answer
If you could change one thing in this game, what would it be and why?	Optional short text answer

From the responses in Table 4.1 above, the overall design and the user interface have no serious problems. For the complexity of the game, the tester's overall opinion indicates that they think the gameplay is a bit too complicated. However, we think such an opinion is acceptable because our game is an educational game with puzzle gameplay which may require a portion of learning before clearly understanding the puzzle. The other 2 question is optional short-answer questions which turn out that there is some improvement suggestion and question about some specific game mechanic but still have no severe problems. So, we continue the implementation with no major change in the design. Please note that the full version of the response can be reviewed in Appendix.

4.2 Implementation Result

After finishing all nine sprints, we can deploy a playable prototype on Windows. However, there are still known and unknown bugs that we are still searching for and fixing. We collect feedback on the game by both testing the game ourselves and inviting people from the target group to play.

The following section will feature game interfaces and their functionalities.

1. Title



Figure 4.1: Title Screenshot

This scene will display every time players enter the game. Its main features include starting new game, continue from old save (if possible), adjust volume in settings, and exit game.

2. Hub

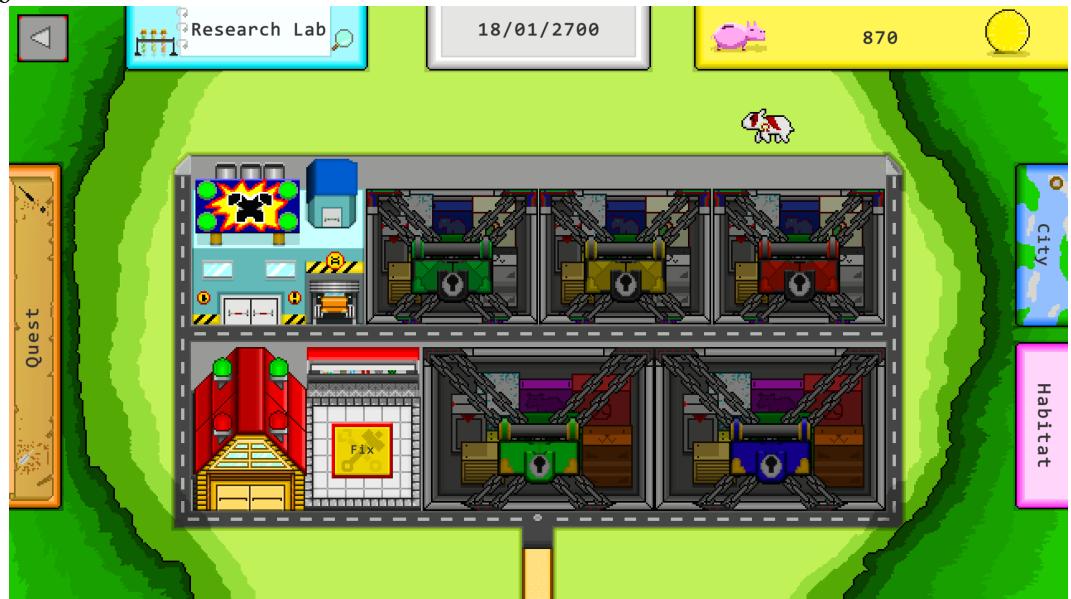


Figure 4.2: Main Hub Screenshot (Overview)



Figure 4.3: Main Hub Screenshot (Fixing facilities)



Figure 4.4: Main Hub Screenshot (Quest Overlay)

This scene serves as a general hub for players to navigate through most parts of the game. In the middle, there are factory (upper row) and farm (lower row) buttons. Apart from taking players to the farm and factory scene, they also indicate the lock/unlock status of the facility and its condition, which players can click on it to fix that facility. The quest tab on the left will display the current main quest and accepted side quests. The habitat, the city, and the research lab will take players to their respective scenes. The calendar tab on the top middle will take players to the calendar scene while displaying the current date in the game. The upper left tab shows the current amount of money players possess. Apart from navigation, there are also chances for a random capybara to spawn, which players could click to pet them, and they might give some rewards.

3. Farm

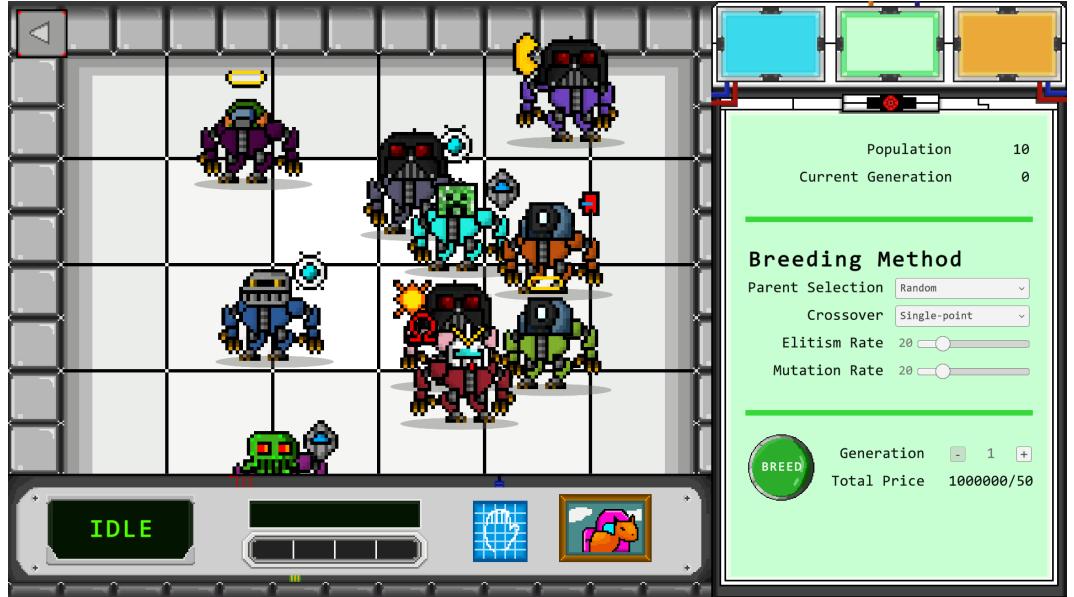


Figure 4.5: Farm Screenshot

Players can breed mechs on the farm to improve their quality or make them fit with their preferences within this scene. On the left side is a space for mechs to walk around, and the panel below displays the status of the farm if it is currently idle, breeding, or broken. The main functionalities of this scene are on the right side, which consists of three panels. The first panel lets players edit the fitness function of this farm so they can prioritize what mechs they want. The second one is for breeding configuration. Players can choose means of parent selection, crossing over, and other relevant variables as they want. The last panel displays the list of the current population on this farm. Players can click to view more details about the mech.

4. Factory



Figure 4.6: Factory Screenshot

Similar to farms, factories let players manage their weapons. It has a similar layout to the farm, with chromosome visualizations on the left, control panels on the right, and a status panel at the bottom.

The only difference is that the first panel is a weapon description instead of fitness adjustments, as the fitness function of the weapon is controlled by the factory.

5. Habitat

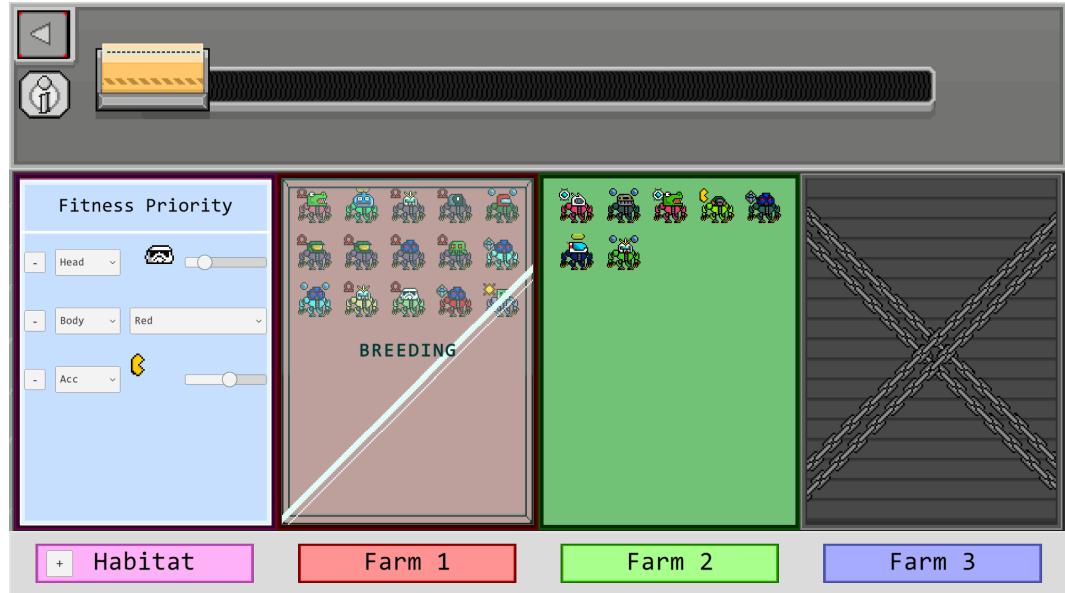


Figure 4.7: Habitat Screenshot

The mechs can be assigned to other facilities (if available) within this scene. The order of the mechs in each farm is set by separated fitness priority, which can be edited by players.

6. Research Lab

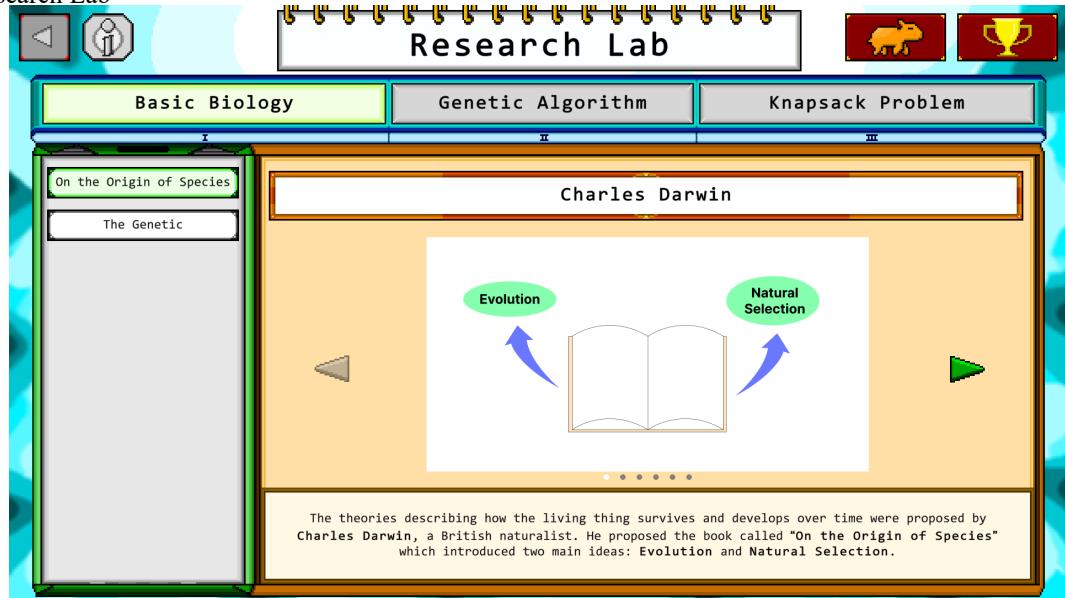


Figure 4.8: Research Lab Screenshot

Players can unlock and review learned lessons in this scene. After unlocking a new lesson, a cutscene will play. Furthermore, the Upper left buttons will navigate players to the Hall of Capybara and Hall of Intelligence respectively.

7. Cutscene

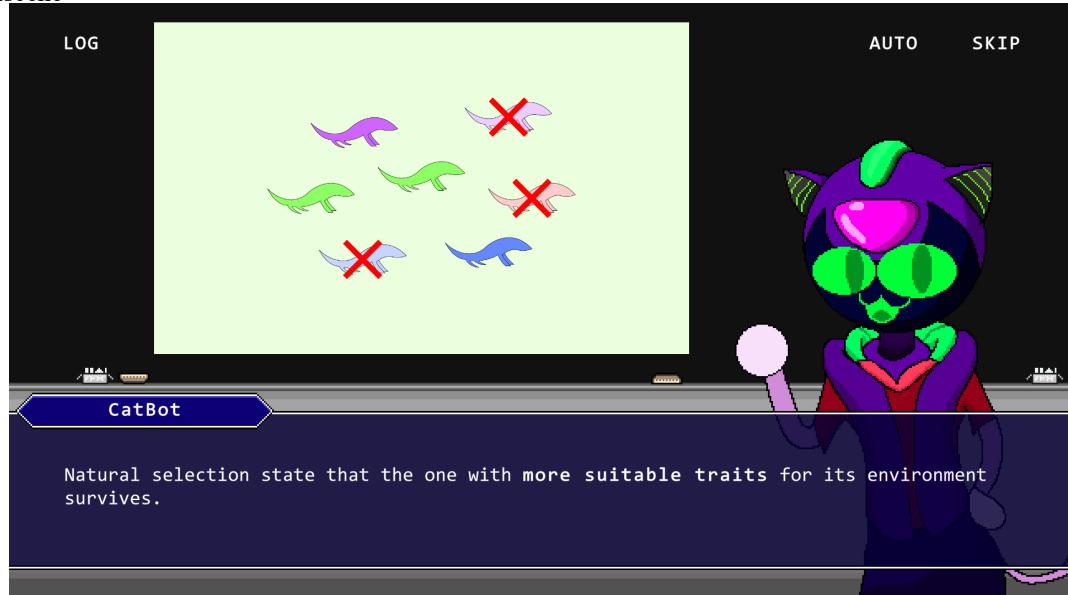


Figure 4.9: Cutscene Screenshot (Dialogue)

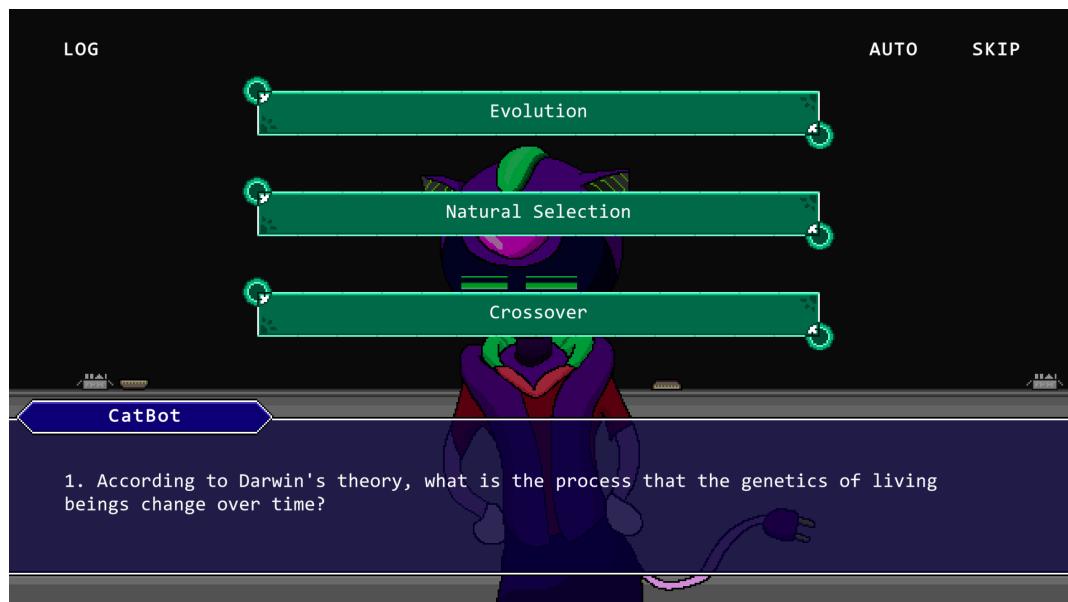


Figure 4.10: Cutscene Screenshot (Choice)

Cutscenes can display stories, lessons, and exams in a Visual Novel style. Players can inspect previous dialogues with the Log button, make the next dialogue display automatically with the Auto button, and skip the whole conversation with the Skip button if it is skippable.

8. Hall of Intelligence



Figure 4.11: Hall of Intelligence Screenshot

This scene shows sets of jigsaw pieces representing categories of achievements players received in the game. When clicking on it, players can take a test to increase their rank. It also shows players' names, the number of passed days, the current main quest, and the number of times players have succeeded or failed each level of puzzles.

9. Puzzles

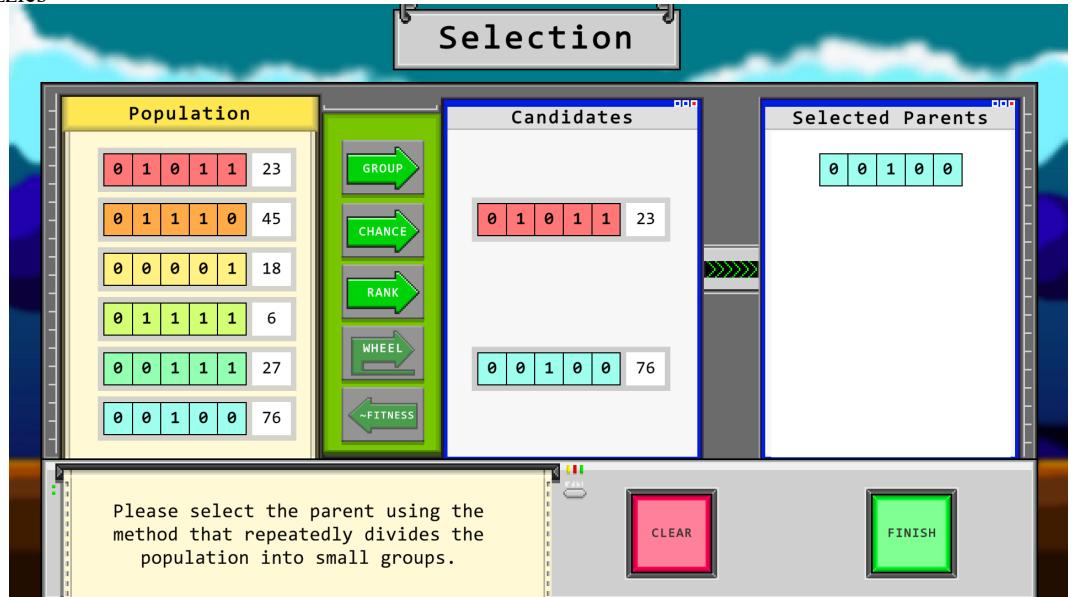


Figure 4.12: Puzzle Screenshot (Selection)

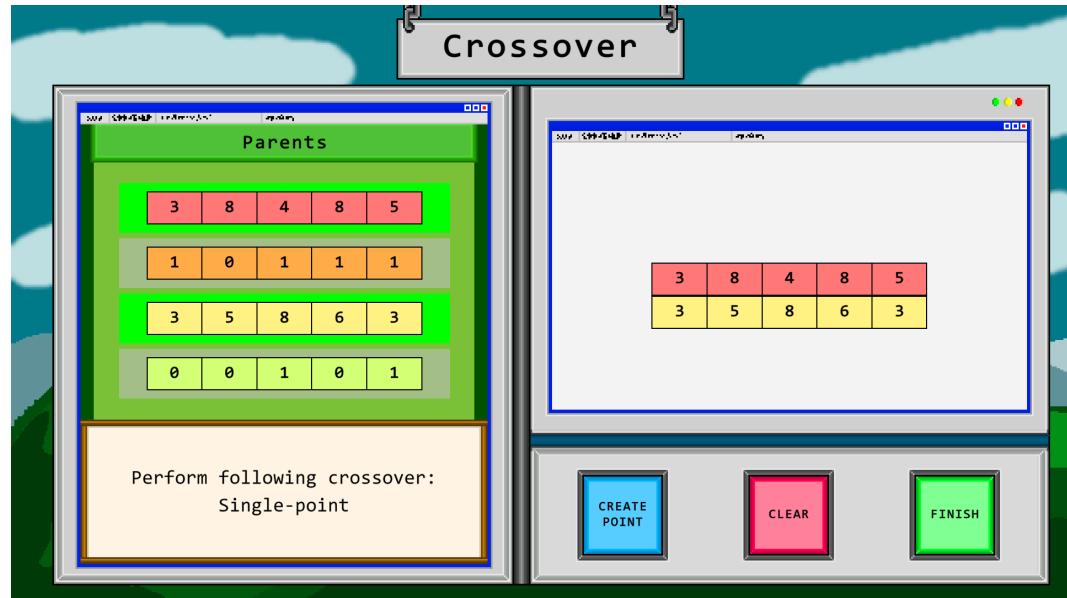


Figure 4.13: Puzzle Screenshot (Crossover)

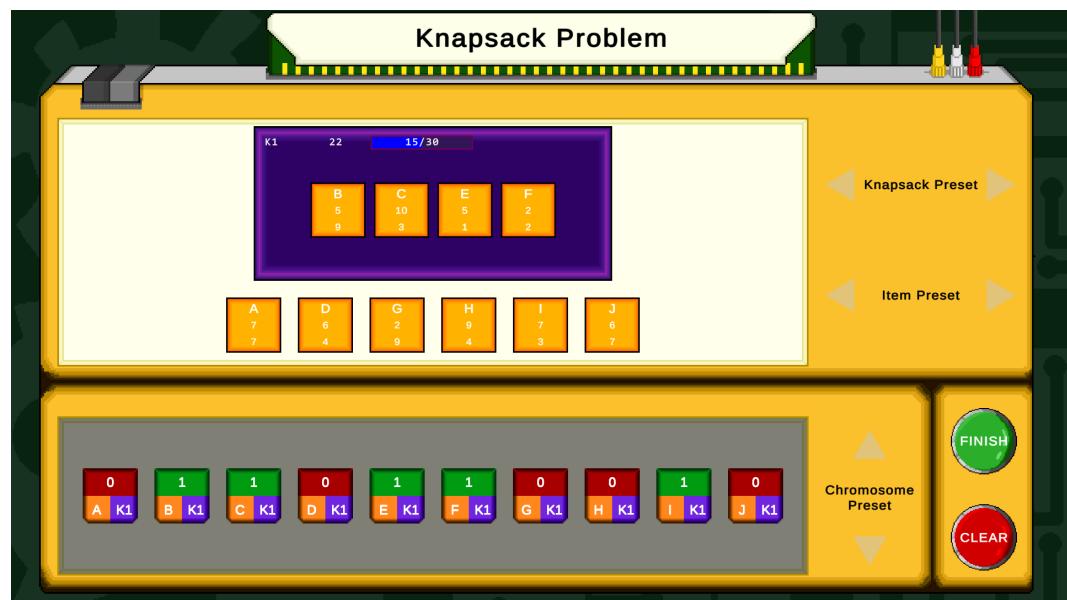


Figure 4.14: Puzzle Screenshot (Knapsack)

Players will solve a puzzle when taking tests to increase a jigsaw rank or fix a facility. There are multiple types of puzzles depending on types of knowledge; knapsack puzzles and genetic algorithm-related puzzles, and level of assessment, such as question, demonstration, and solving.

10. Hall of Capybara



Figure 4.15: Hall of Capybara Screenshot

This scene shows which capybara players have seen and how often they have greeted them.

11. Calendar

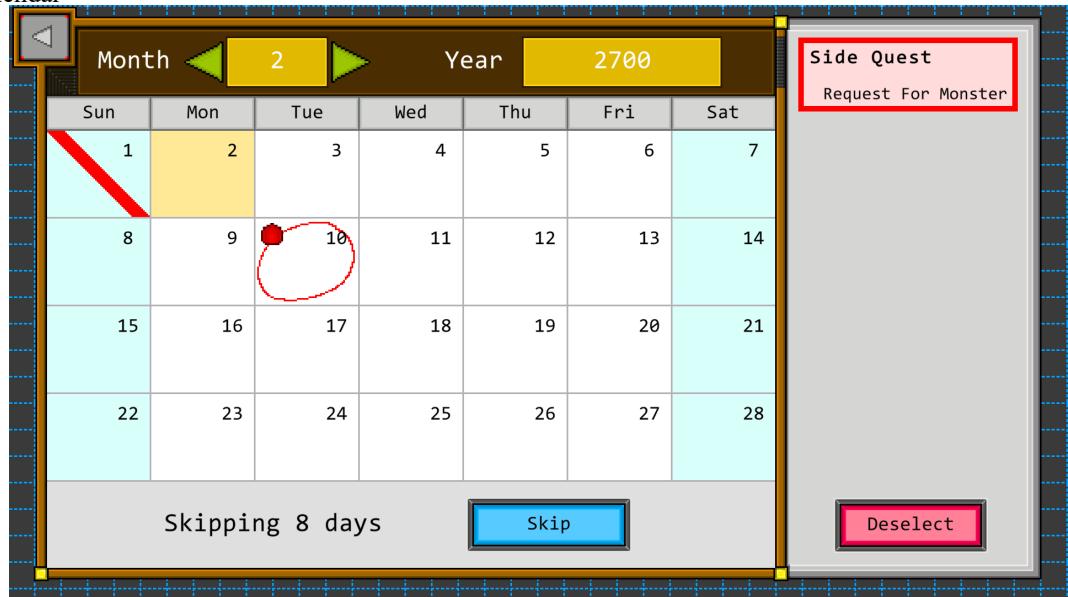


Figure 4.16: Calendar Screenshot

The calendar shows the current date, or the selected date events such as side quest submission deadline. Players can either end their day or skip to the selected date.

12. City

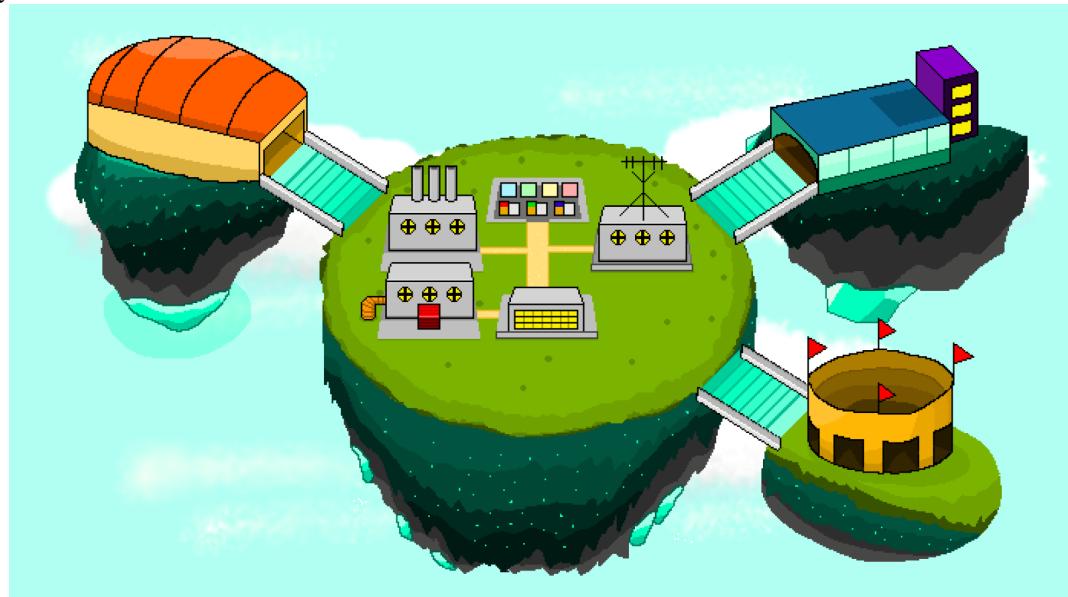


Figure 4.17: City Screenshot

This scene serves as a secondary hub. It takes players to other scenes: the quest, the arena, the shop, and the main hub.

13. Questboard

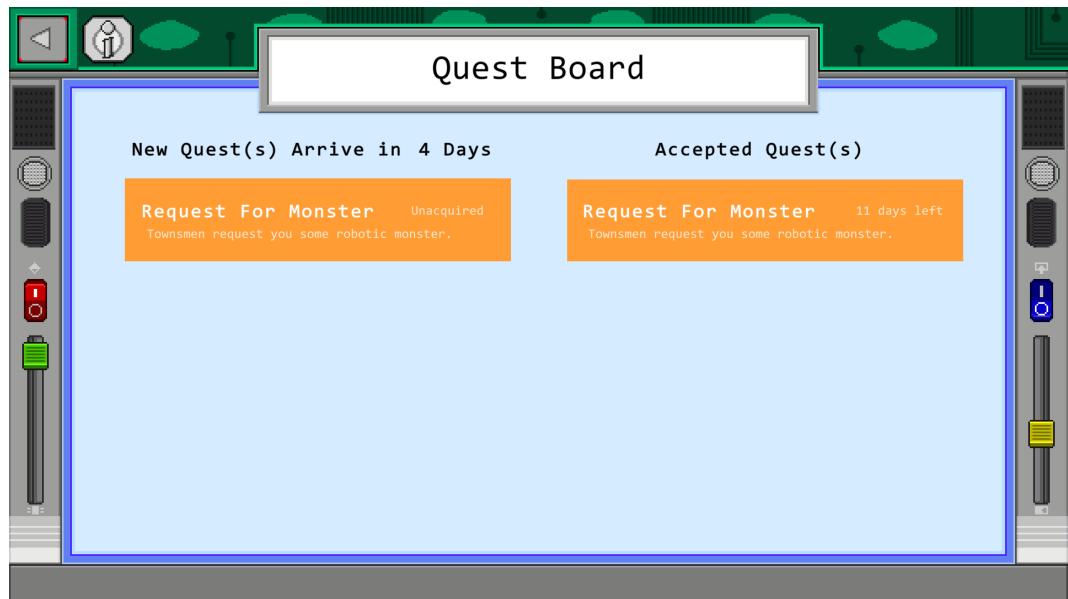


Figure 4.18: Questboard Screenshot (Overview)

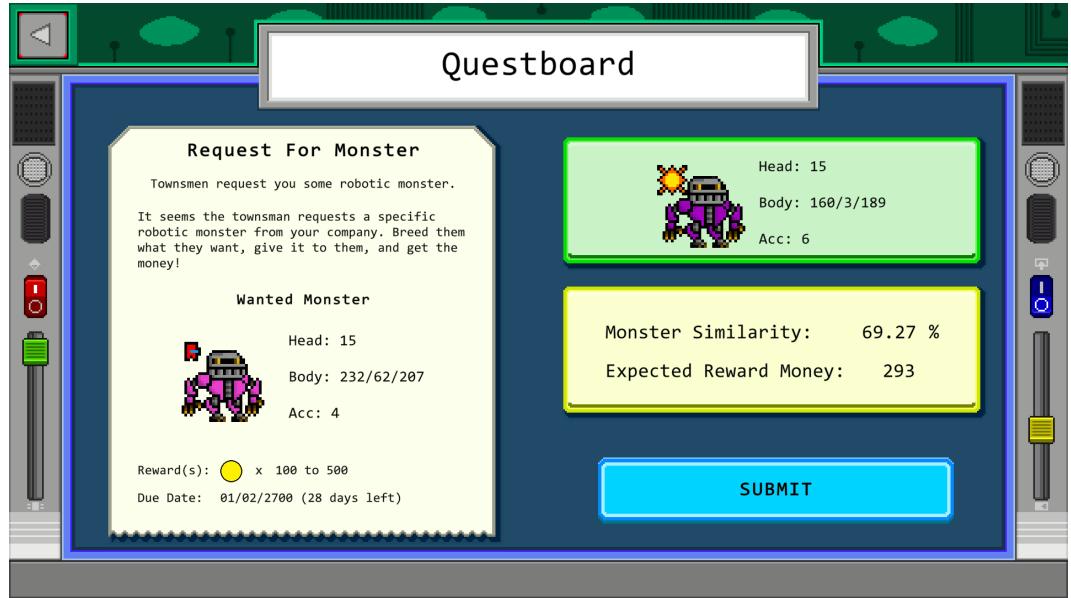


Figure 4.19: Questboard Screenshot (Detail)

This page helps players manage their side quests. They can accept available quests on the left side, and they will appear on the right side to be submitted. When players submit a quest, they will receive rewards according to the similarity between the requested mech and the submitted mech. If players fail to submit a quest before its deadline, players will have to abandon that quest and pay a penalty.

14. Shop



Figure 4.20: Shop Screenshot

Players can replenish their mech population by purchasing new mechs in this scene. They can inspect its information by hovering over it. After a mech is bought, it will be out of stock until the shop is restocked.

15. Arena



Figure 4.21: Arena Screenshot (Preparation)

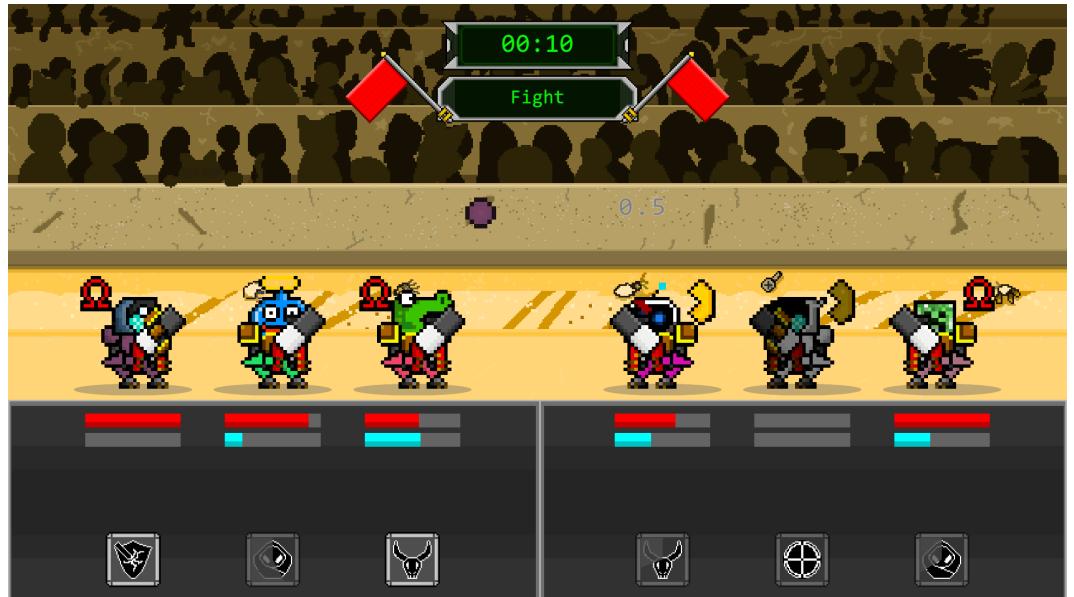


Figure 4.22: Arena Screenshot (Battle)

The arena consists of two phases, the preparation, and the battle. Players can select which mech and weapon to put in the team by themselves or set it automatically. The battle record of the day and the element multiplier are displayed on the top. After selecting an opponent, the battle sequence will be initiated. In the end, players will receive a reward or penalty depending on the battle result.

4.3 User Satisfaction Evaluation Result

We conducted user satisfaction testing with the completed version of the game using a survey form. Determining the satisfaction level of each system in the game from the user's playthrough and the knowledge of the Genetic Algorithm gained.

The test took place from May 10th to May 17th, 2023, and the survey was distributed in a Google Form. The form included a link to the itch.io page of the game, where players could choose to play the game in their

preferred browser or download it to play locally on their desktop.

In the form, testers were required to play the game and achieve at least a silver jigsaw in each of the trays in the Hall of Intelligence. They were asked to upload a screenshot of their Hall of Intelligence, as shown in Figure 4.23 to provide their progress in the game.

The results provided a comprehensive overview of user perceptions and highlighted opportunities for improvement. The summary of these responses from 92 testers will be described in the following sections.

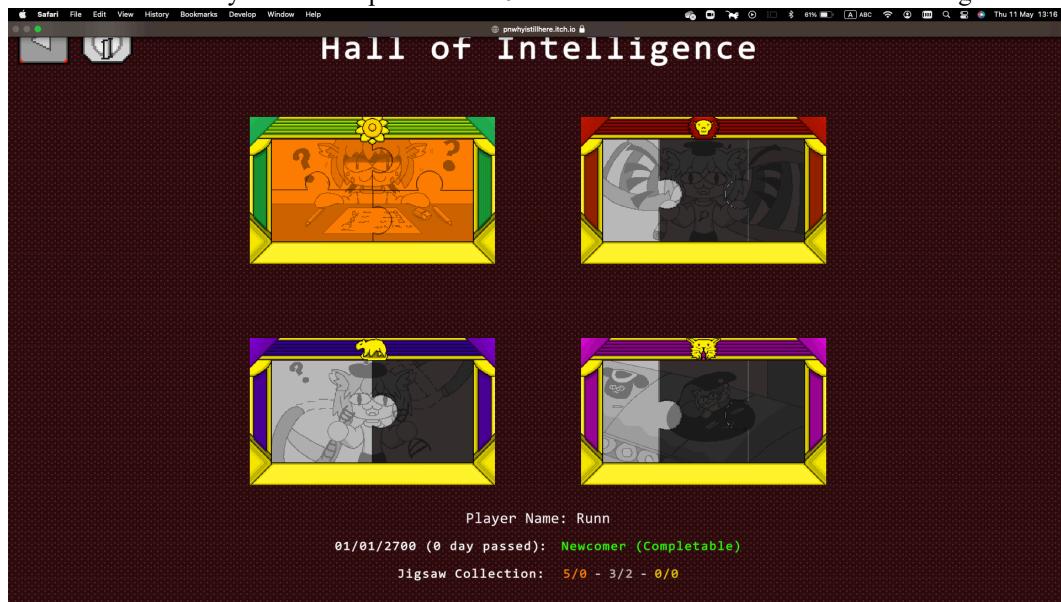


Figure 4.23: Example of Screenshot of the Hall Of Intelligence from the tester

The results of the survey provide a comprehensive overview of user perceptions and highlight opportunities for improvement. The summary of the responses from 30 testers will be described.

- Summary Of Tester Data

There were 92 participants who responded to the survey form, all of whom were university students. Out of the participants, 83 were freshmen or first-year students, 8 were seniors or final-year students, and there was one sophomore or second-year student. Among the participants, 88 were studying computer engineering, while 4 participants were studying in each of the following fields: computer and information technology, computer science, media technology game development, Institute of field robotics (FIBO), and business administration.

- Overall Satisfaction Result

The overall results of the satisfaction levels for each system in the game are shown in the Table 4.2 below.

Table 4.2: The Summary Response of User Satisfaction Survey to the Overall System of the Game

Question	Response
What your satisfaction level of the learning materials and teaching cutscene in Research Lab?	From the score of 1 (Very hard to understand) to 5 (Very easy to understand) An average score is equal to 3.91 Mode of the score is 4 (43 answers)

Table 4.2: The Summary Response of User Satisfaction Survey to the Overall System of the Game

Question	Response
What your overall satisfaction level of all the Puzzles used in the game?	From the score of 1 (Very Dissatisfied) to 5 (Very Satisfied) An average score is equal to 4.01 Mode of the score is 4 (36 answers)
What your overall satisfaction level for the Research Lab and Hall of Intelligence System?	From the score of 1 (Very Dissatisfied) to 5 (Very Satisfied) An average score is equal to 3.92 Mode of the score is 4 (40 answers)
What your overall satisfaction level for the Arena System?	From the score of 1 (Very Dissatisfied) to 5 (Very Satisfied) An average score is equal to 3.89 Mode of the score is 4 (29 answers)
What your satisfaction level of the Farm & Factory Management interface, music and sound effects?	From the score of 1 (Very Dissatisfied) to 5 (Very Satisfied) An average score is equal to 3.84 Mode of the score is 4 (35 answers)
What your satisfaction level of the Overall Art Design interface?	From the score of 1 (Ugly) to 5 (Very Beautiful) An average score is equal to 4.09 Mode of the score is 4 (43 answers)
Overall, is the game fun to play?	From the score of 1 (No, it is very boring) to 5 (Very fun) An average score is equal to 3.74 Mode of the score is 4 (47 answers)

From the responses in Table 4.2 above, the satisfaction level of the overall system of the game is the result at around 4th level or ‘Satisfied’ with the design and implementation. However, at the beginning of the response receiving period, many people claimed that the arena was unplayable. When we investigated the reasons, we found some bugs preventing some players from getting the monster and entering the arena. This bug existed only in the browser version of the game. We fixed the bug and replaced the browser version of the game on the itch.io page.

- Educational Result

In the survey, we asked the testers about their level of knowledge about Genetic Algorithms before and after playing the game to determine the success of the teaching from the game. The results of the questionnaire are shown in Figure 4.24, which is before playing the game, and Figure 4.25, which is after playing the game.

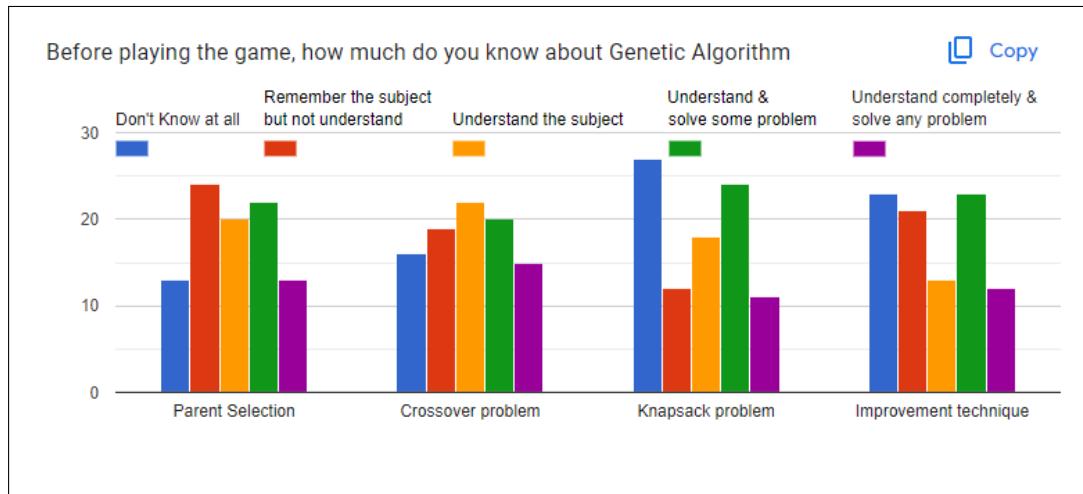


Figure 4.24: Level of knowledge about topics in Genetic Algorithm before playing the game

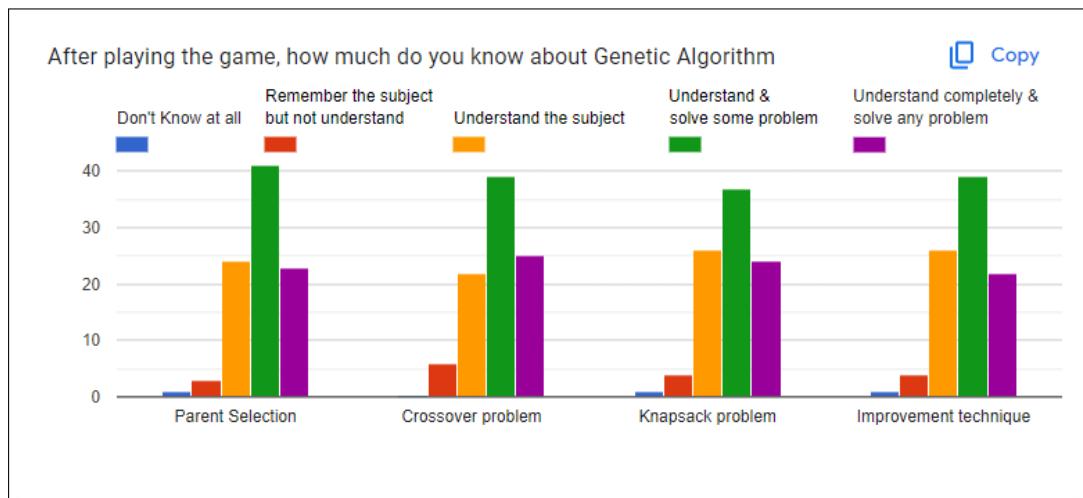


Figure 4.25: Level of knowledge about topics in Genetic Algorithm after playing the game

Based on Figure 4.24 and Figure 4.25, we have determined that the game effectively enhances players' knowledge about various topics related to the Genetic Algorithm. These topics align with the assessment criteria outlined in Table 3.1: Rubric for Assessing the Learner from Chapter 3.

- Suggestions and Problem Encounter in the game

In the survey, we open for the testers to give suggestions to help us improve the game in the future and to identify the problem that they might encounter when testing the game. From the given suggestions, we summarize all the main points that the tester wants us to change or add to make the game better as follows.

1. Add a setting feature to the game to configure sound and music volume and change the game resolution.
2. Add more animation to the cutscene of the game because the game uses cutscene to teach player for the first time, but there are not a lot of animation to make the game more engaging.
3. Add more detail to the tutorial and teach players about the systems in the game step by step because there are a lot of systems in the game and it might be too overwhelming for the new player.

4. Revise multiple interfaces as they were claimed to be confusing and difficult to use, such as ambiguous instructions and navigations, a misleading trophy icon, using jigsaws as buttons, lack of confirmation prompt, etc.

In the survey, the testers reported bugs and errors that only occur when playing in the browser version of the game. The bugs most likely occur from extending the platform that the game was also built using Web Graphics Library(WebGL) instead of only Windows desktop in order to make it more convenient for the testers to play without downloading the game files. The Fatal bugs that occurred in the early version of the game that stop the players from continuing to play the game were fixed and deployed in the later version of the browser version of the game when we were aware of the bugs.

CHAPTER 5 CONCLUSIONS

5.1 Work Process

We use Scrum as a main process because of its high agility, allowing us continually adjust the work plan to suit the current progress and expected result. As an educational game, we also adopt ADDIE and OBE frameworks to ensure that the educational outcomes can be completed and measurable.

We begin the operation by designating the backlog of the project. Then perform ten 3-week sprints to complete all the planned backlog. The first four sprints focus on the design and take place in the first semester. The later six sprints focus on implementation and take place in the second semester. Although we initially planned to perform nine sprints, we had to conduct the tenth sprint to finish the leftover backlog due to the vast fraction of misestimated work.

5.2 Problems and Solution

During the operation, we encountered and solved several problems. Three main problems that cause a high impact on the project are the educational game development framework, workload management, and technical problem. All these problems will be described in detail as follows.

1. 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.

2. Workload Management Problem

During the operation in the first semester, there were several changes in the scope of the project. We handled the changes at the very beginning of the project by revising the project proposal due to the increment of the project members. We also had to adjust and perform a vast fraction of redesigning after that because of developing framework changes. We solve these problems by continually monitoring and adjusting the remaining backlog to suit the limited time, redesigning only the work in the necessary part, and keeping others the same to reduce the extra work as much as possible.

During the development in the second semester, we found several misestimated workloads, including documentation, knapsack puzzle, art asset, and saving systems. Since we refactored the working software along with the development, we decided to decrease the workload on quality assurance, reallocating the initial workload for those misestimated topics, and speed up the pacing of software development.

After the last planned sprint, which is the ninth sprint end, there still has a remaining backlog and time. So we conduct the extra sprint to clear all the leftover work. Figure ?? shows the project backlog with an extra sprint. We estimate that all the work is done on time because there is only a tiny amount of

document work left.

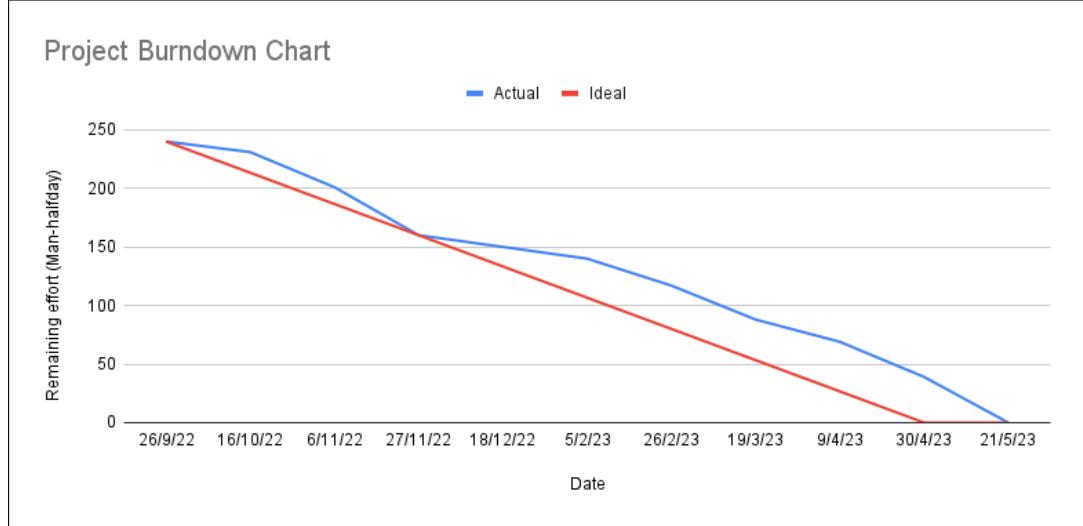


Figure 5.1: Project Burndown Chart

3. Technical Problem

As the code had to be written by multiple people, we set some coding standards at the beginning of the development phase, such as naming git branches and variables. Yet, the codes were still hard to understand for various reasons, such as lack of comments, unable to follow the coding standards, or messy logic. This issue led to various major problems. After facing some troubles, we decided to revise the standards and remind each other to follow them.

At the beginning of the development process, we decided to utilize scriptable objects for data saving in the game. However, we later discovered that certain related functions would not function properly in the final build. Thus, we had to modify the saving system by selecting the data to save and writing it into JSON files instead. Unfortunately, due to misunderstandings in the code structures, the adjusted saving system proved ineffective in preserving essential data across gaming sessions. This resulted in various gameplay bugs, such as problems with puzzle generation, an inability to progress to the next day, and occasional game crashes. Upon receiving feedback, each function owner compiled lists of the necessary information to be included in the saving process before incorporating them into the saving function.

At the end of the development, we decided to publish the game on the WebGL platform alongside the Windows platform, as we thought it would be easier for test players to play the game online without downloading. Despite successfully compiling and building the game, it was still full of bugs, as WebGL was not initially the releasing target platform, and there was not enough time to test. One of the major bugs was due to the fact that WebGL did not support some functions in the code, which was never a problem in the Windows platform. Therefore, we had to remove those problematic keywords and rewrite the affected functions.

There were multiple minor bugs. For instance, some codes were written in shallow copy manners, fixed bugs reappeared after merging branches, malfunctioning systems, faulty display on different screen resolutions, and many more. With the feedback of playtesters, some of these bugs were found and successfully fixed. Nevertheless, there were still more hidden bugs to be addressed and fixed.

5.3 Future Works

Despite completing the game, there are still several issues that require attention in the future. These issues have been identified based on valuable feedback from playtesters and a list of planned features that were not implemented in the final build. They can be divided into three categories: fixing bugs, adjusting usability, and improving the quality of life.

First, we want to fix the remaining bugs in the game before going any further. As for known bugs, we have resolved some of them according to the previous section, such as modifying the saving function and fixing unsolvable puzzle generation. However, many of them still persist. The most significant one is that the game is unsupported on other resolutions than Full HD. Apart from that, the fatal bugs are yet to be found, and the remaining bugs seem to be minor. We will try to fix them and search for more later.

Our next step is to improve the usability of the game. As per various feedbacks, many game elements are too complicated, and the complications come from different sources, such as the interface, the tutorial, or even the system itself. The scene navigation in the game and the layout of the buttons are confusing and sometimes misleading. For instance, the button accessing the Hall of Intelligence, a trophy icon, makes people think it was an achievement page. Some said that many puzzle instructions are too vague and hard to understand, and some tutorials are too ambiguous, so they suggested adding animation to those guides. We will try to fix these issues by revising the interface design and adding interactive tutorials to make it more user-friendly.

To enhance the quality of life, we are planning to add multiple things to the game and modify some features to balance them. To accomplish this goal, several system adjustments will be implemented. For instance, add side quests configuration of requesting multiple monsters at once, add a sell option to the shop (so players could exchange some of their monsters for money), add a skip button to the arena, etc. Many nice-to-have features did not make it to the main game due to insufficient time; for example, an achievement system to keep track of players in other aspects apart from educational topics and more mini-games to keep players' interest. However, with the early feedback from the tester, some of these features, such as volume adjusters, were added already.

Besides technical systems, we also want to add and improve the aesthetics of the game, such as proofreading and adjusting texts in the game, adding more NPC interactions, diverse side quest flavor texts, and more storytelling. Most importantly, we want to test and balance the in-game economy, ways to get and use monsters, and the whole arena system, as many elements are worth inspecting. We want to balance the monster's status growth and its relationship with various battle skills, adjust enemy generation to match suitable difficulty for players and change the reward and penalty of battle outcomes so it gives/takes a suitable amount of money.

5.4 Conclusion

Project G is an educational game for learning Genetic Algorithms developed for people interested in computers, especially for university students interested in Genetic Algorithms. To ensure the quality of a game and the measurability of learning outcomes, we adopt several frameworks, including Scrum, ADDIE, and OBE. We create an educational game using an approach that the CAI is designed first, then adding other gameplay to increase the external learning motivation of the player. Operating this way result in a visual novel and puzzle gameplay as a core for learning and assessing, management of the breeding facilities as a game progressor, and extra activities like the battle system to give the player more fun.

Even with the vast fraction of the change that occurred in the designing and development phases, with the extra sprint, we successfully finished the game and evaluated them. The user satisfaction testing shows us good feedback on both the game systems and the educational outcomes. The game on the web browser, which is the extended platform from our scope, shows a fatal flaw that causes the game to be unplayable at some point.

But with informative feedback from the testers, we had already fixed it. And there also are many suggestions on the game that can enhance our gaming experience. Opening the possibility for us to improve the game in the future.

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. Jan L Plass, Bruce D Homer, and Charles K Kinzer, 2015, “Foundations of game-based learning,” **Educational psychologist**, vol. 50, no. 4, pp. 258–283, 2015.
4. Steve Rabin, Ed., 2005, **Introduction to game development**, chapter 2, Hingham, MA : Charles River Media.
5. 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.
6. Chris Sniezak, 2016, “The Eight Tyeps of Fun,” <https://gnomestew.com/the-eight-types-of-fun/>, Last accessed 16 November 2022.
7. Christine Persaud, 2021, “Bloom’s Taxonomy: The Ultimate Guide,” <https://tophat.com/blog/blooms-taxonomy/>, Last accessed 15 November 2022.
8. KMUTT C4ED, “Outcome Based Education,” <https://cilt.wu.ac.th/backEnd/myfile/attUKPSF/OBE%20WorkSheet%20CUPT.pdf>, Last accessed 15 November 2022.
9. Regine Rafer, “CHARACTERISTICS OF GOOD LEARNING OUTCOMES,” https://www.academia.edu/33889461/CHARACTERISTICS_OF_GOOD_LEARNING_OUTCOMES, Last accessed 24 November 2022.
10. ISFET, “What is the ADDIE Model?,” <https://www.isfet.org/pages/addie-model>, Last accessed 15 November 2022.
11. Michigan State University, “PILOT TESTING IN THE SHARED DISCOVERY CURRICULUM,” <https://curriculum.chm.msu.edu/about/sdc-pilots>, Last accessed 25 November 2022.
12. 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.
13. K.F. Man, K.S. Tang, and S. Kwong, 1996, “Genetic algorithms: concepts and applications [in engineering design],” **IEEE Transactions on Industrial Electronics**, vol. 43, no. 5, pp. 519–534, 1996.
14. Nidhi, 2017, “A Comparative Analysis of Genetic Algorithm Selection Techniques,” **International Research Journal of Engineering and Technology (IRJET)**, vol. 4, pp. 2453–2455, 2017.
15. Janne Koljonen and Jarmo T Alander, 2006, “Effects of population size and relative elitism on optimization speed and reliability of genetic algorithms,” in **Proceedings of the ninth Scandinavian conference on artificial intelligence (SCAI 2006)**, 2006, pp. 54–60.
16. Vinicius Fulber Garcia, 2022, “Elitism in Evolutionary Algorithms,” <https://www.baeldung.com/cs/elitism-in-evolutionary-algorithms>, Last accessed 25 November 2022.
17. Soukaina Laabadi, Mohamed Naimi, Hassan El Amri, Boujemâa Achchab, et al., 2018, “The 0/1 multi-dimensional 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.
18. Bhayani Arpit, 2022, “Genetic Algorithm to solve the Knapsack Problem,” <https://arpitbhayani.me/blogs/genetic-knapsack>, Last accessed 6 November 2022.
19. Nick Babich, 2021, “User & Usability Testing Questions: Ultimate Guide,” <https://xd.adobe.com/ideas/process/user-testing/usability-testing-questions-tips-examples/>, Last accessed 24 March 2023.

20. Jakob Nielsen, 2012, “How Many Test Users in a Usability Study?,” <https://www.nngroup.com/articles/how-many-test-users/#:~:text=For%20really%20low%2Doverhead%20projects,5%20users%20per%20usability%20test.>, Last accessed 24 March 2023.
21. Unity, 2021, “2021 GAMING REPORT,” https://images.response.unity3d.com/Web/Unity/%7B4eb56531-e6aa-492f-8fda-c68ae20af950%7D_2021_Gaming_Report_-_Operate_Solutions.pdf, Last accessed 25 November 2022.
22. Unity, 2022, “2022 GAMING REPORT,” https://images.response.unity3d.com/Web/Unity/%7B10460b81-b6e7-4784-a735-e7347afdf06e%7D_Unity-Gaming-Report-2022.pdf, Last accessed 25 November 2022.
23. Perforce, 2022, “2022 GAMING DEVELOPMENT TRENDS & FORECAST,” https://mma.prnewswire.com/media/1880817/Game_Development_Trends.pdf, Last accessed 25 November 2022.
24. Ben Tristem, 2022, “Unity vs. Unreal: Which Game Engine is Best For You?,” <https://blog.udemy.com/unity-vs-unreal-which-game-engine-is-best-for-you/>, Last accessed 25 November 2022.
25. Daniel Schwarz, 2022, “Figma review,” <https://www.creativebloq.com/reviews/figma>, Last accessed 25 November 2022.
26. Janelynn Camingue, Elin Carstensdottir, and Edward Melcer, 2021, “What is a Visual Novel?,” **Proceedings of the ACM on Human-Computer Interaction**, vol. 5, pp. 1–18, 10 2021.
27. Yingfang Zhang, Zhiyu Shen, and Yifan Zhang, 2019, “Research on the Design of Puzzle Games Based on Unity 3D,” in **2019 IEEE 2nd International Conference on Knowledge Innovation and Invention (ICKII)**, 2019, pp. 145–148.
28. Mary Jo Dondlinger, 2007, “Educational video game design: A review of the literature,” **Journal of applied educational technology**, vol. 4, no. 1, pp. 21–31, 2007.
29. MobyGames, “Omega,” https://www.mobygames.com/game/omega_, Last accessed 15 November 2022.
30. 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.

1 Project - G

Start game
Continue
Exit

2

Evolution
Natural Selection

100 species competing for the living. A species and its lineage over time were presented by Charles Darwin, a British naturalist. He proposed the book called "On the Origin of Species" which introduced him with ideas Evolution and Natural Selection.

Priestess

Unlockable

When the game start, the player will be introduced to the story prologue and setting of this game. The knowledge about Basic Biology including Charles Darwin, Evolution and Natural Selection also be integrated with the prologue.

3

Research Lab

Basic Biology | Genetic Algorithm | Real-World Problems

Natural Selection

The mechanism is a mechanism of natural selection. The propagation of living things can change over time due to environmental factors. This process is called natural selection. It is a process where the fittest individuals are more likely to survive and reproduce, passing on their genetic traits to the next generation.

On the Origin of Species
The Genetic

Round
Yellow

Genotype

Priestess

When the game start, the player will be introduced to the story prologue and setting of this game. The knowledge about Basic Biology including Charles Darwin, Evolution and Natural Selection also be integrated with the prologue.

4

Research Lab

Basic Biology | Genetic Algorithm | Real-World Problems

Phenotype

Round
Yellow

Genotype

Priestess

When the game start, the player will be introduced to the story prologue and setting of this game. The knowledge about Basic Biology including Charles Darwin, Evolution and Natural Selection also be integrated with the prologue.

5

Research Lab

Basic Biology | Genetic Algorithm | Real-World Problems

Natural Selection

The mechanism is a mechanism of natural selection. The propagation of living things can change over time due to environmental factors. This process is called natural selection. It is a process where the fittest individuals are more likely to survive and reproduce, passing on their genetic traits to the next generation.

On the Origin of Species
The Genetic

Test

Image

Priestess

After players finished the learning, the 'Test' button will be shown. The players are required to complete the test before unlocking new facility.

6

Research Lab

Basic Biology | Genetic Algorithm | Real-World Problems

Natural Selection

Priestess

Choice 1
Choice 2

Image

Priestess

The test may vary on the subject. But the main system used is a conversation with NPC and a puzzle.

7

Research Lab

Skip

Congratulation
You get the test prize

X

Priestess

After completing the test, the reward can be claimed.

8

Research Lab

Brief Description

01 / 01 / 2579

Money

Locked
Locked
Locked
Locked
Locked

Locked
Locked
Locked

Factory
City
Habitat

The tutorial phase end with this scence where the brief description of each button/system is introduced.

Figure A.1: Storyboard part 1 out of 4

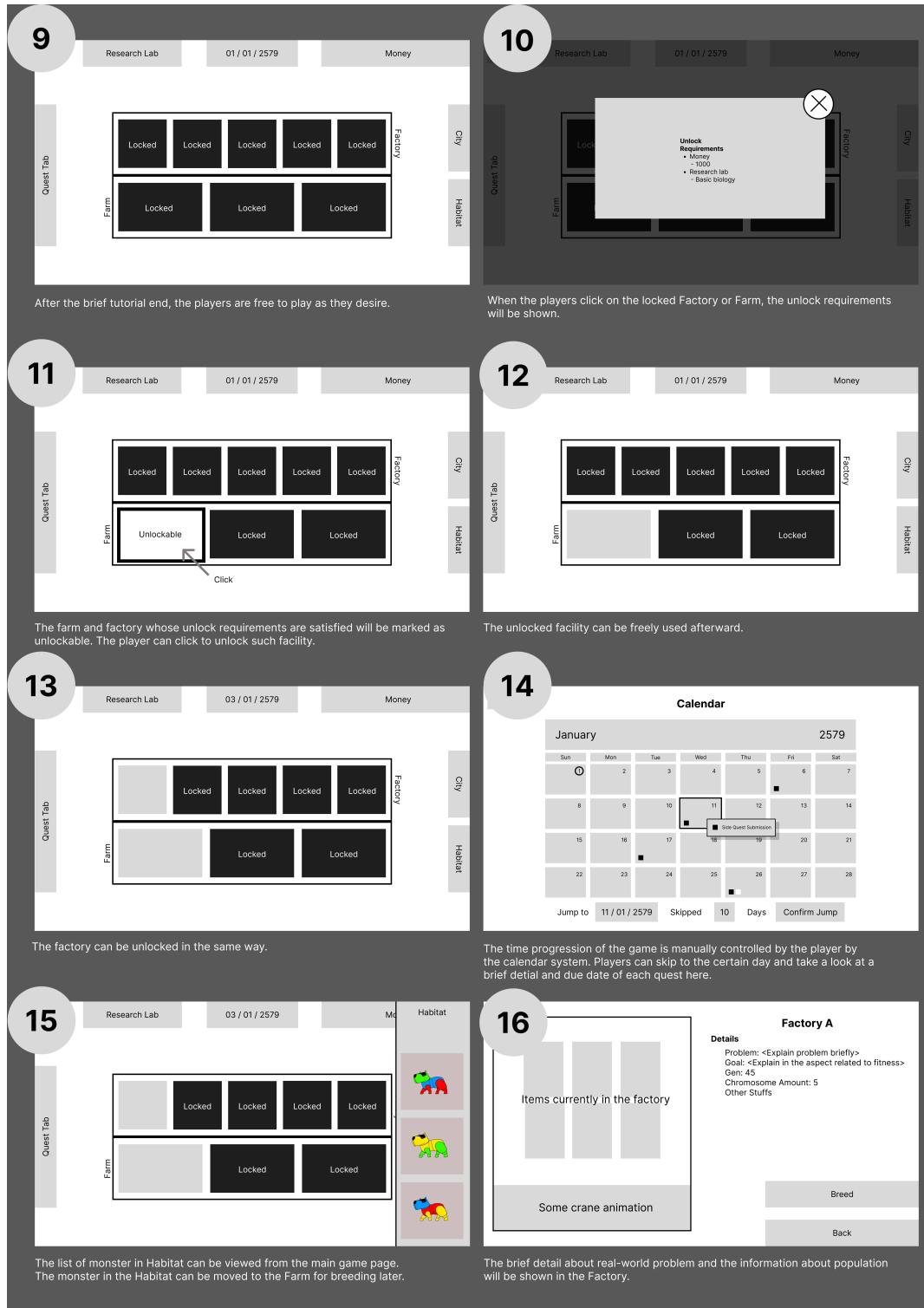


Figure A.2: Storyboard part 2 out of 4

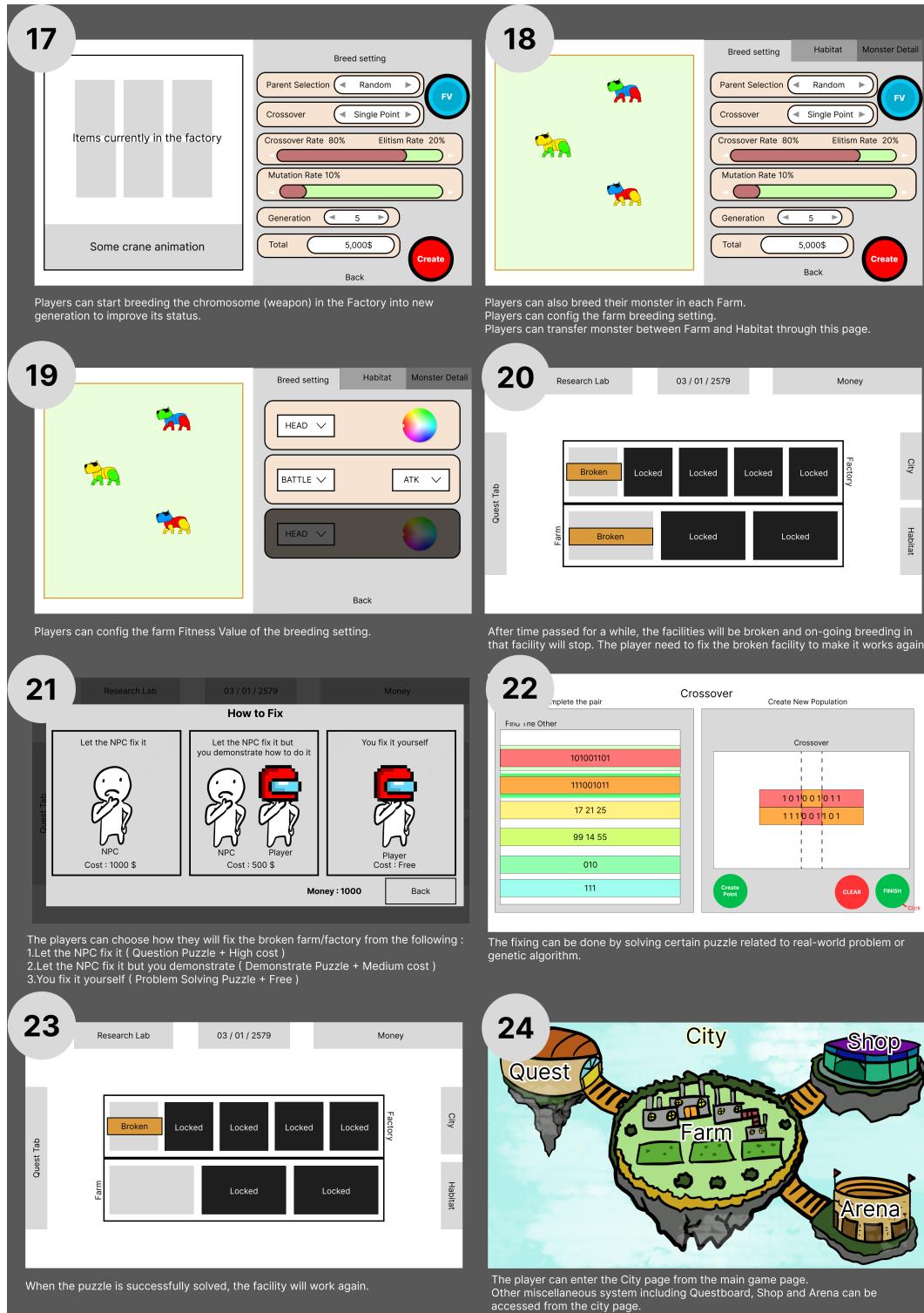


Figure A.3: Storyboard part 3 out of 4



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. The learning material will be grouped into a topic as we described in the system details under the gameplay section in Chapter 3.

- Basic Biology

1. On the Origin of Species

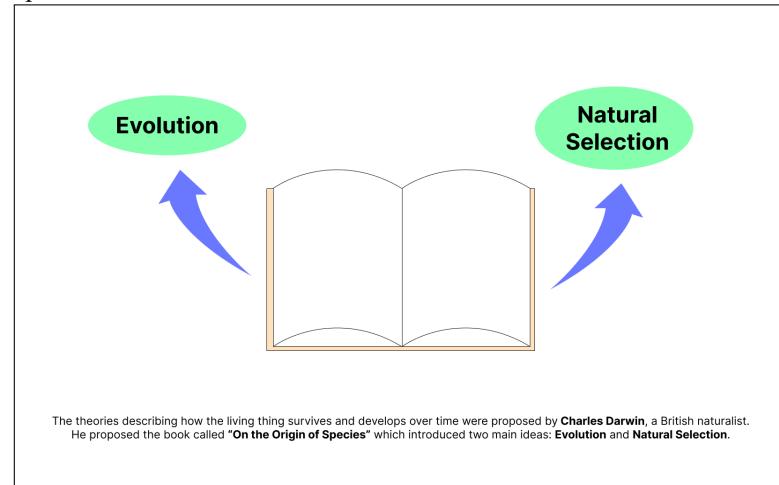


Figure B.1: Learning material 1 out of 6 of the topic: On the Origin of Species

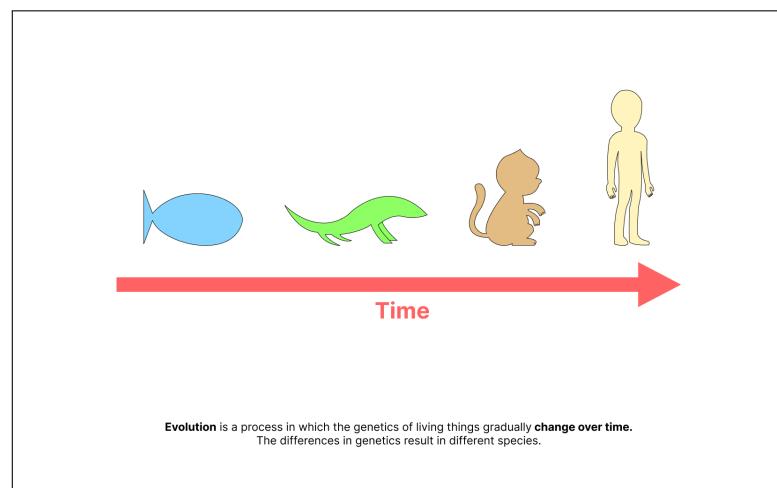


Figure B.2: Learning material 2 out of 6 of the topic: On the Origin of Species

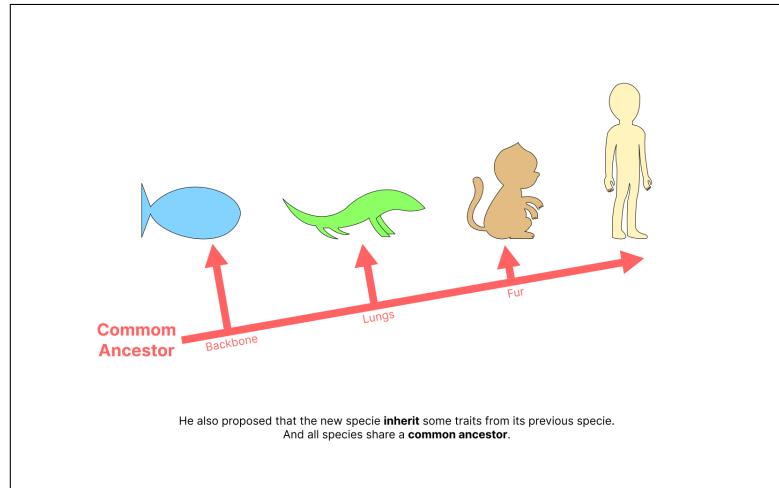


Figure B.3: Learning material 3 out of 6 of the topic: On the Origin of Species

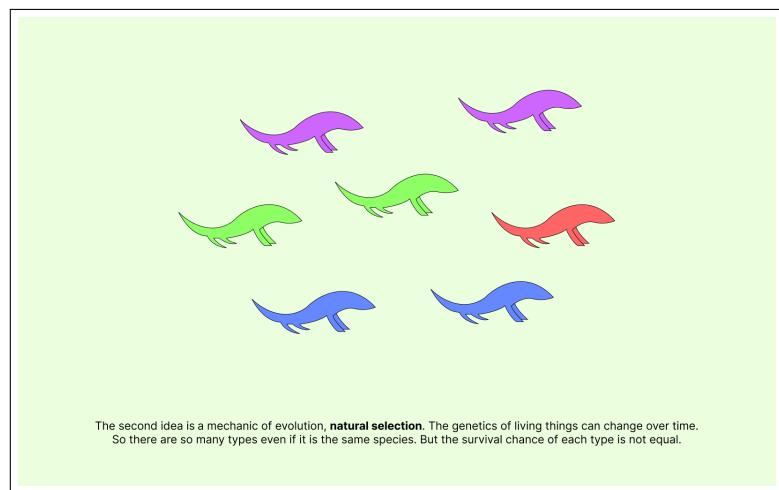


Figure B.4: Learning material 4 out of 6 of the topic: On the Origin of Species

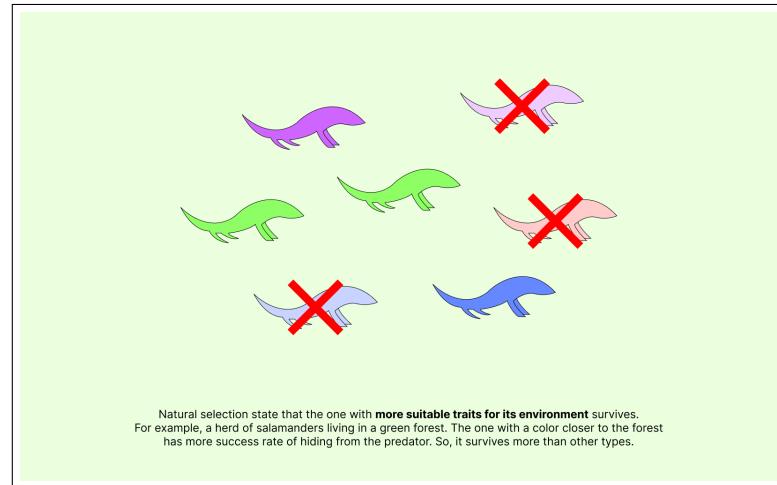


Figure B.5: Learning material 5 out of 6 of the topic: On the Origin of Species

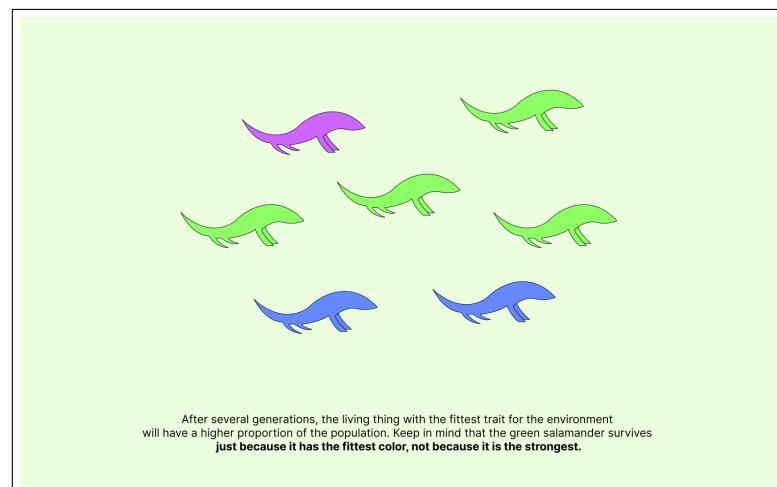


Figure B.6: Learning material 6 out of 6 of the topic: On the Origin of Species

2. The Genetic

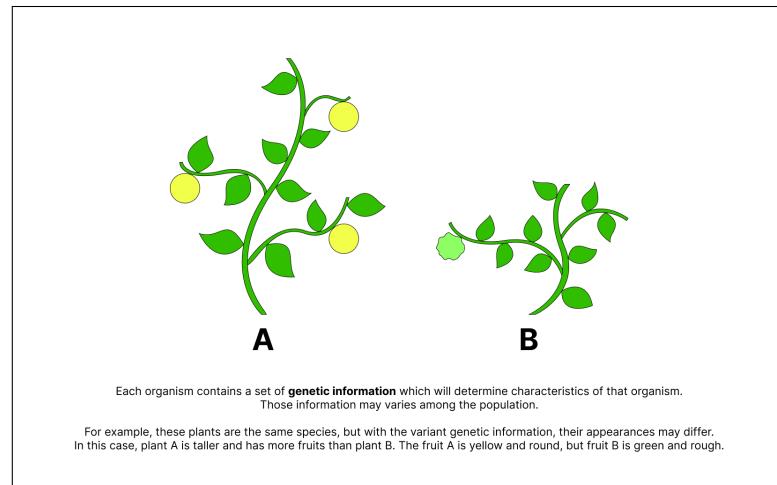


Figure B.7: Learning material 1 out of 4 of the topic: The Genetic

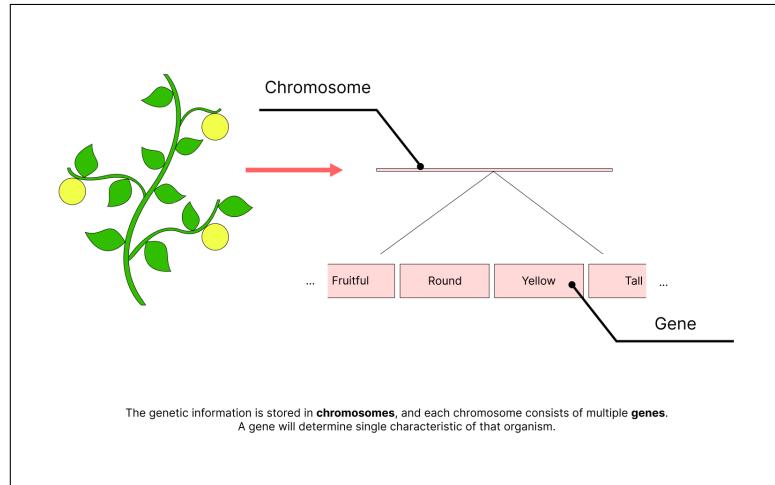


Figure B.8: Learning material 2 out of 4 of the topic: The Genetic

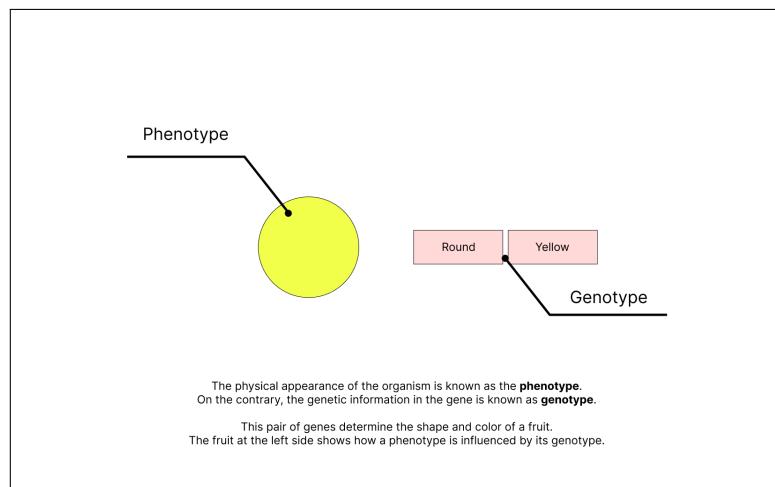


Figure B.9: Learning material 3 out of 4 of the topic: The Genetic

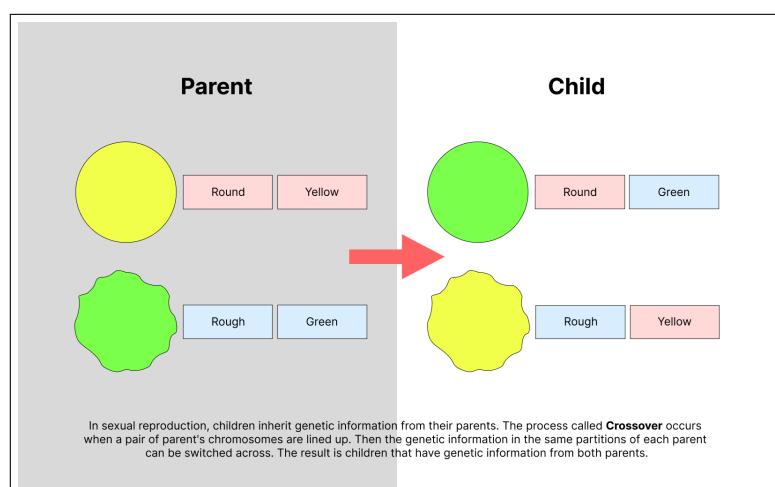


Figure B.10: Learning material 4 out of 4 of the topic: The Genetic

- Genetic Algorithm

1. The Flow of Genetic Algorithm

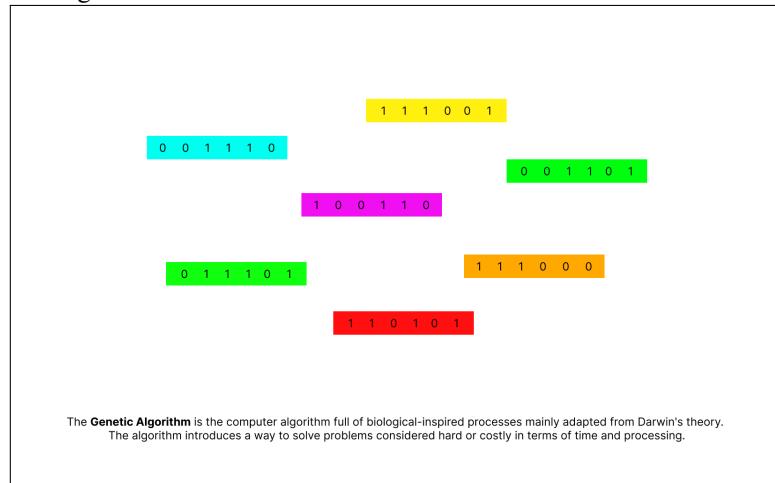


Figure B.11: Learning material 1 out of 9 of the topic: The Flow of Genetic Algorithm

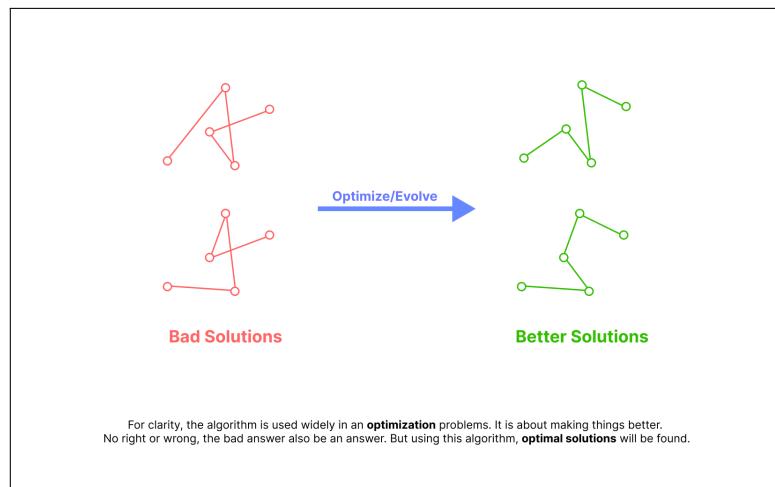


Figure B.12: Learning material 2 out of 9 of the topic: The Flow of Genetic Algorithm

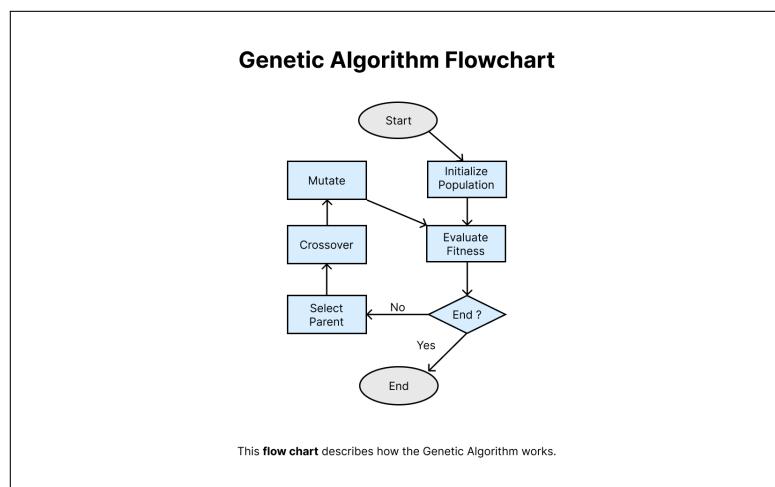


Figure B.13: Learning material 3 out of 9 of the topic: The Flow of Genetic Algorithm

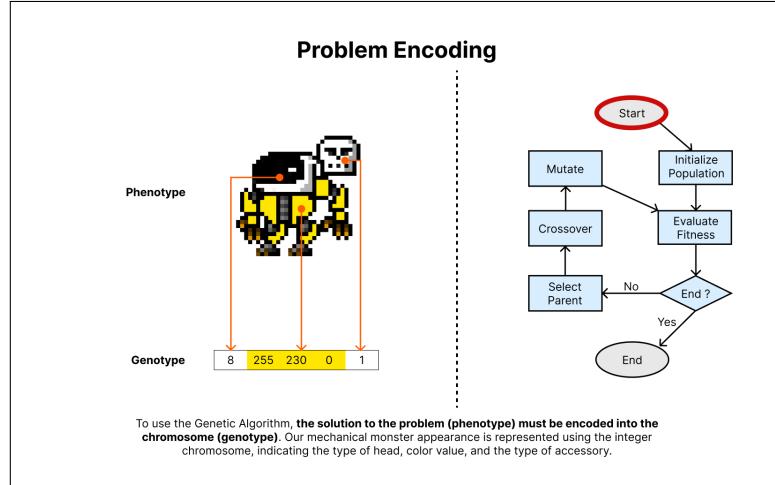


Figure B.14: Learning material 4 out of 9 of the topic: The Flow of Genetic Algorithm

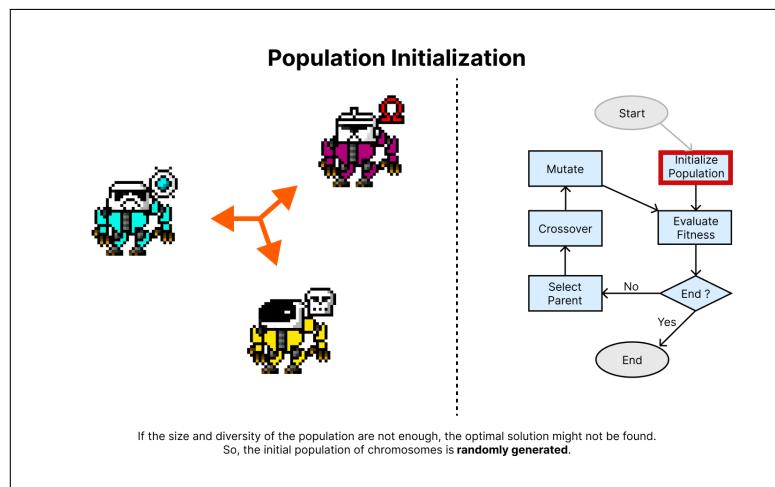


Figure B.15: Learning material 5 out of 9 of the topic: The Flow of Genetic Algorithm

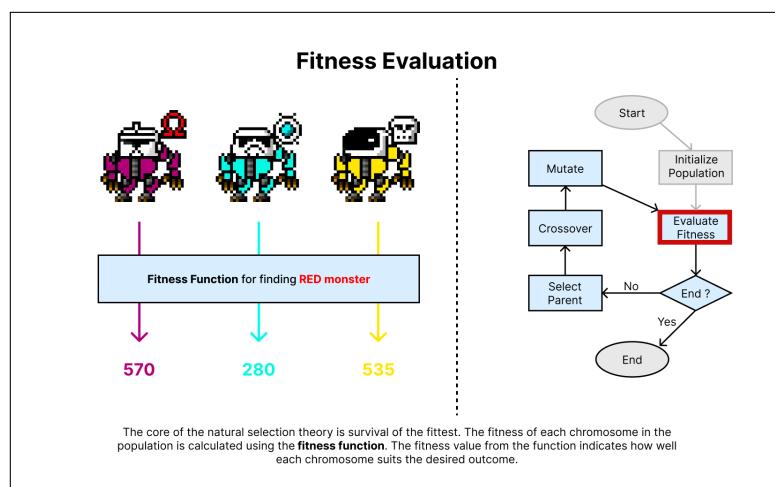


Figure B.16: Learning material 6 out of 9 of the topic: The Flow of Genetic Algorithm

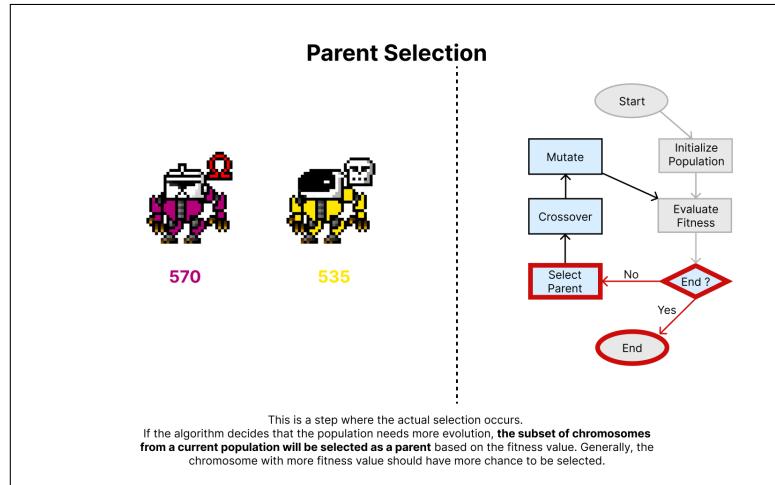


Figure B.17: Learning material 7 out of 9 of the topic: The Flow of Genetic Algorithm

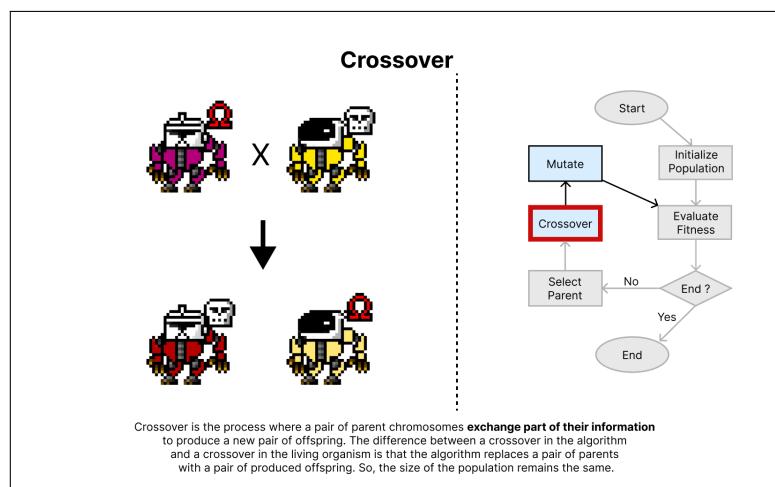


Figure B.18: Learning material 8 out of 9 of the topic: The Flow of Genetic Algorithm

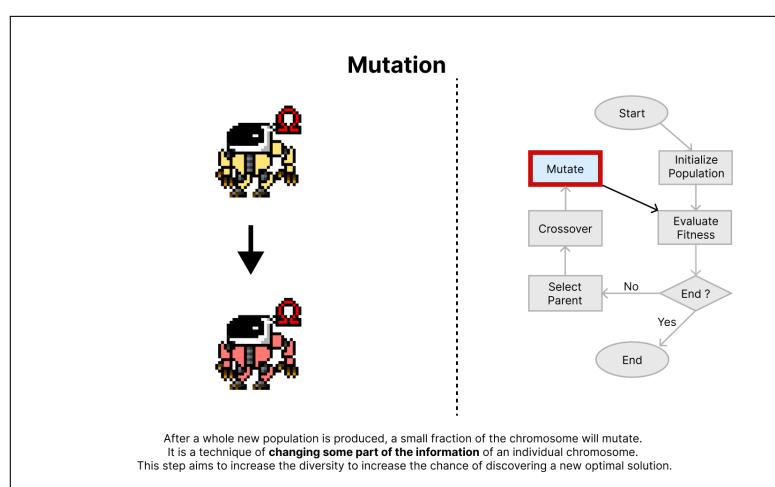


Figure B.19: Learning material 9 out of 9 of the topic: The Flow of Genetic Algorithm

2. Parent Selection

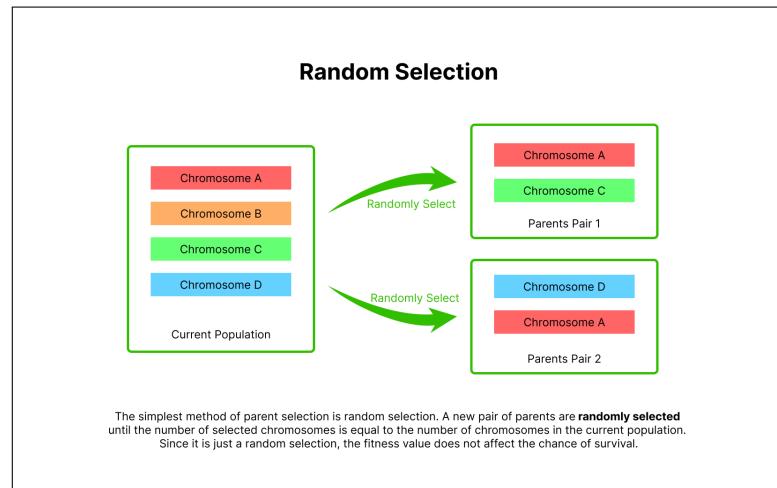


Figure B.20: Learning material of the topic: Random Selection

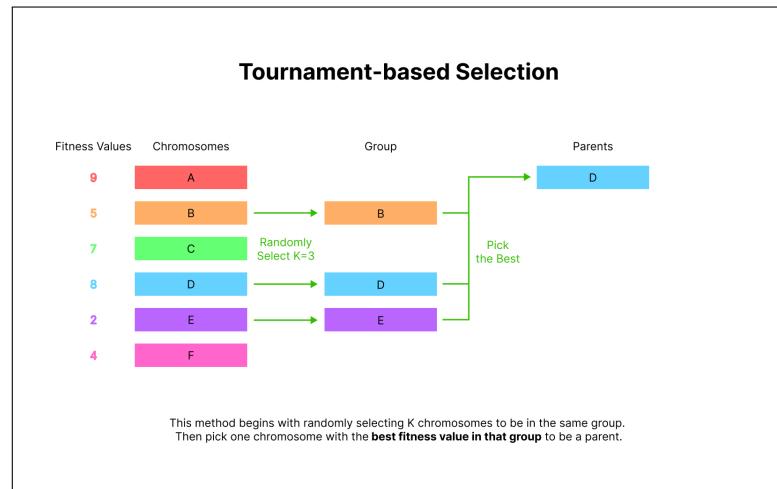


Figure B.21: Learning material 1 out of 2 of the topic: Tournament-based Selection

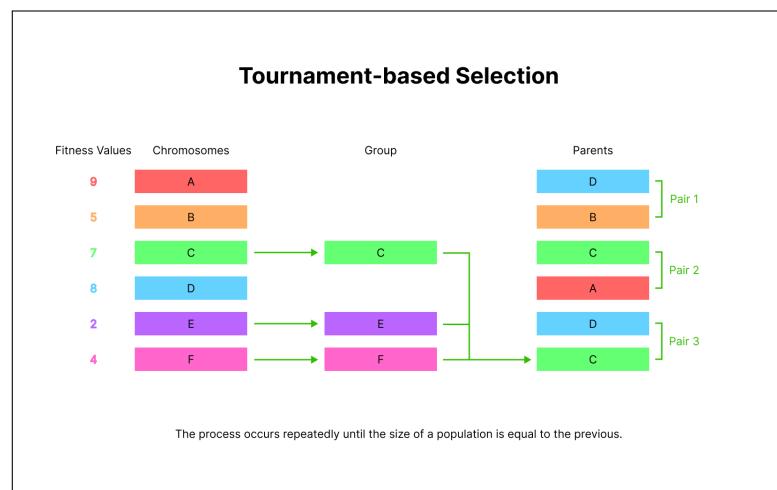


Figure B.22: Learning material 2 out of 2 of the topic: Tournament-based Selection

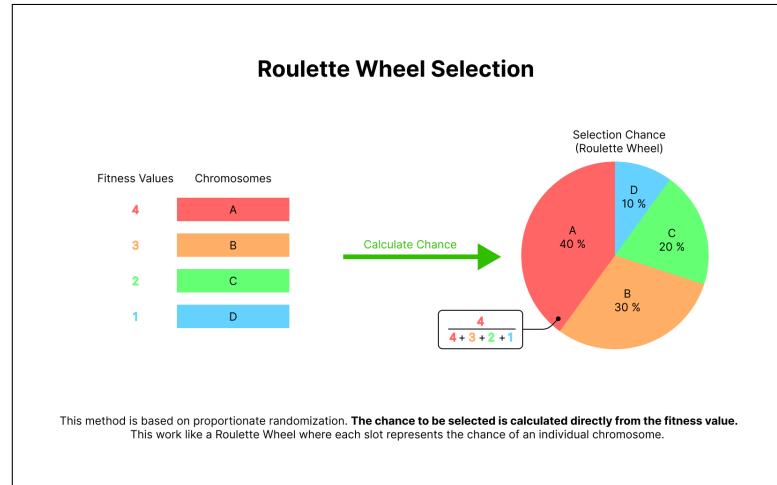


Figure B.23: Learning material 1 out of 2 of the topic: Roulette Wheel Selection

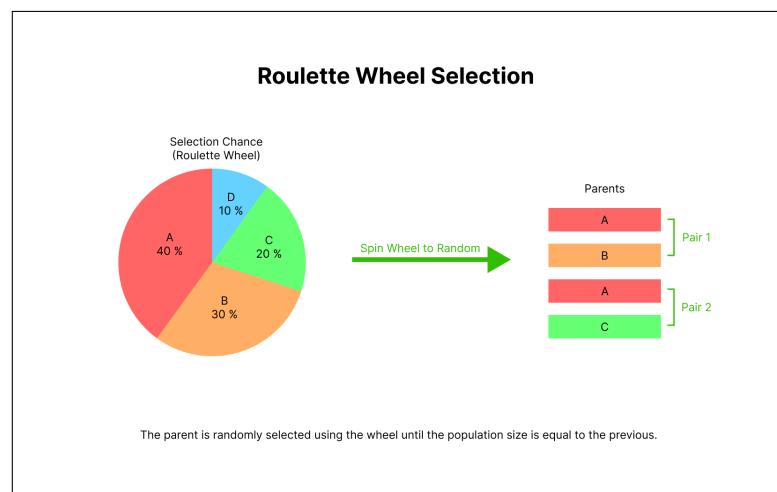


Figure B.24: Learning material 2 out of 2 of the topic: Roulette Wheel Selection

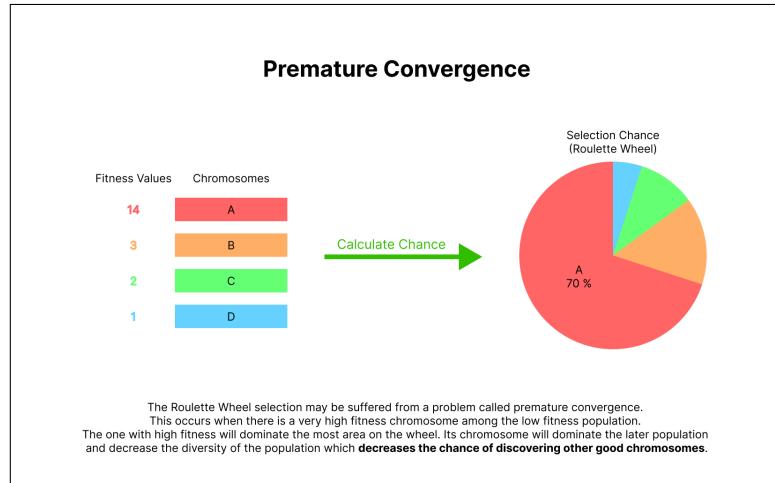


Figure B.25: Learning material of the topic: Premature Convergence

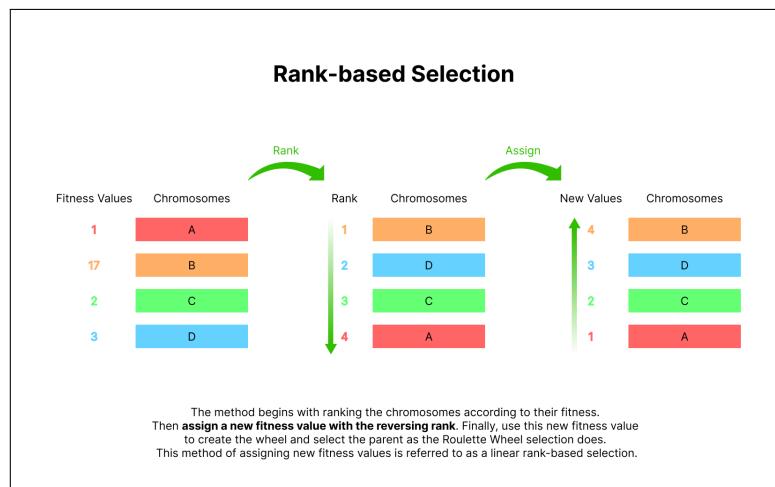


Figure B.26: Learning material of the topic: Rank-based Selection

3. Crossover

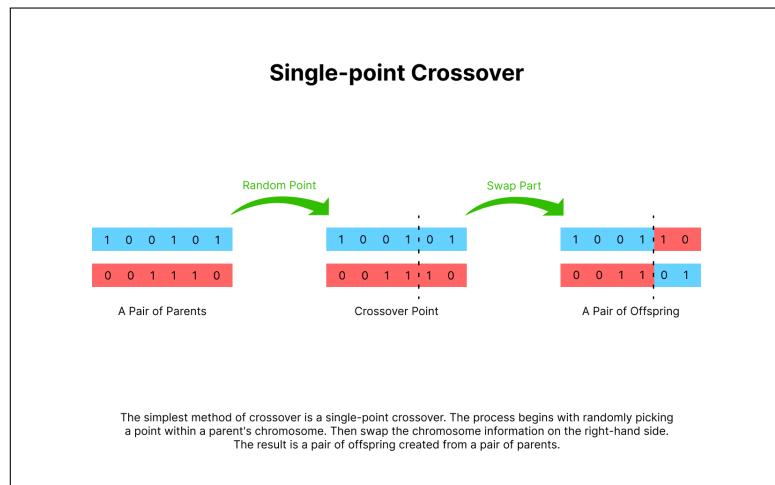


Figure B.27: Learning material of the topic: Single-point Crossover

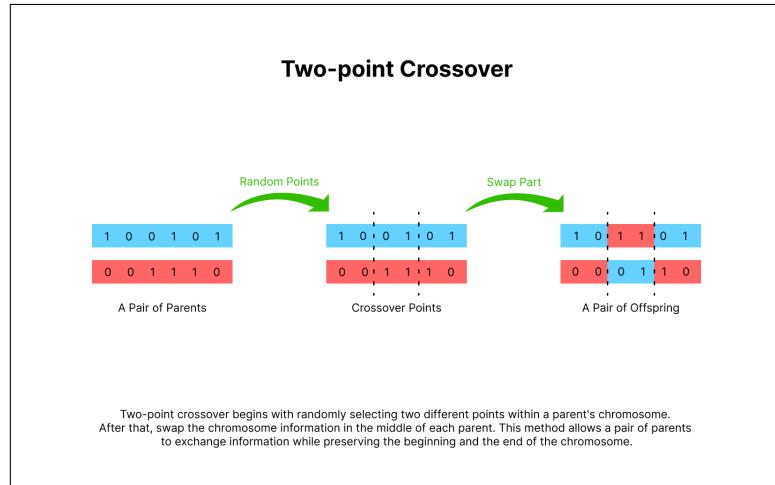


Figure B.28: Learning material of the topic: Two-point Crossover

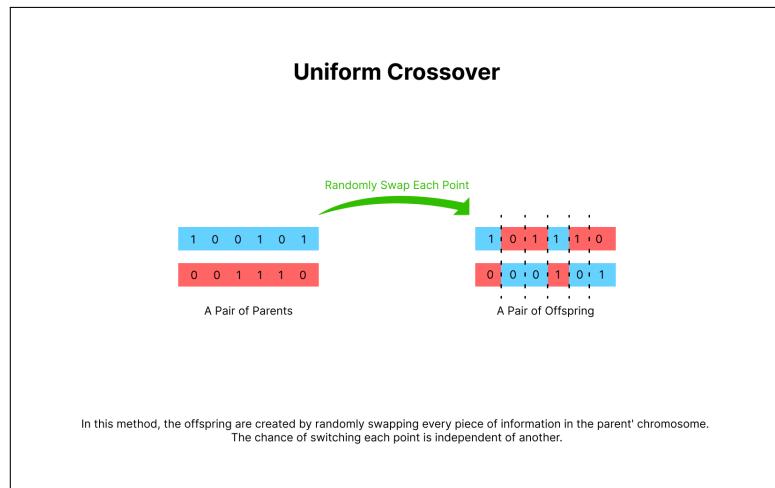


Figure B.29: Learning material of the topic: Uniform Crossover

4. Improvement Techniques

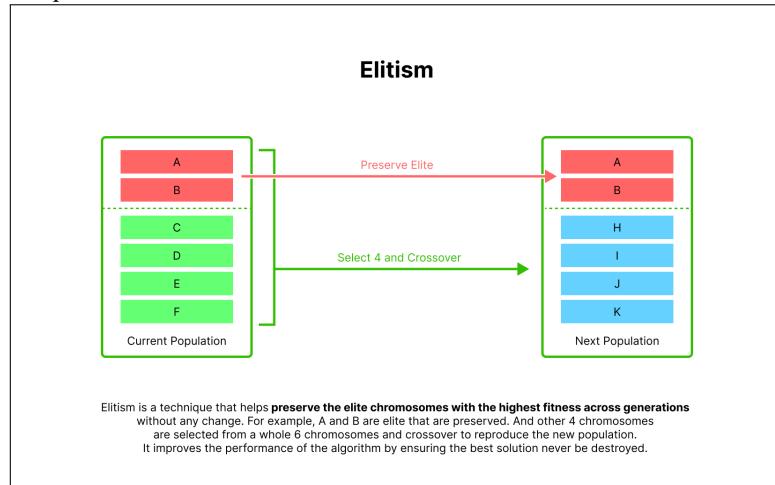


Figure B.30: Learning material of the topic: Elitism

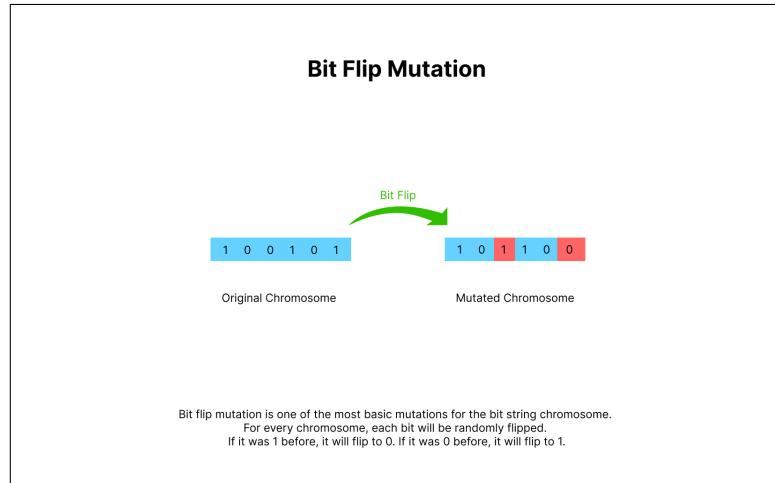


Figure B.31: Learning material of the topic: Bit Flip Mutation

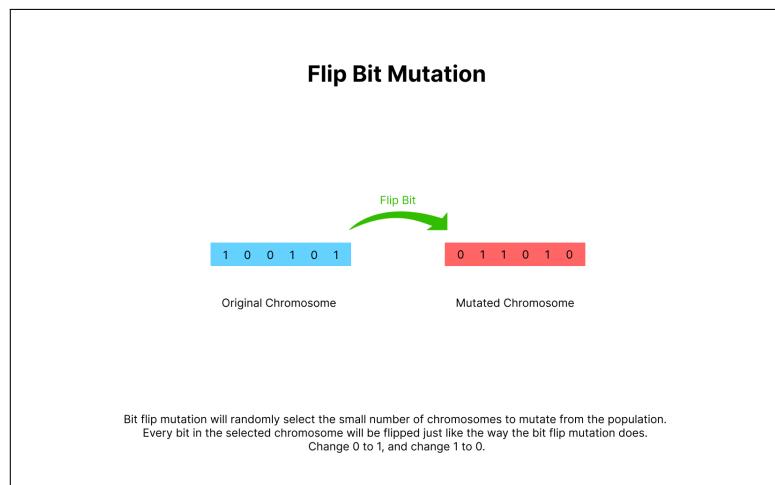


Figure B.32: Learning material of the topic: Flip Bit Mutation

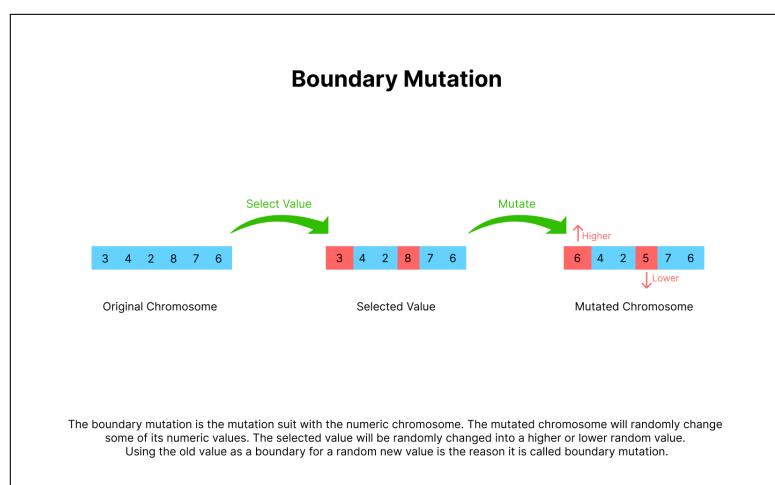


Figure B.33: Learning material of the topic: Boundary Mutation

- Knapsack Problem

1. Standard Knapsack Problem

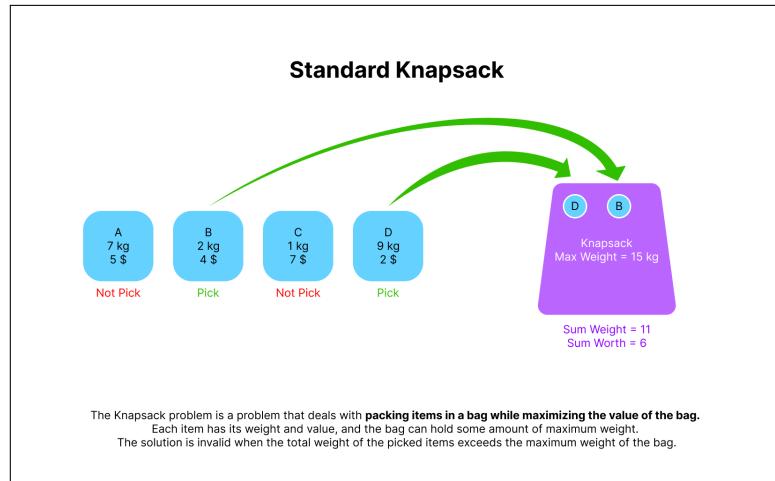


Figure B.34: Learning material 1 out of 5 of the topic: Standard Knapsack Problem

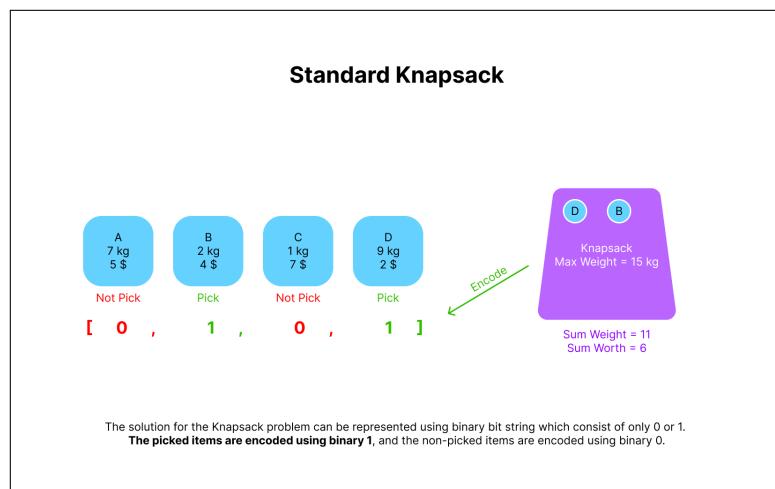


Figure B.35: Learning material 2 out of 5 of the topic: Standard Knapsack Problem

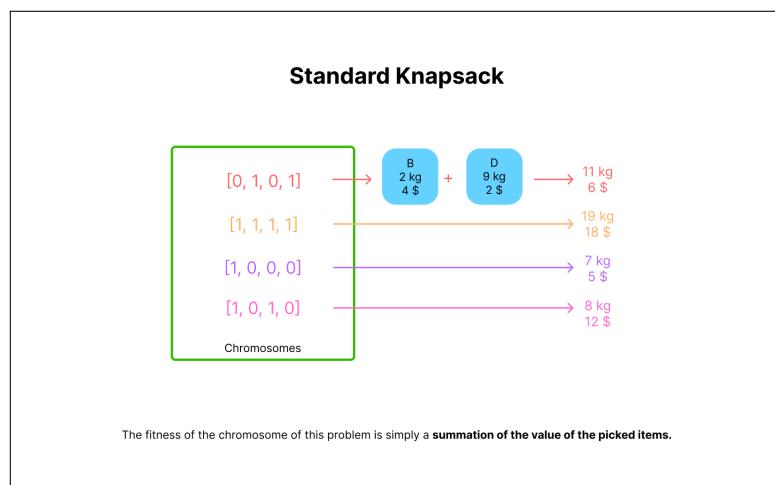


Figure B.36: Learning material 3 out of 5 of the topic: Standard Knapsack Problem

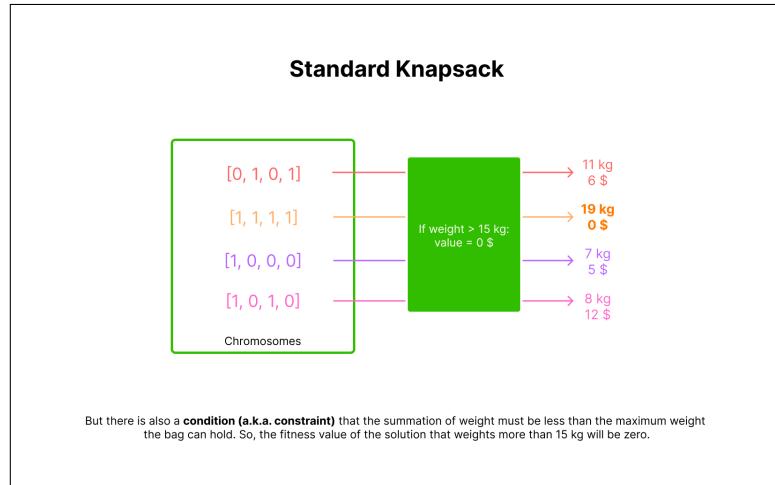


Figure B.37: Learning material 4 out of 5 of the topic: Standard Knapsack Problem

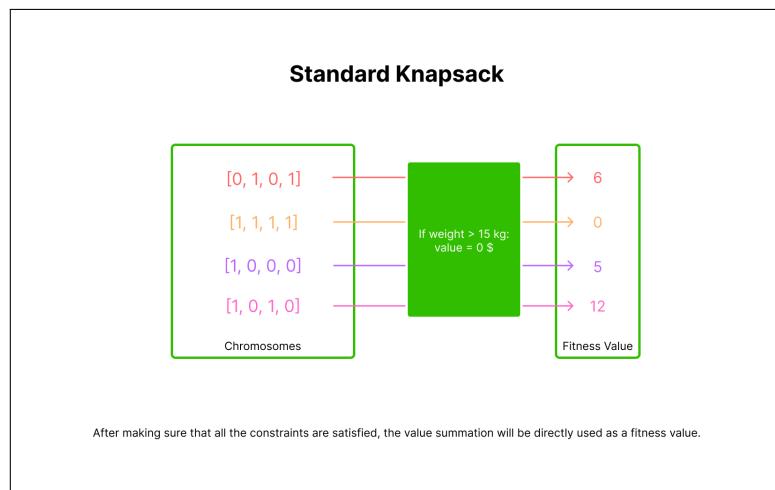


Figure B.38: Learning material 5 out of 5 of the topic: Standard Knapsack Problem

2. Multidimensional Knapsack Problem

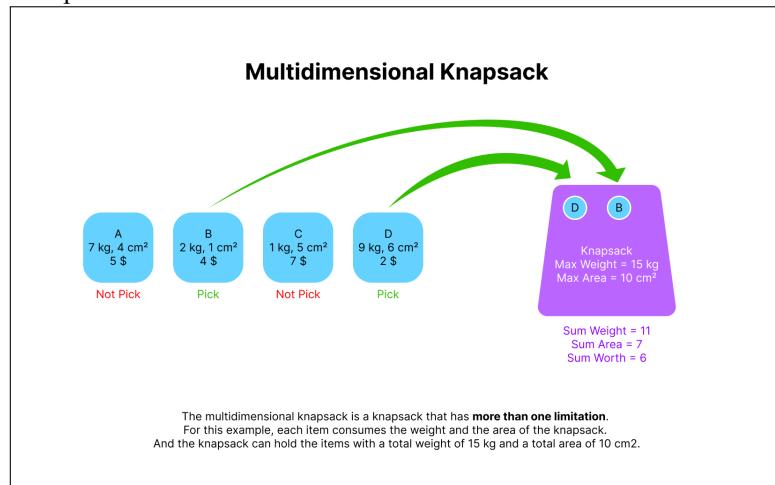


Figure B.39: Learning material 1 out of 5 of the topic: Multidimensional Knapsack Problem

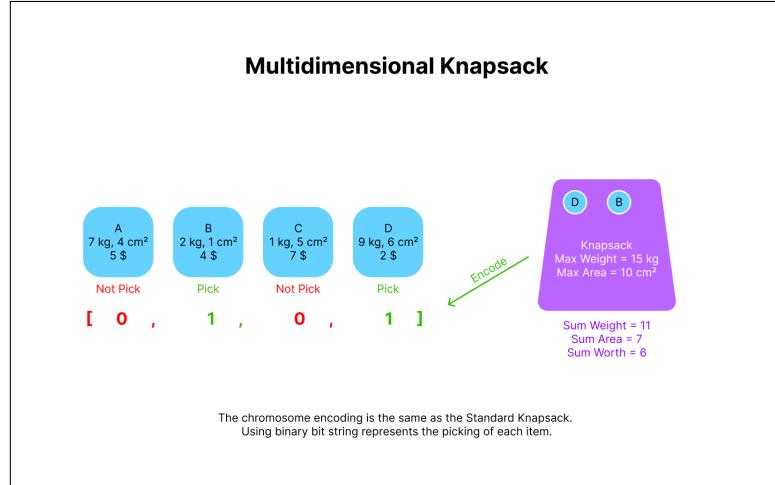


Figure B.40: Learning material 2 out of 5 of the topic: Multidimensional Knapsack Problem

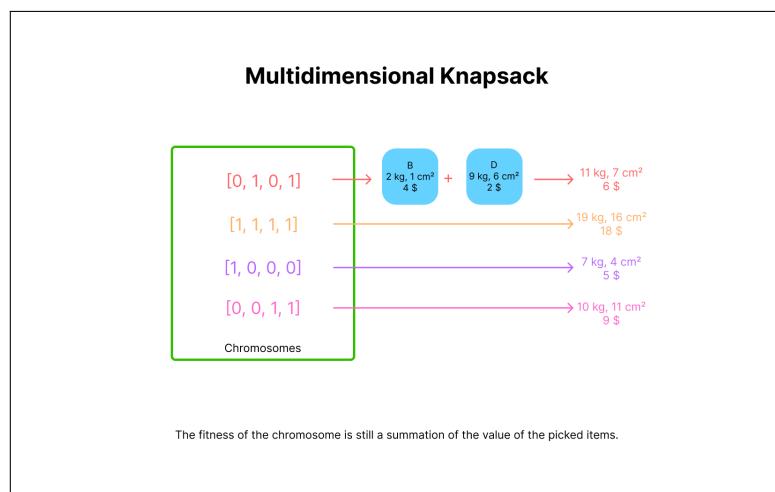


Figure B.41: Learning material 3 out of 5 of the topic: Multidimensional Knapsack Problem

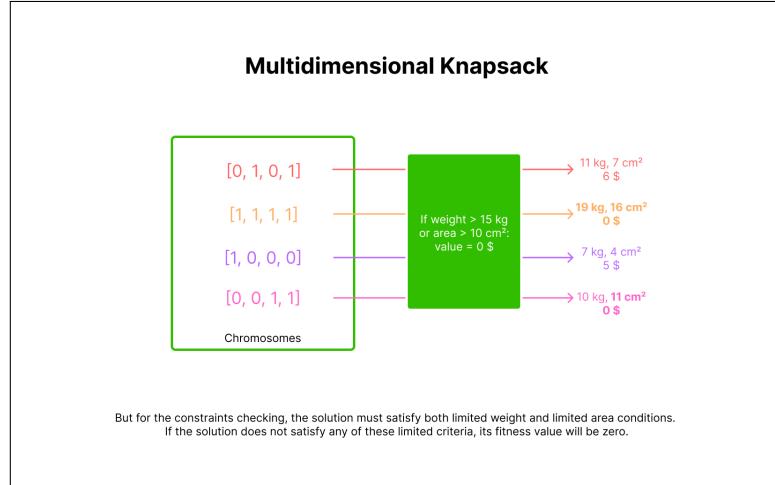


Figure B.42: Learning material 4 out of 5 of the topic: Multidimensional Knapsack Problem

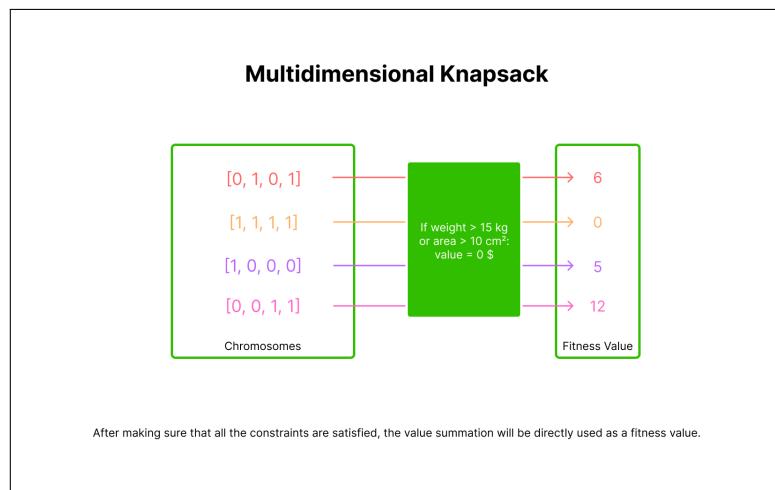


Figure B.43: Learning material 5 out of 5 of the topic: Multidimensional Knapsack Problem

3. Multiple Knapsack Problem

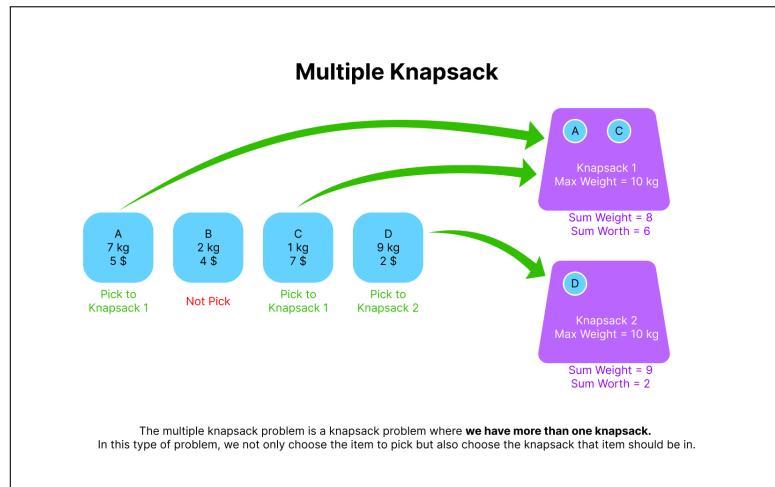


Figure B.44: Learning material 1 out of 5 of the topic: Multiple Knapsack Problem

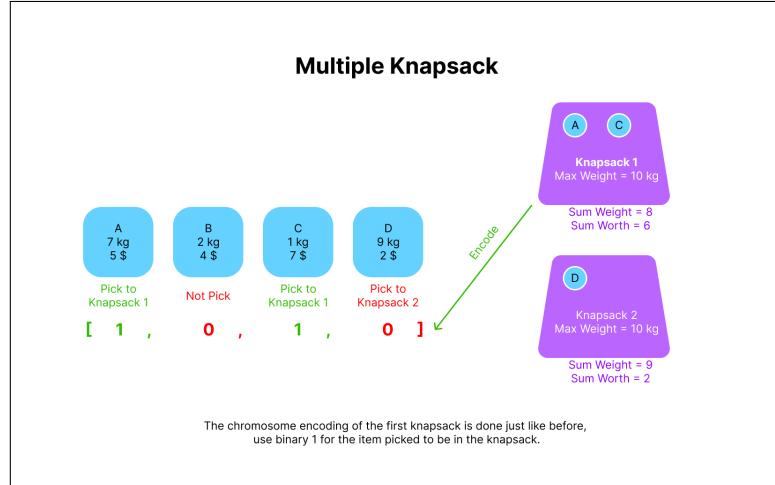


Figure B.45: Learning material 2 out of 5 of the topic: Multiple Knapsack Problem

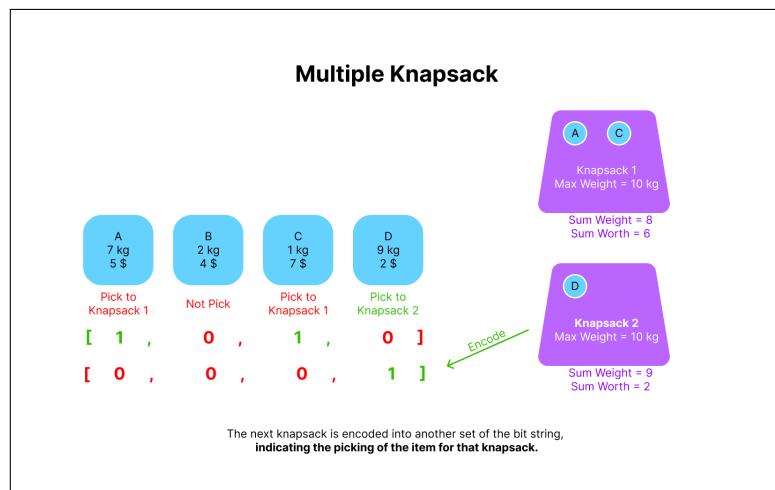


Figure B.46: Learning material 3 out of 5 of the topic: Multiple Knapsack Problem

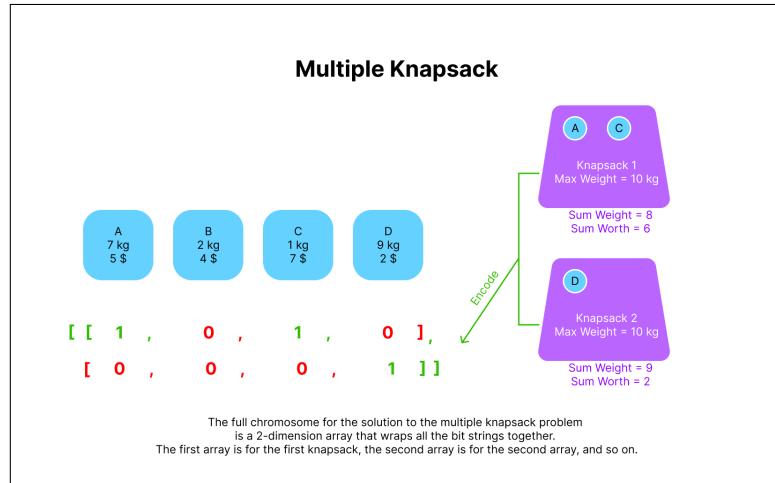


Figure B.47: Learning material 4 out of 5 of the topic: Multiple Knapsack Problem

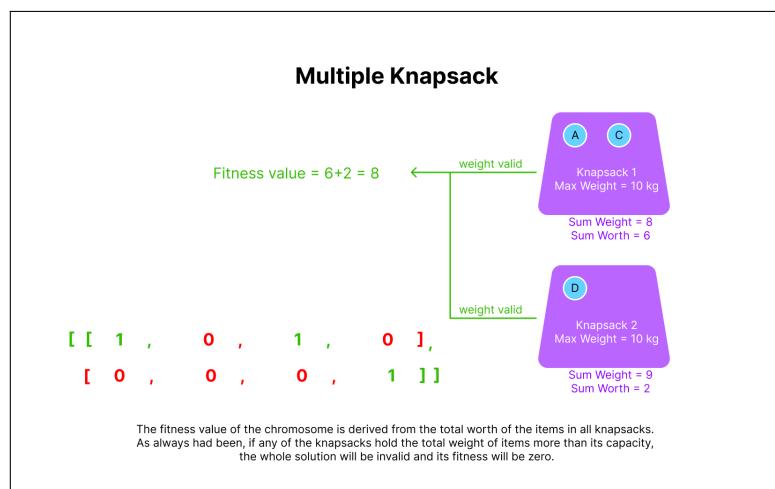


Figure B.48: Learning material 5 out of 5 of the topic: Multiple Knapsack Problem

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

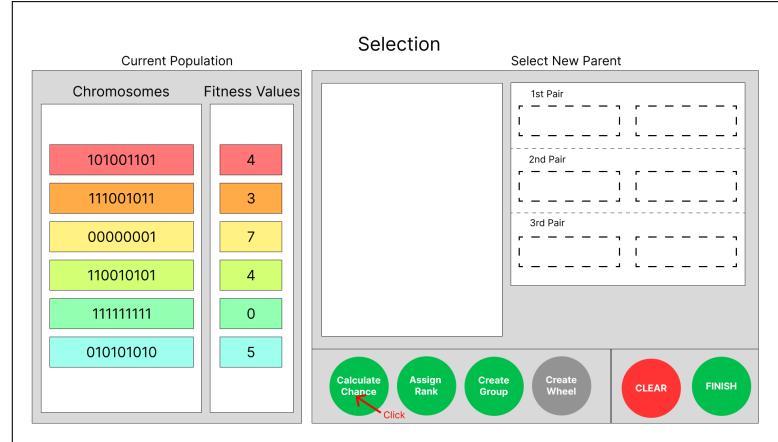


Figure C.1: Demonstration Puzzle of Roulette Wheel Selection 1 out of 6

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

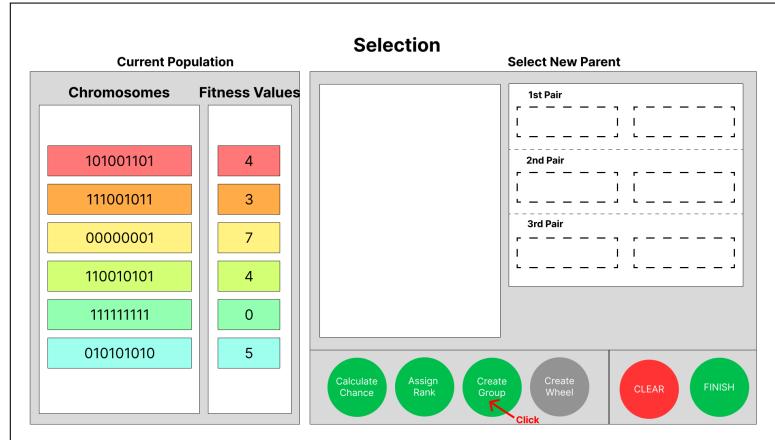


Figure C.7: Demonstration Puzzle of Tournament-based Selection 1 out of 4

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.

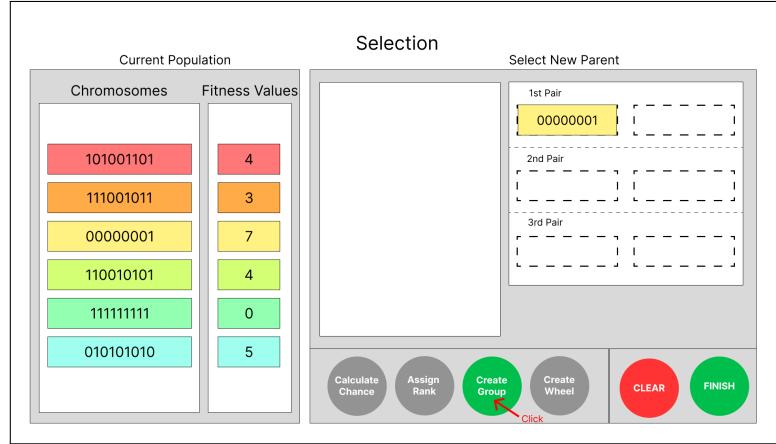


Figure C.9: Demonstration Puzzle of Tournament-based Selection 3 out of 4

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

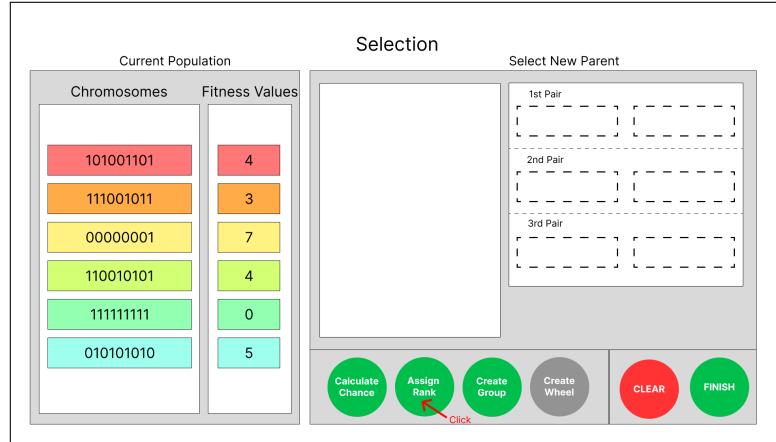


Figure C.11: Demonstration Puzzle of Rank-based Selection 1 out of 11

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

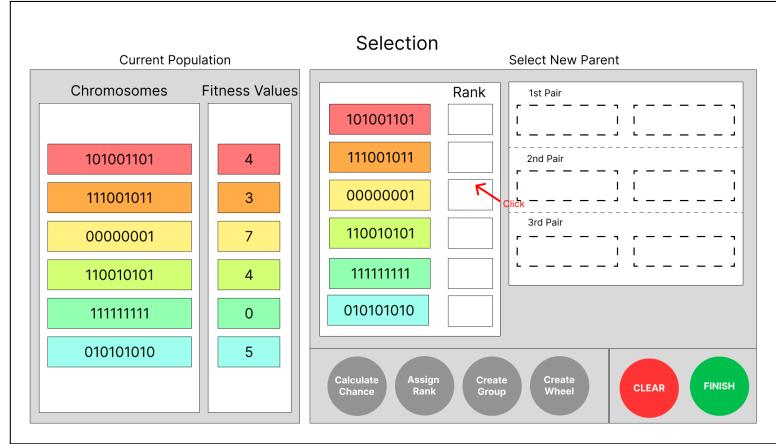


Figure C.12: Demonstration Puzzle of Rank-based Selection 2 out of 11

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

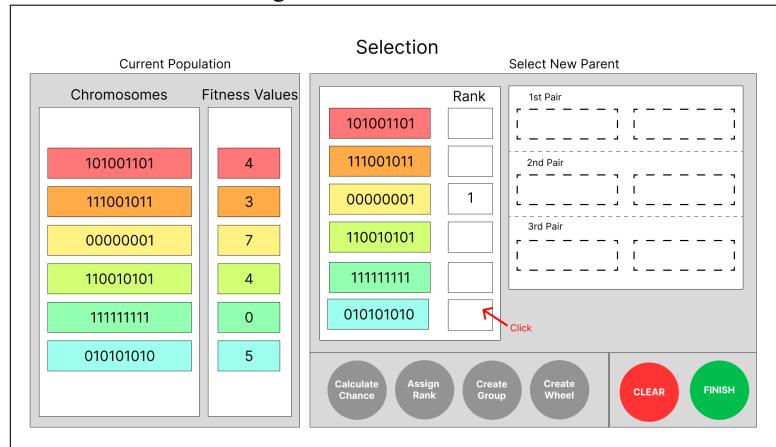


Figure C.13: Demonstration Puzzle of Rank-based Selection 3 out of 11

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.

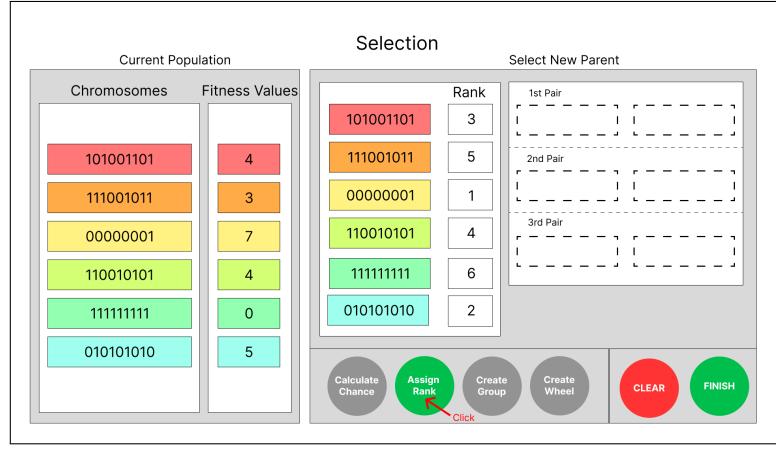


Figure C.15: Demonstration Puzzle of Rank-based Selection 5 out of 11

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

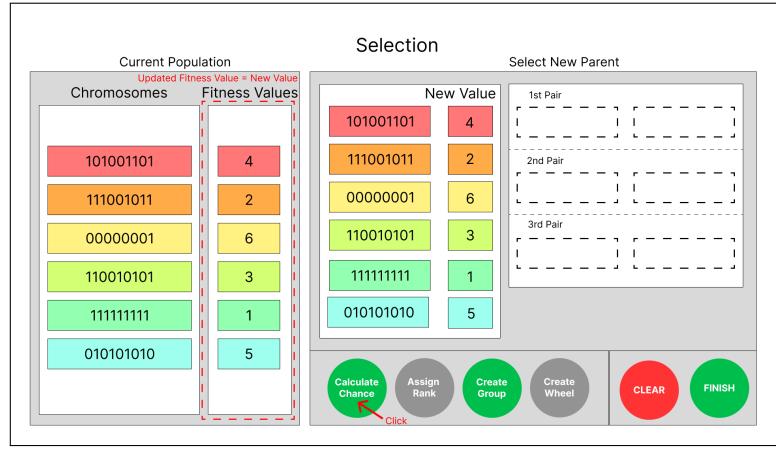


Figure C.16: Demonstration Puzzle of Rank-based Selection 6 out of 11

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

- Single-point Crossover

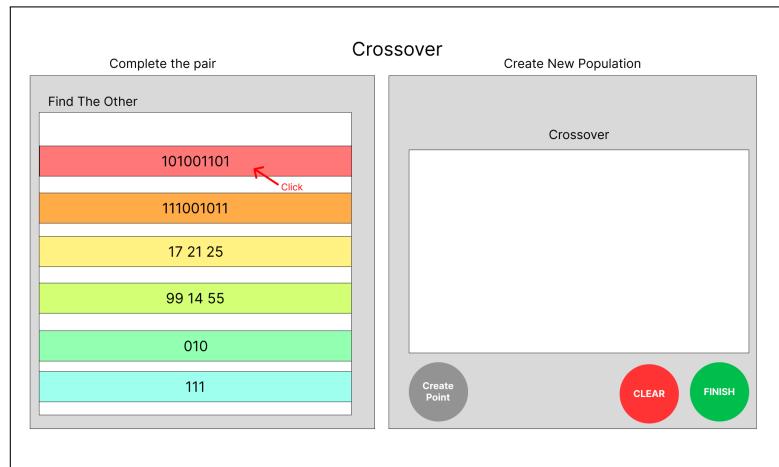


Figure C.22: Demonstration Puzzle of Single-point Crossover 1 out of 5

The players click the chromosome they want to use.

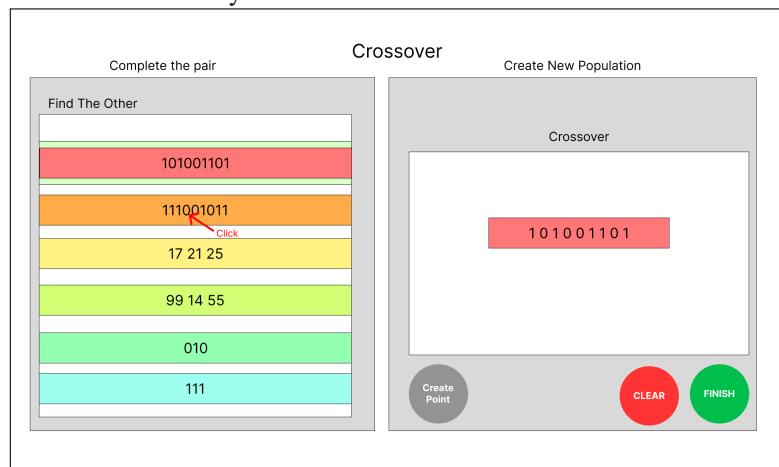


Figure C.23: Demonstration Puzzle of Single-point Crossover 2 out of 5

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

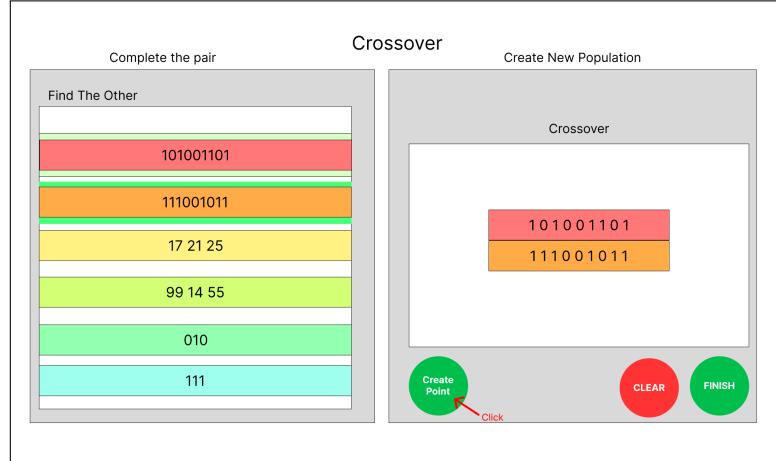


Figure C.24: Demonstration Puzzle of Single-point Crossover 3 out of 5

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

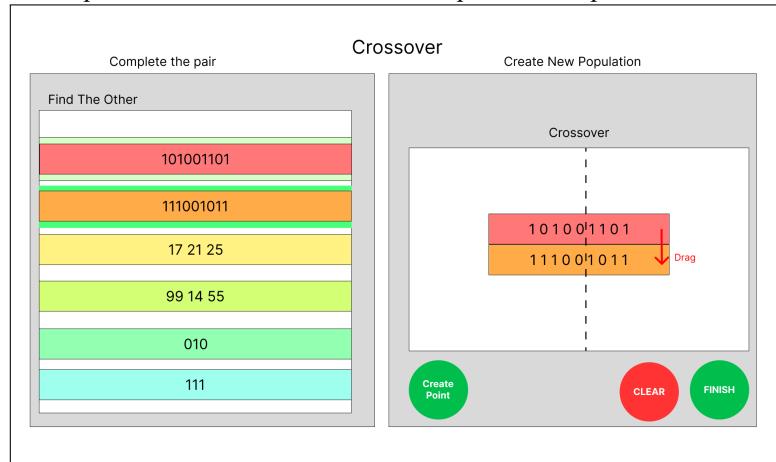


Figure C.25: Demonstration Puzzle of Single-point Crossover 4 out of 5

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

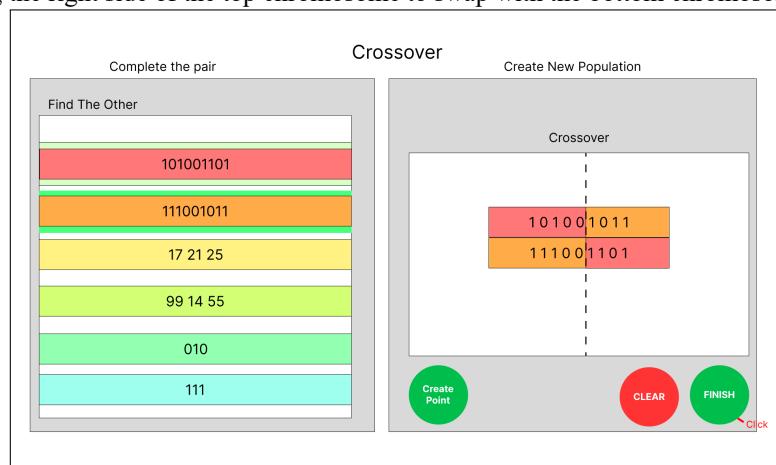


Figure C.26: Demonstration Puzzle of Single-point Crossover 5 out of 5

The players click finish to send the answer.

2. Two-point Crossover

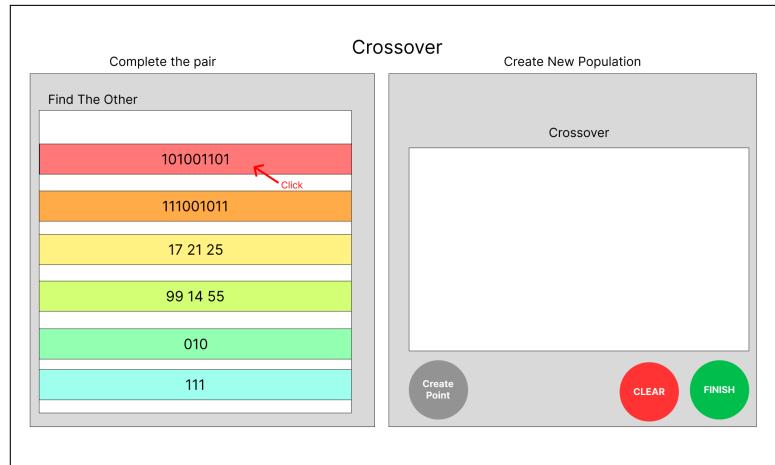


Figure C.27: Demonstration Puzzle of Two-point Crossover 1 out of 6

The players click the chromosome they want to use.

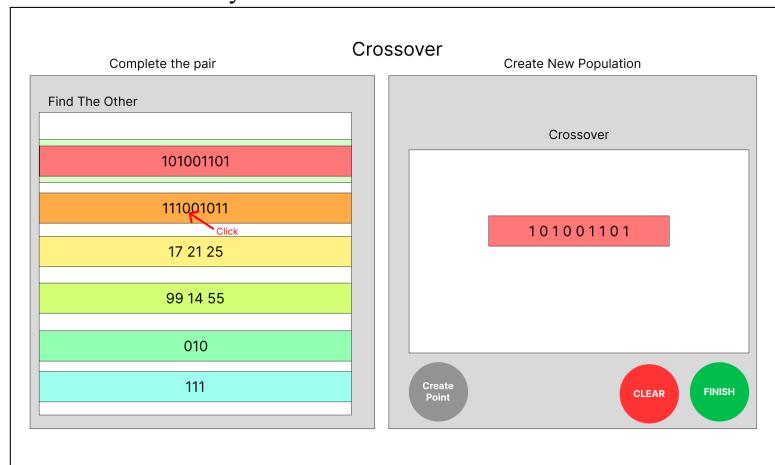


Figure C.28: Demonstration Puzzle of Two-point Crossover 2 out of 6

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

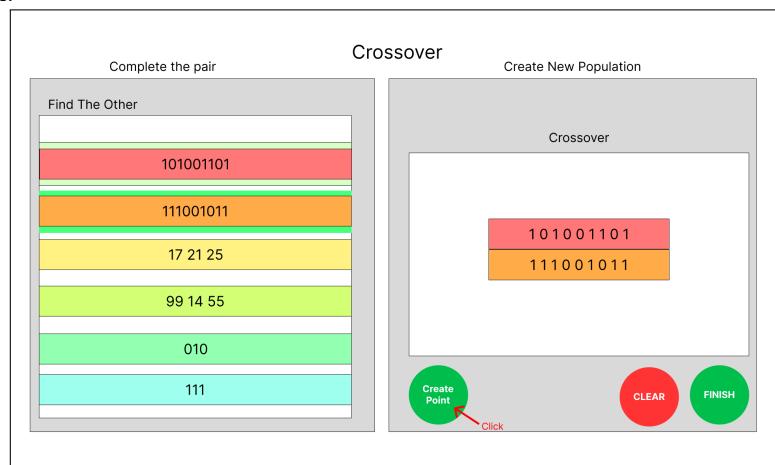


Figure C.29: Demonstration Puzzle of Two-point Crossover 3 out of 6

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

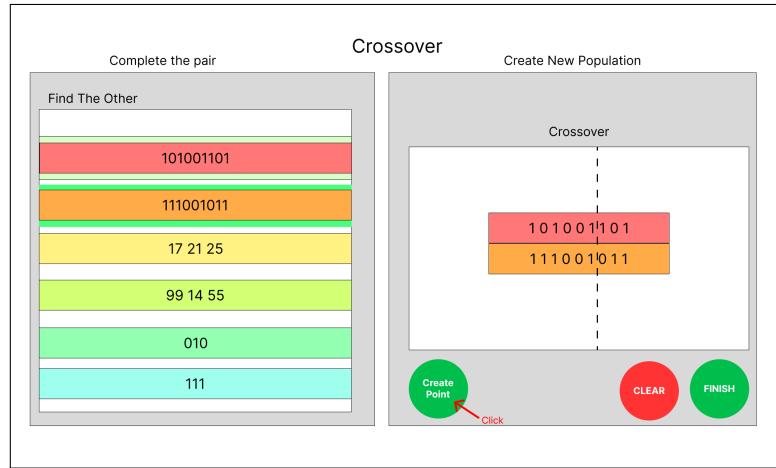


Figure C.30: Demonstration Puzzle of Two-point Crossover 4 out of 6

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

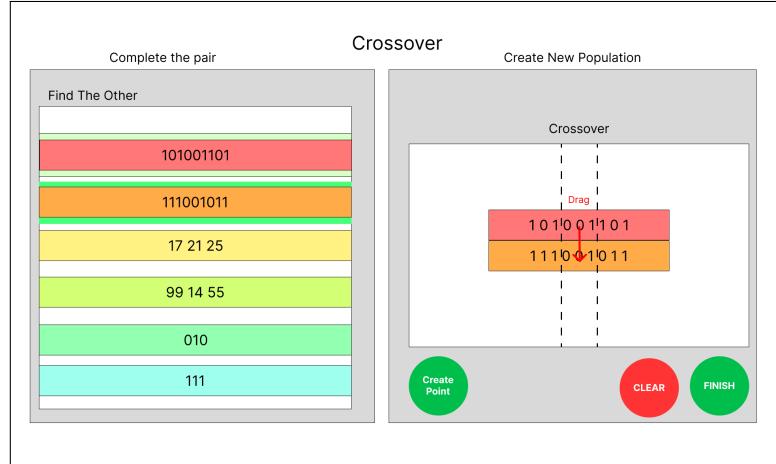


Figure C.31: Demonstration Puzzle of Two-point Crossover 5 out of 6

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

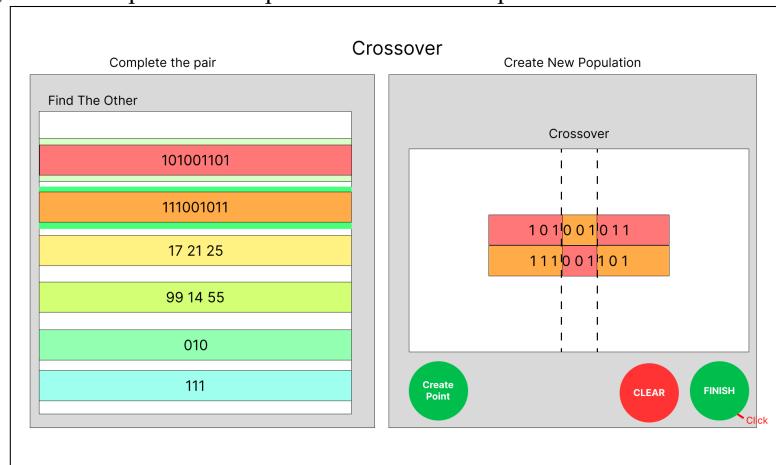


Figure C.32: Demonstration Puzzle of Two-point Crossover 6 out of 6

The players click finish to send the answer.

- Selection Problem-solving

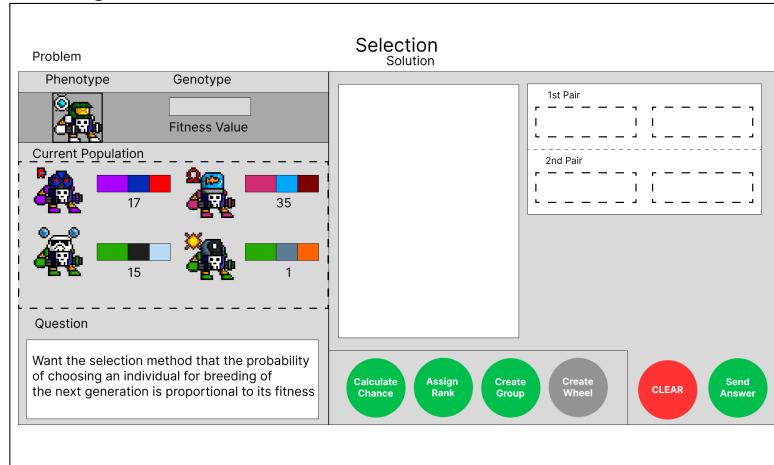


Figure C.33: Problem-solving Puzzle of Roulette Wheel Selection

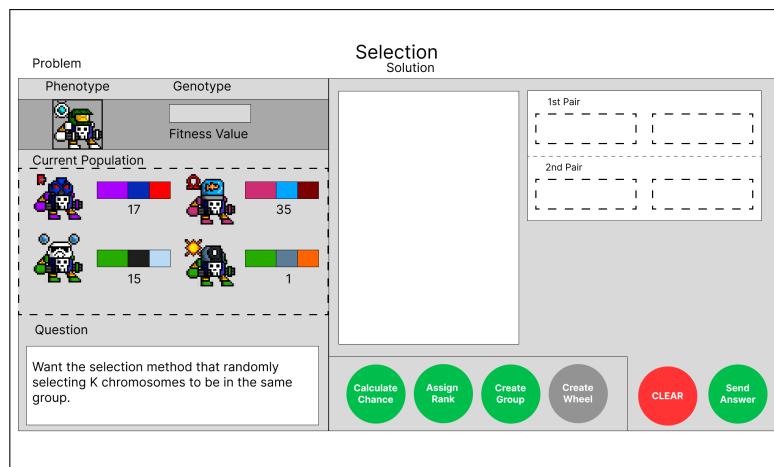


Figure C.34: Problem-solving Puzzle of Tournament-based Selection

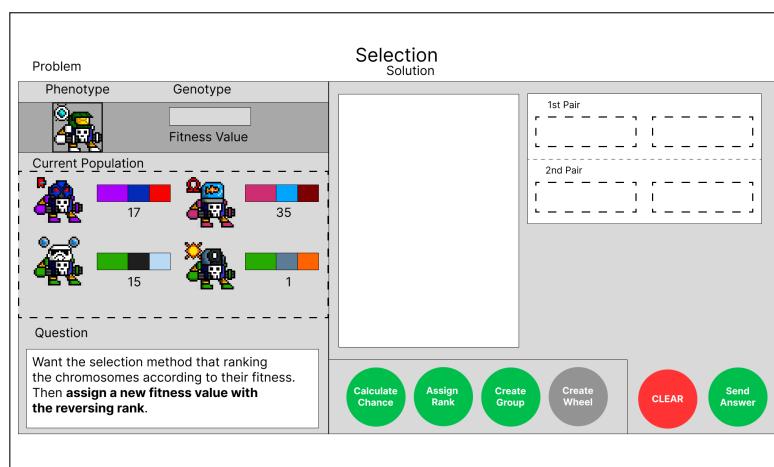


Figure C.35: Problem-solving Puzzle of Rank-based Selection

- Crossover Problem-solving

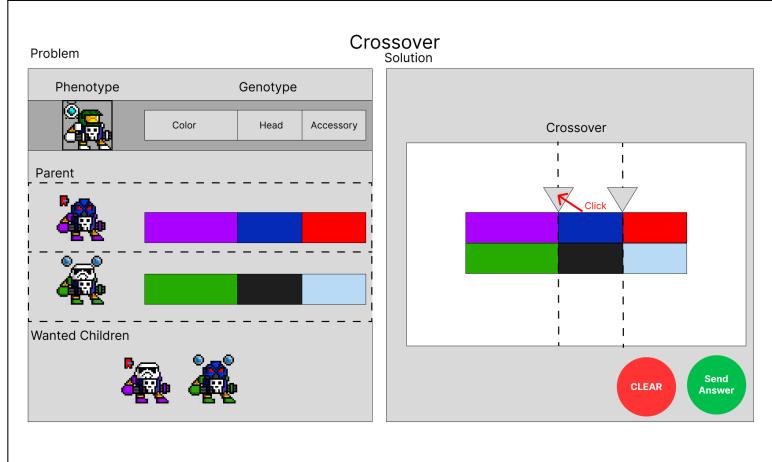


Figure C.36: Problem-solving Puzzle of Crossover 1 out of 4

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

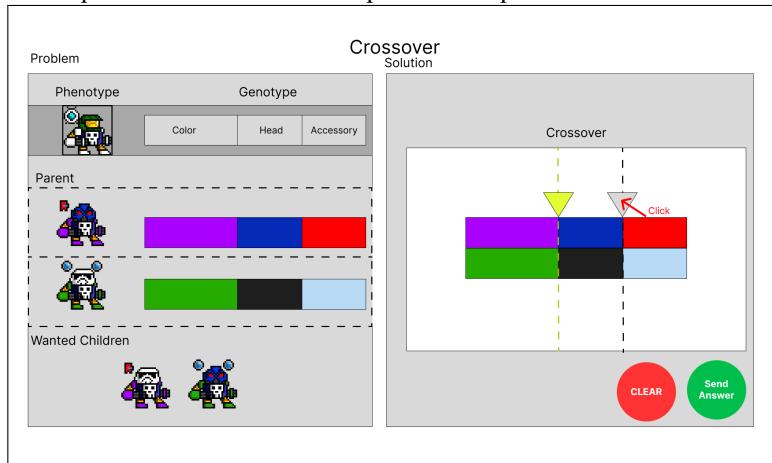


Figure C.37: Problem-solving Puzzle of Crossover 2 out of 4

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

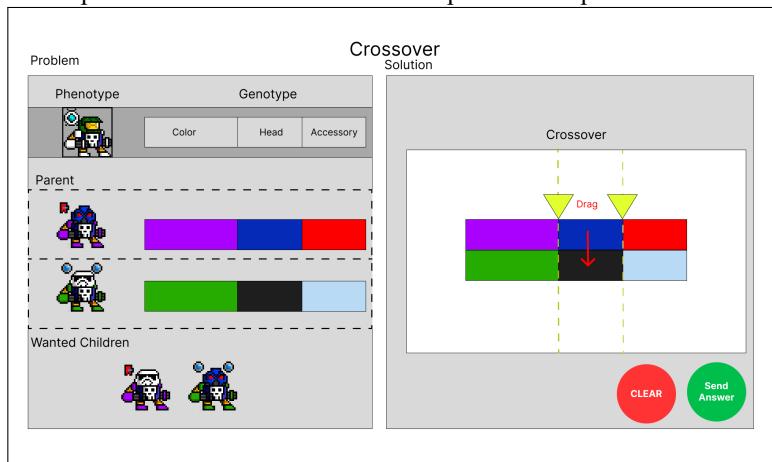


Figure C.38: Problem-solving Puzzle of Crossover 3 out of 4

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

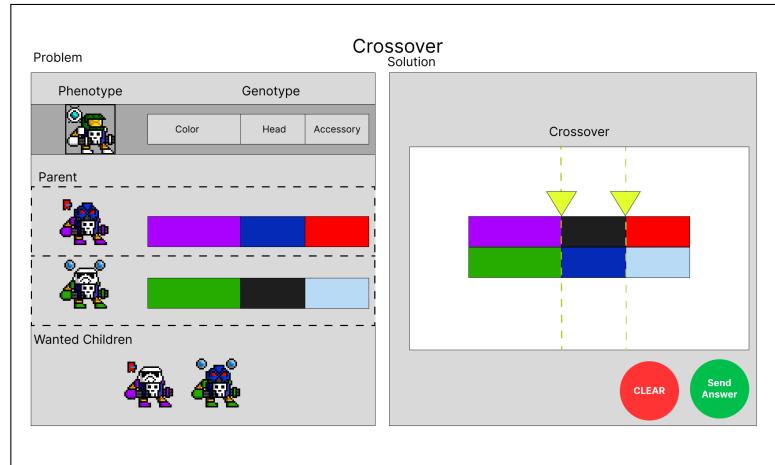


Figure C.39: Problem-solving Puzzle of Crossover 4 out of 4

The players click Send Answer to send the answer.

APPENDIX D
CHARACTER SPRITES

Monster 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.

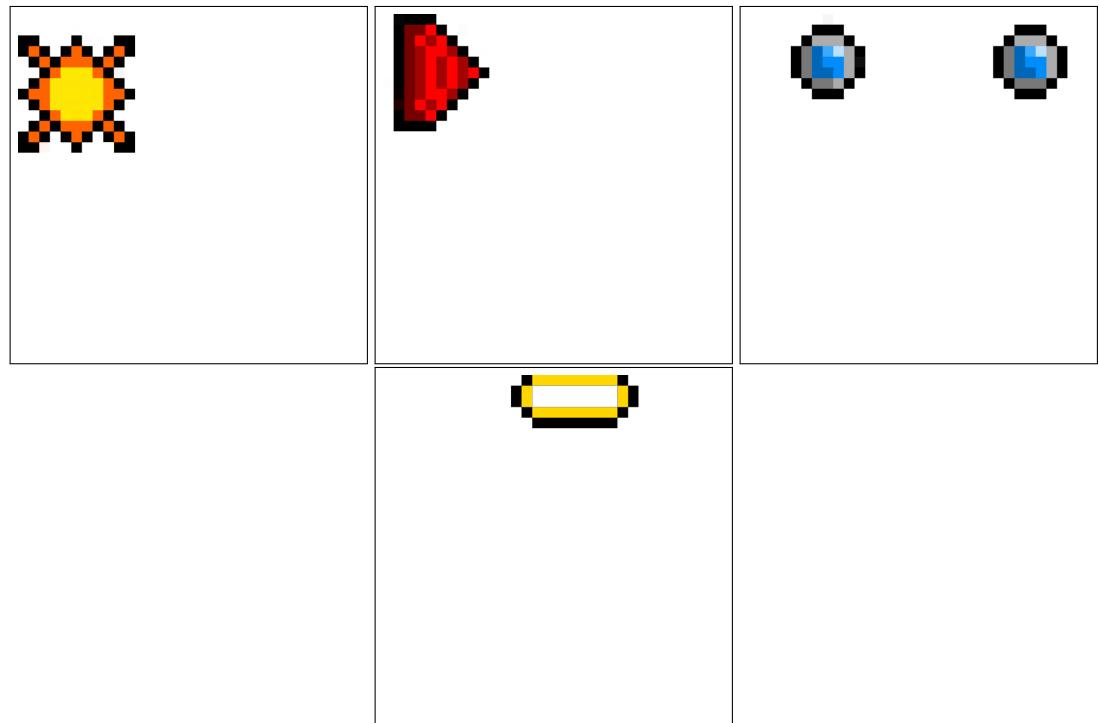
1. Head Variances



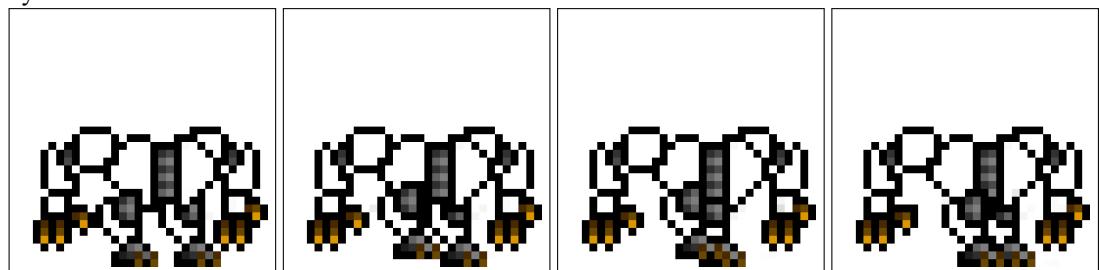


2. Accessory Variances





3. Body



4. Body with weapon use in Arena



5. Assembling Examples



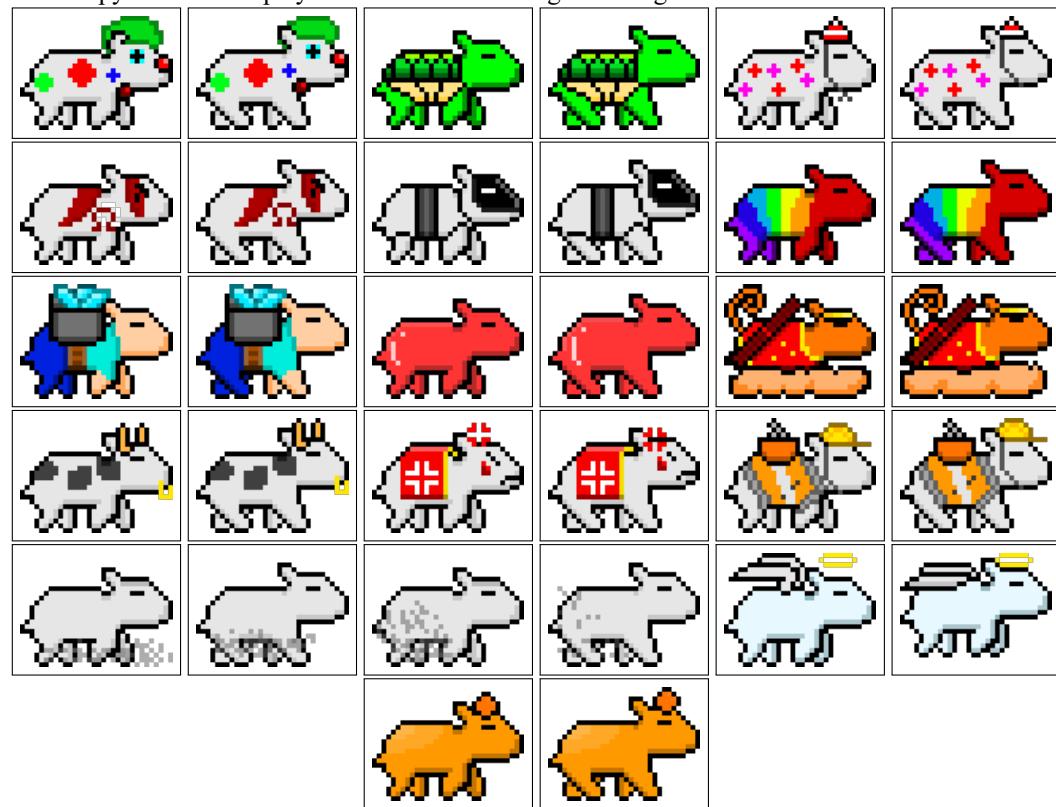
NPC Sprites

Sprites of NPC name “CatBot” use in the cutscene of the game to teach the players about the Genetic Algorithms and progress through the main quest of the game.



Capybara Sprites

Sprites of Capybara that the player can encounter throughout the game.



APPENDIX E
USABILITY TEST RESPONSES

Usability Test Responses

There are six questions in total in the usability test questionnaire. After conducting the usability test in the first week of February 2023, 13 testers responded to the questionnaire. The response to each question in the questionnaire will be listed.

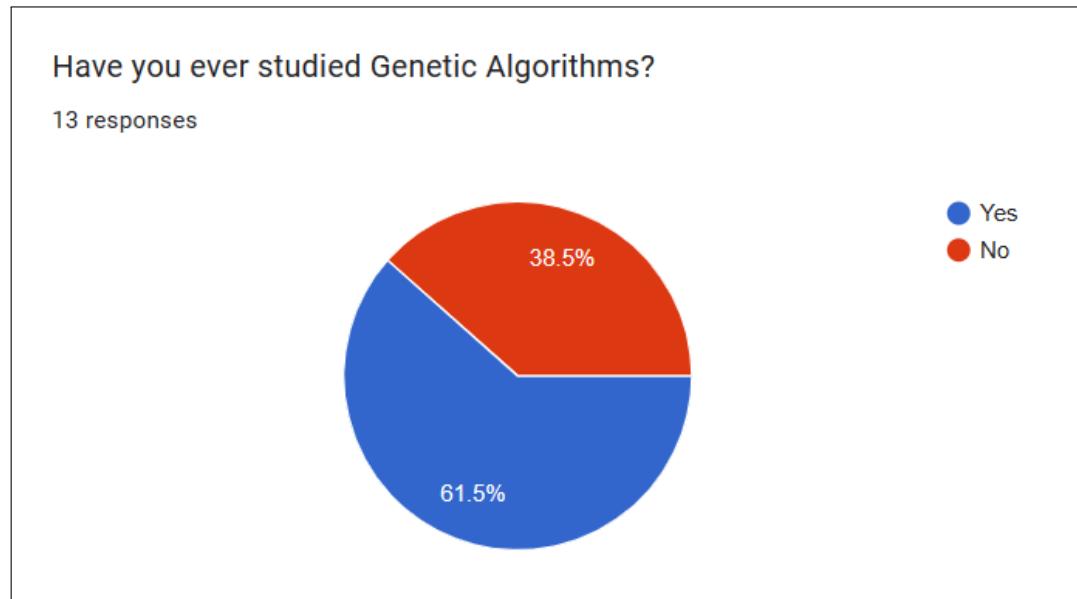


Figure E.1: Response to the usability question 1

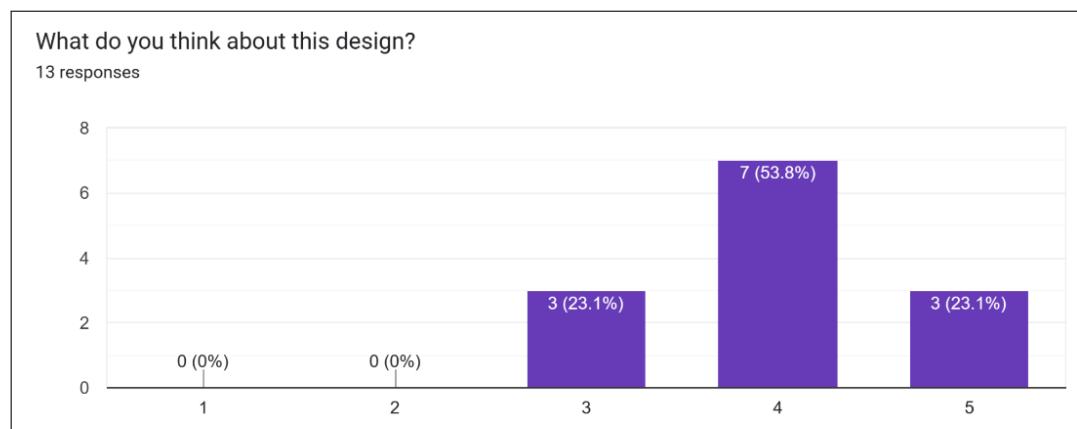


Figure E.2: Response to the usability question 2

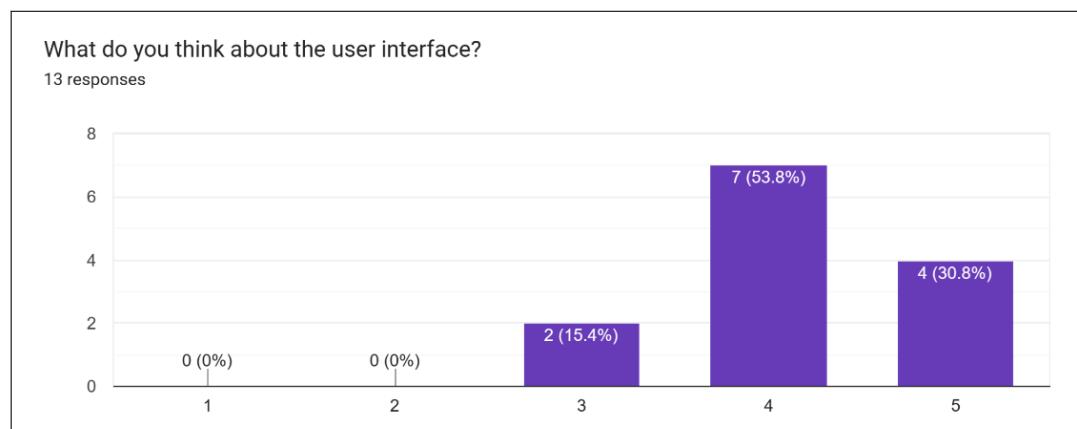


Figure E.3: Response to the usability question 3

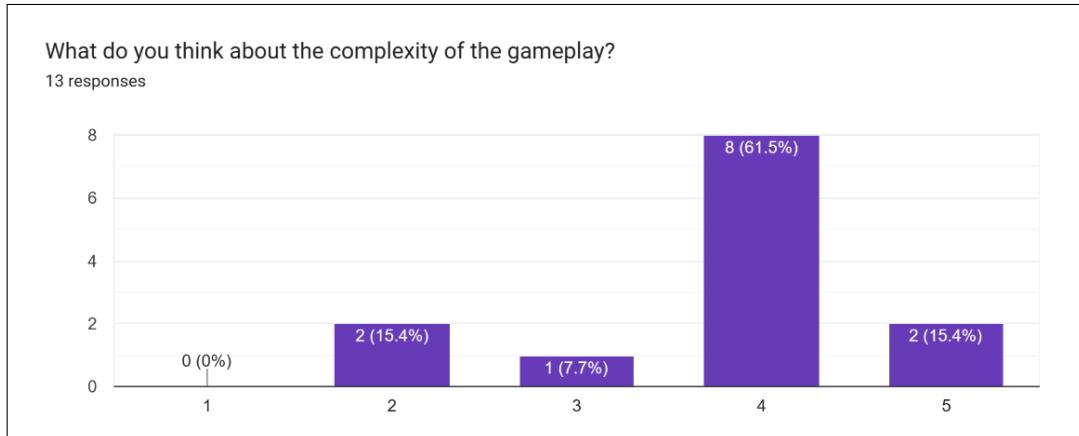


Figure E.4: Response to the usability question 4

Figure E.5 shows the raw response to the question. We created the choice for this question using a score of 1 representing too simple to a score of 5 representing too complicated. Since we prefer the response to be too simple over too complicated, we rearrange by reversing the score to make the higher score the more preferable option like the other question, resulting in Figure ??.

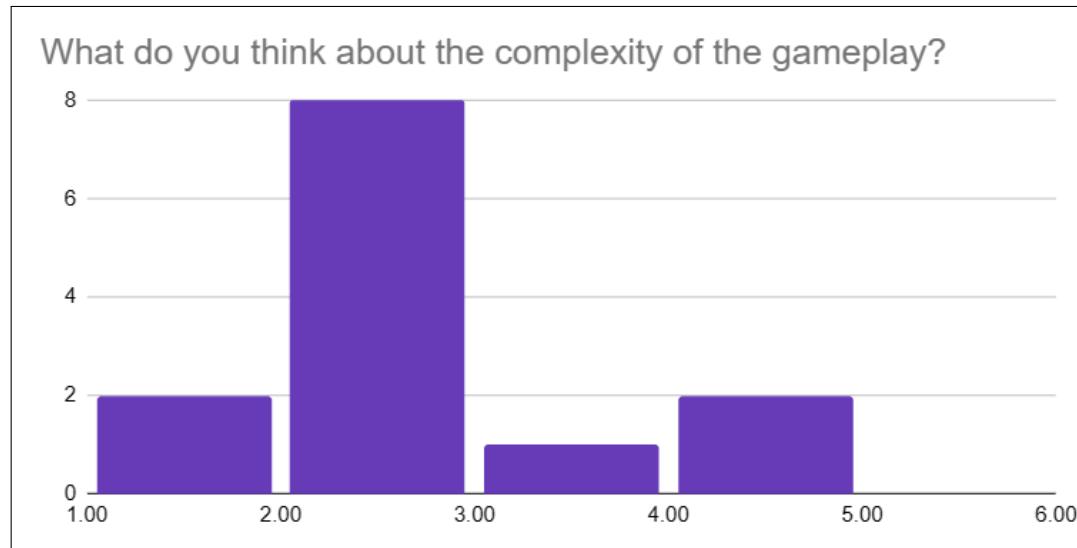


Figure E.5: Response to the usability question 4 after rearrangement

If you notice any issue that makes this game terrible, what is it?

4 responses

The puzzle is too complicated and too technical.

I don't think there are that much issue for now, the game mechanic especially genetic mechanic takes a bit of time to understand, I think adding a quick archive on some pages for a quick look would be a good idea. Like adding a genetic code sheet or sth.

About arena, quest, or shop, I think they're good already.

if i understand your game well so i think the main gameplay is farming robots and equipment then

What happen if factory break in the middle of skipping days

Figure E.6: Response to the usability question 5

If you could change one thing in this game, what would it be and why?

3 responses

-

icon อาจารของ mecha ตอนต่อสู้ไม่ค่อยโดดเด่น ดูคลอนข้างมาก (อาจเนื่องด้วยยังไม่fix final design) อาจจะบ้ายไปอีกข้างบนสุดแทน

Maybe robot ability can have synergy with other ability

Figure E.7: Response to the usability question 6

APPENDIX F
FARM AND FACTORY UNLOCK REQUIREMENTS

Farm and Factory Unlock Requirements

These are the unlock requirements of each facility which is used to control the game progression.

Table F.1: Monster Farm Unlock Requirements

Monster Farm	Unlock Requirements
Farm 1	1. Obtain Jigsaws 1.1. Basic Bio (Copper) 1.2. Basic Selection (Copper) 1.3. Basic Crossover (Copper) 1..4. Improvement Techniques (Copper) 4.5. Single-point Crossover (Silver) 2. Money
Farm 2	1. Unlock Factory 2 2. Money
Farm 3	1. Unlock Factory 3 2. Money

Table F.2: Weapon Factory Unlock Requirements

Weapon Factory	Unlock Requirements
Factory 1	1. Unlock Farm 1 2. Obtain Jigsaws 2.1. Standard KP (Silver) 2.2. Two-point Crossover (Silver) 3. Money
Factory 2	1. Unlock Factory 1 2. Obtain Jigsaws 2.1. Multidimensional KP (Silver) 2.2. Tournament-based Selection (Silver) 3. Money
Factory 3	1. Unlock Factory 2 2. Obtain Jigsaws 2.1. Multiple KP (Silver) 2.2. Roulette Wheel Selection (Silver) 3. Money
Factory 4	1. Unlock Factory 3 2. Obtain Jigsaws 2.1. Rank-based Selection (Silver) 3. Money

APPENDIX G
USER SATISFACTION SURVEY RESPONSES

User Satisfaction Survey Responses

This is the full result from the User Satisfaction Survey on 17th May 2023 with 92 participants.

Questions 1 to 5 are designed to collect tester data in order to help us understand the demographics of the testers and how they may impact the survey results. Specifically, questions 3 to 5 are intended for university students participating as testers.

Three questions involve personal data and will not be shown fully here as follows:

- Question 1: What is your full name?
- Question 3: What is your student ID? (If you aren't currently study in KMUTT, please fill "-")
- Question 6: Upload your picture of Hall of Intelligence

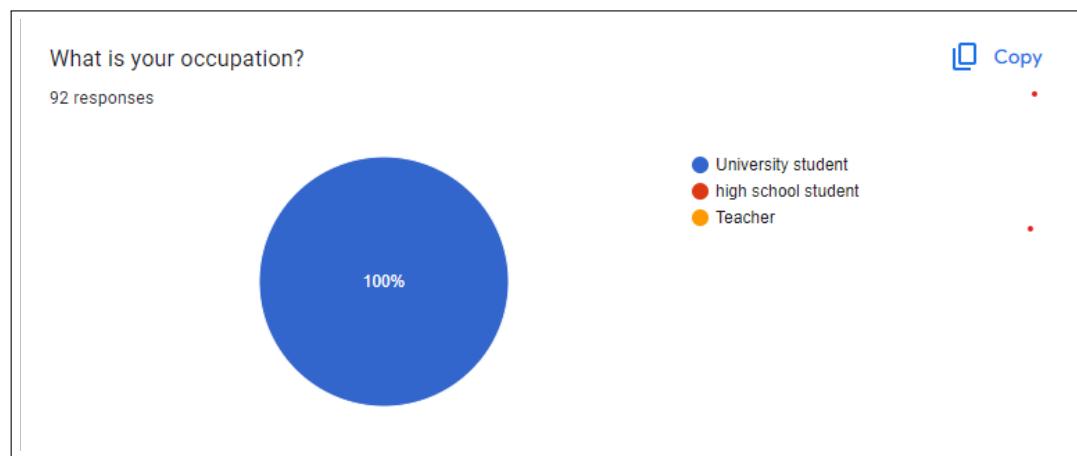


Figure G.1: Response to the satisfaction question 2

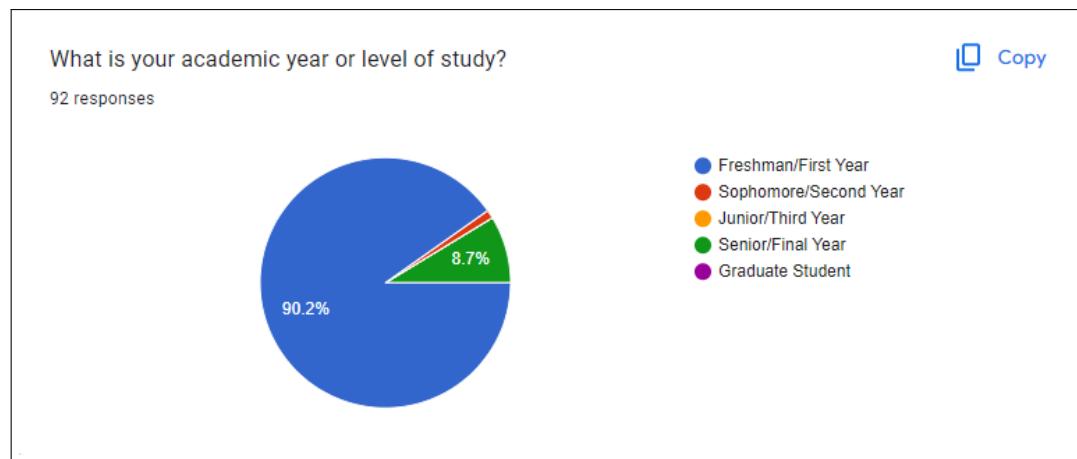


Figure G.2: Response to the satisfaction question 4

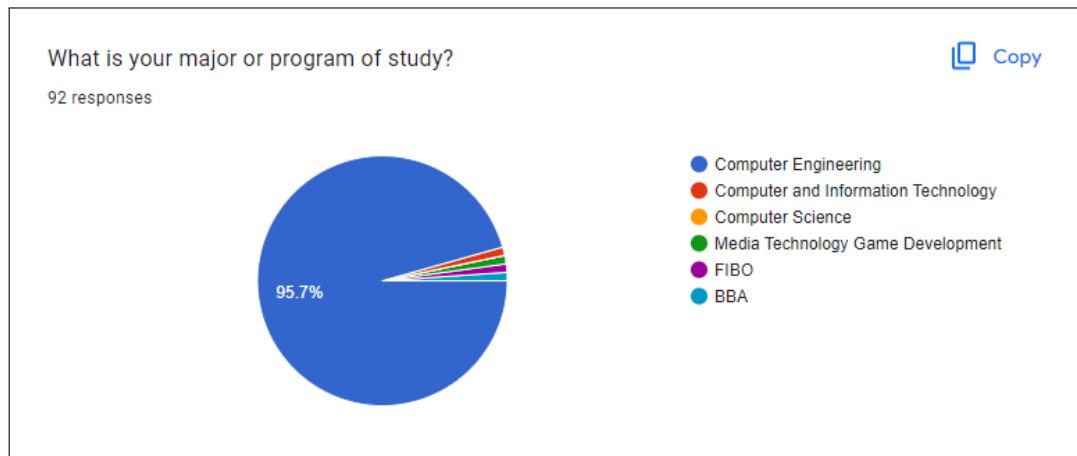


Figure G.3: Response to the satisfaction question 5

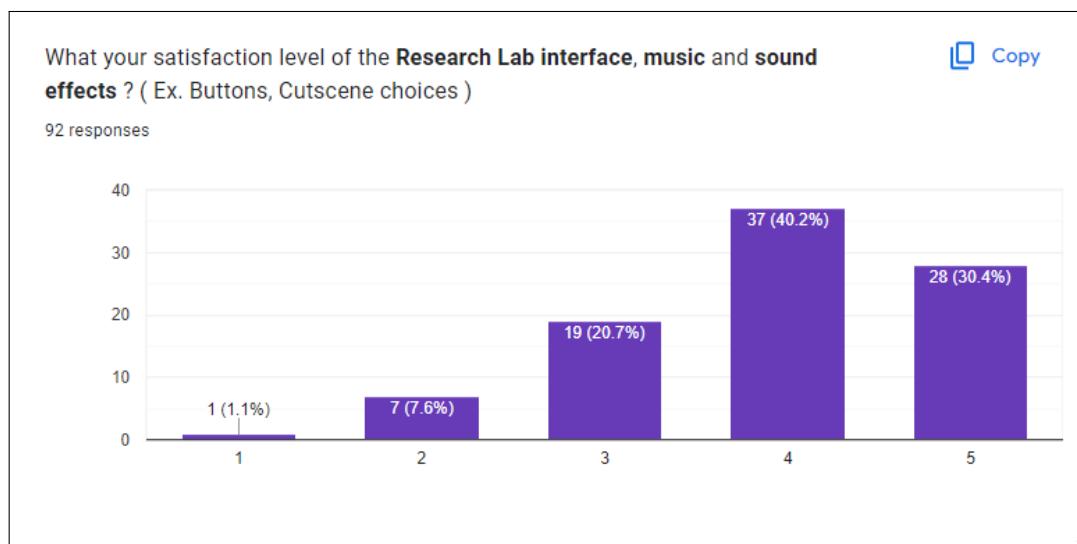


Figure G.4: Response to the satisfaction question 7

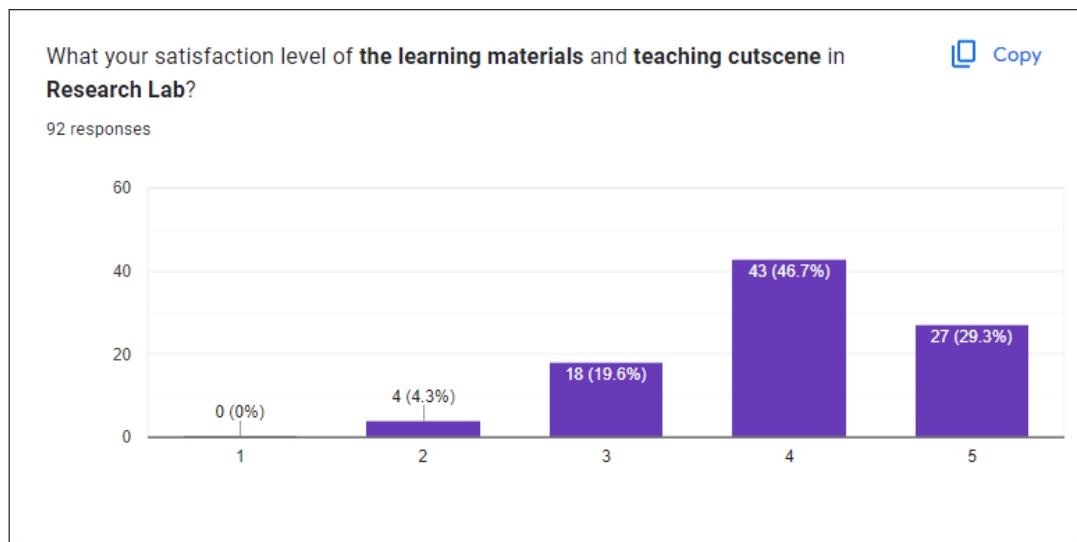


Figure G.5: Response to the satisfaction question 8

Do you have any suggestion/problem in the learning materials and teaching cutscene?	
-	อย่างไรก็ตามเบ็ดเตล็ดในพาร์ทนี้
No	Fun and enjoy playing it
add menu option. the sound volume too high.	ต้องแต่ละรอบมาก
ท่า เป็น animation น่าสนใจ	picture to small and not that interesting
I think the cutscene of topic can show when player do an activity in hall of intelligence. I think it can make player focus on what topic they need to know for playing those part of jigsaw.	no
OK	it's ok
MORE ANIMATION	ค่อนข้างไปความรู้ที่จำเพาะเฉพาะ เช่นบางอย่างไม่ได้มีในชุดของที่ให้เรียน จึงต้องวิเคราะห์กับคนที่มีความรู้ในเรื่องนี้หรือต้องไปเรียนรู้
It would be good if there are other media in the conversation, or even full-CG in it since it was based on visual novel.	Actually there should be some instruction during the game .But it's okay to just trial and error till I get the solution right. (Maybe I'm just skipping it).
I think it should have option to decrease volume of music and sound effects.	อยากให้สามารถเลือกและตั้งค่ามากขึ้น
Reading the slides after the cutscenes works better for me	Research Lab interface is quite confusing especially sections and subsections.

Figure G.6: Response to the satisfaction question 9

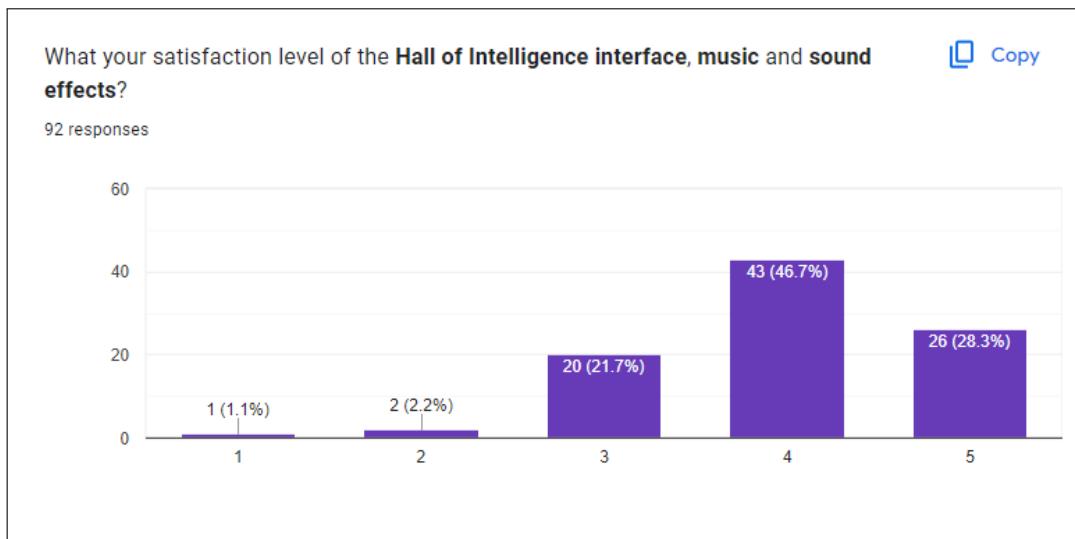


Figure G.7: Response to the satisfaction question 10

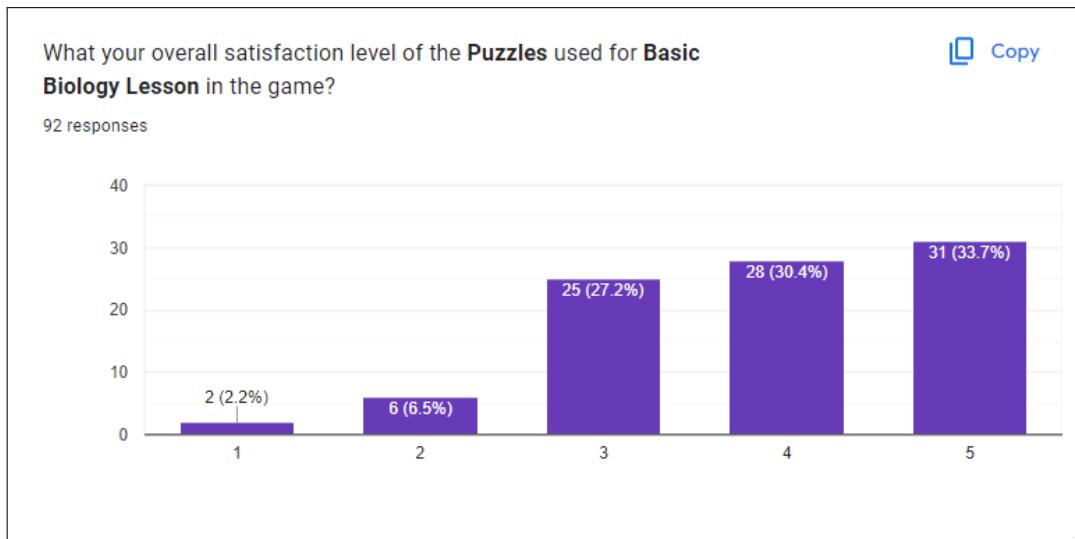


Figure G.8: Response to the satisfaction question 11

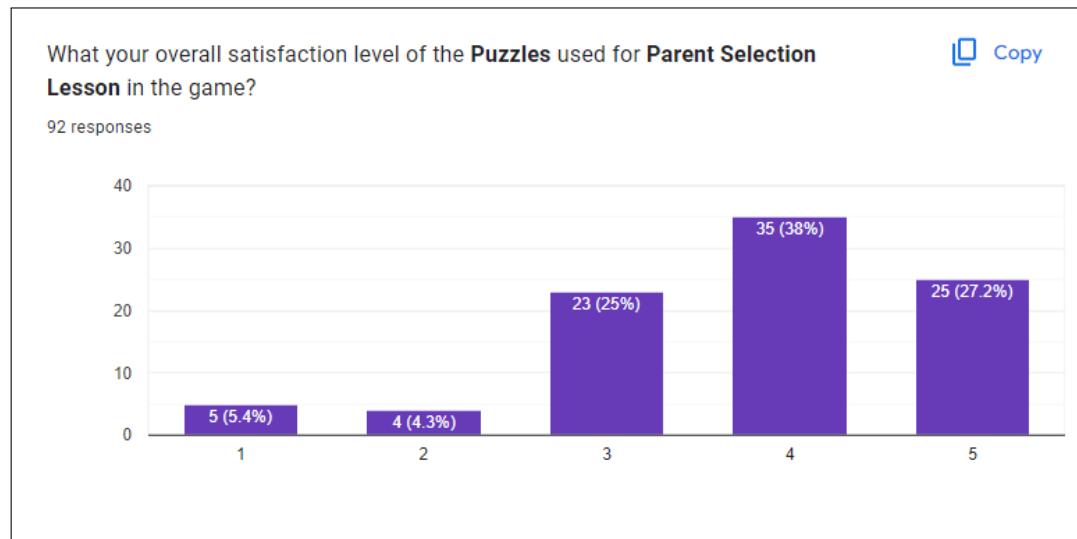


Figure G.9: Response to the satisfaction question 12

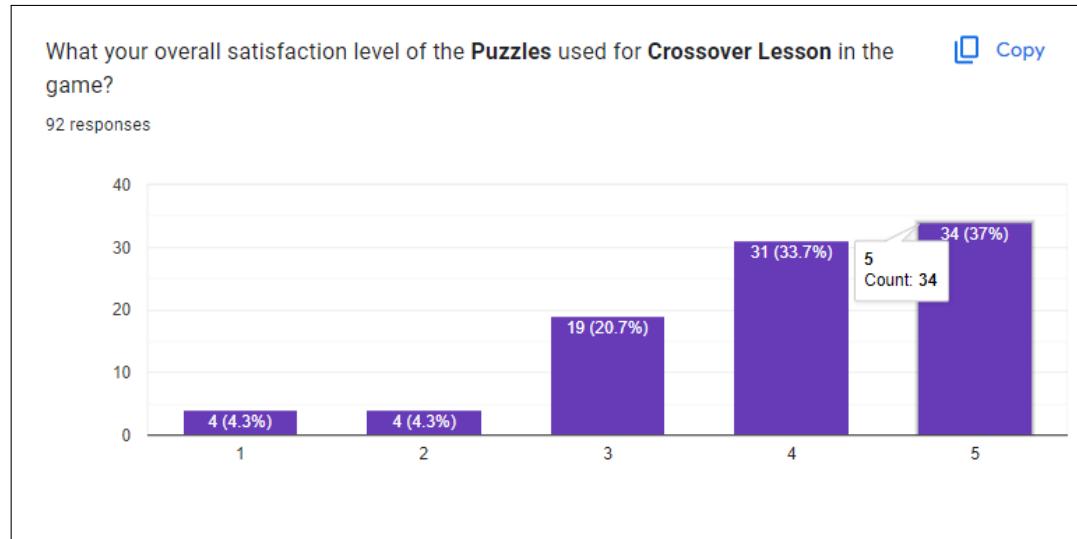


Figure G.10: Response to the satisfaction question 13

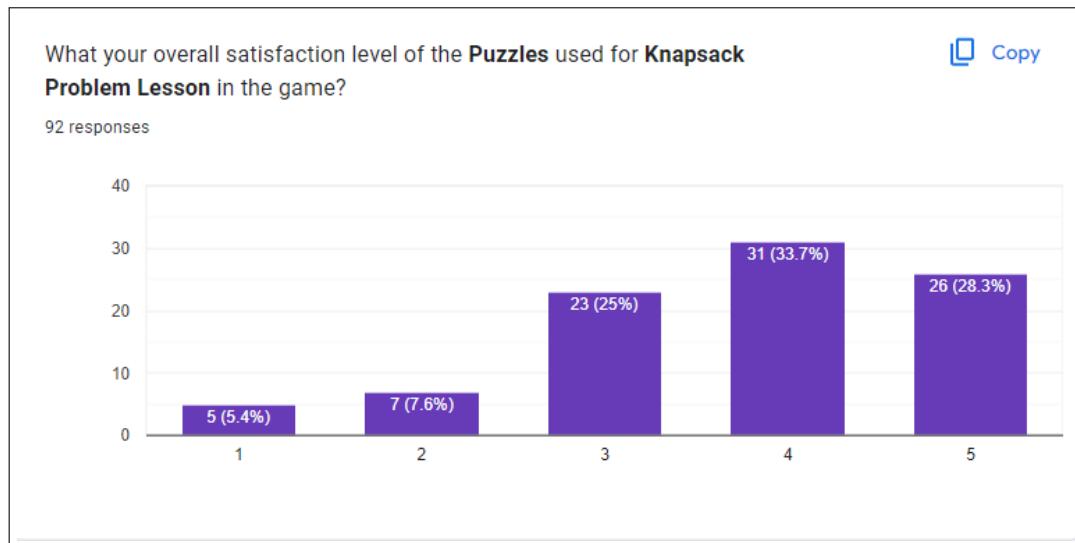


Figure G.11: Response to the satisfaction question 14

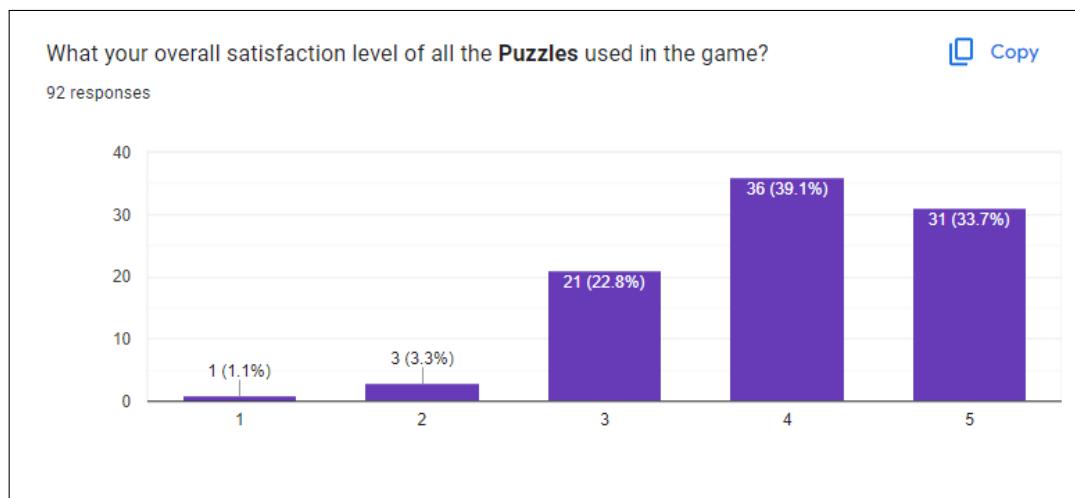


Figure G.12: Response to the satisfaction question 15

Do you have any suggestion/problem in Puzzles used for testing player in Hall of Intelligence ?	
-	In Stage 3, there is a bug.
Quest ที่ไม่ใช้ชื่อของมนุษย์ต่างๆ ก็สามารถทำได้ในช่วง detail Goal ได้ซึ่กันเอง	It's hard to click on something I want in there
For puzzle should have a tutorial level to demonstrate the function of each puzzle first. For question when the result window appears the text behind is unreadable.	ข้อ 4 มันมีดีรับ
ระบบ resolution ของ UI ไฟล์เก็บไว้หน้า research lab มี UI ย้อน ฟื้นฟูต่อ	ไม่มีดีรับ
If failed once in knapsacks problem, when re-entering, all blocks will become 0 bits and old block will be in the same spots before reset.	There are some bug in Knapsack Problem Puzzle scene that all of the chromosome preset are red
I think it already great that you give all the method in one play, it can measure that how much player know in that topic?	Some UI bug in the web player
Hard to understand and use too much knowledge	เข้าใจยากกันไป เพราะเดียวจะลองทุกอย่างแล้วดูว่าอะไรจะดี แต่ไม่ได้มีมาบอก และไม่มีบุคคลสอนให้ด้วยการกระทำการสอนหน้าเพื่อเราดูกันได้แล้วหนึ่งเดียวต้อง自行摸索
Instruction very unclear	No
A bit glitch	no
Some randomize problem can't be solved. Need to limit the possibilities of getting unsolveable problem.	It should be abortable or at least should have some warning sign such as "Entering this test must be finished before going out"
it's good	(การจะออกจากปัญหานี้ต้องวิเคราะห์ความคิดของคนที่ต้องทำแบบทดสอบให้จบก่อนเท่านั้นถึงจะสามารถนั่งแบบทดสอบได้)
description of the question was confused after reading it.	it's too hard to understand what i should do
I think at the first its look interesting but when I've passed first jigsaw and see that it has many assessment to be completed, I felt bored at that moment.	เมื่อมาสู่ห้องทดลองที่มีความรู้ในลักษณะนี้
I misunderstand in problem. The puzzle is not hard but problem is hard to understand, but not much	If this game doesn't have a bug, That's great.
Please add a skip option for the spinning wheel animation.	No I'm okay with that.
some of lesson description maybe not clear and hard to understand. Puzzle are literally hard to understand process for first time. Knapsack section have a bug when enter to the puzzle	The instruction should be better and should be cleared that what player can do and can not.
	puzzle ของ Knapsack หรือมีมีดีรับ เวลาเล่นไม่ยอมเลือกตัวที่ต้องการได้เลย
	ในการทดลอง gold สรุปหลังมีการทดสอบเบิกบานต่อจะก็ finish ก่อนที่จะให้ item preset ก่อนมานี้เป็นปกตินะครับ

Figure G.13: Response to the satisfaction question 16

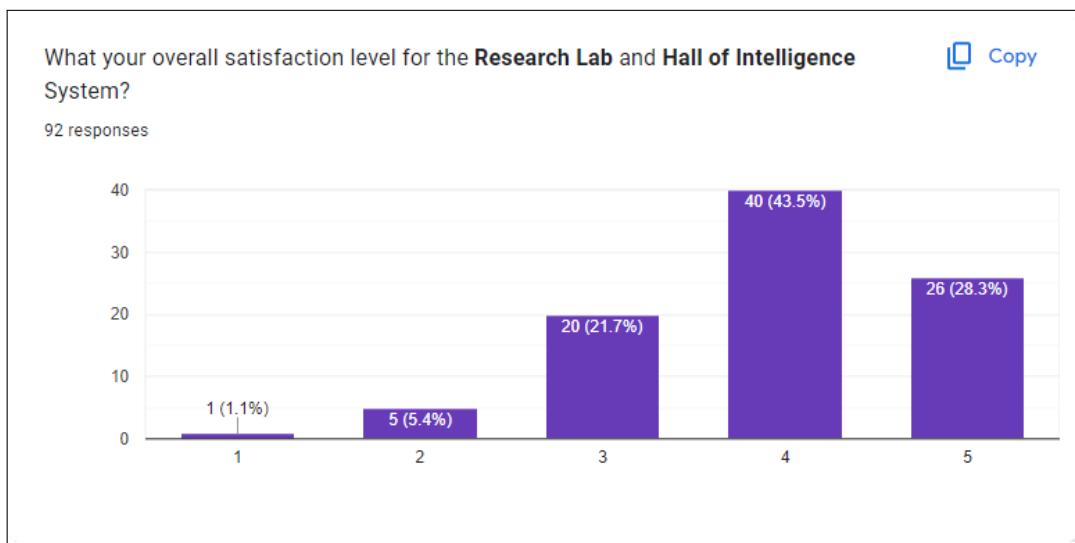


Figure G.14: Response to the satisfaction question 17

Do you have any suggestion for the Research Lab and Hall of Intelligence system of the game?	
42 responses	
-	Move this function to main menu.
เปลี่ยน Hall Of Intelligence ให้มหานคร ต่อไปนี้ research lab ก็จะดีมาก	ไม่รู้ว่าจะคลิกชิ้นๆ ได้หรือเปล่า
Change Hall of Intelligence system icon to something else because now it looks like achievement or status menu. It doesn't seem like a quiz to proceed.	ไม่รู้ว่าจะคลิกชิ้นๆ ได้หรือเปล่า
The trophy icon makes (me) misunderstand that it is achievement.	ไม่ใช่
Fix some bug I can't get any monster than I can't play arena and some kapibara that I was catch don't show if I don't get cursor on it	ไม่ใช่
Instruction very unclear	อ่านยาก
I can't breed my monster because of bug	nothing
More clean interface would be good (eg. dropdown from sub-category or different color when entering sub-category)	ก็โอเค
I prefer reading the slide and doing minigame to learn	ก็โอเค

Figure G.15: Response to the satisfaction question 18

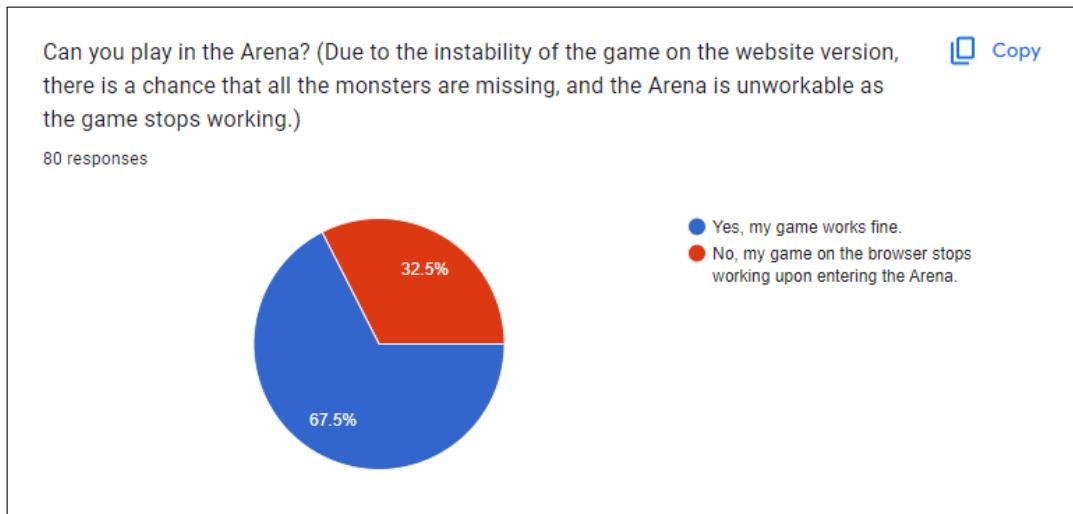


Figure G.16: Response to the satisfaction question 19

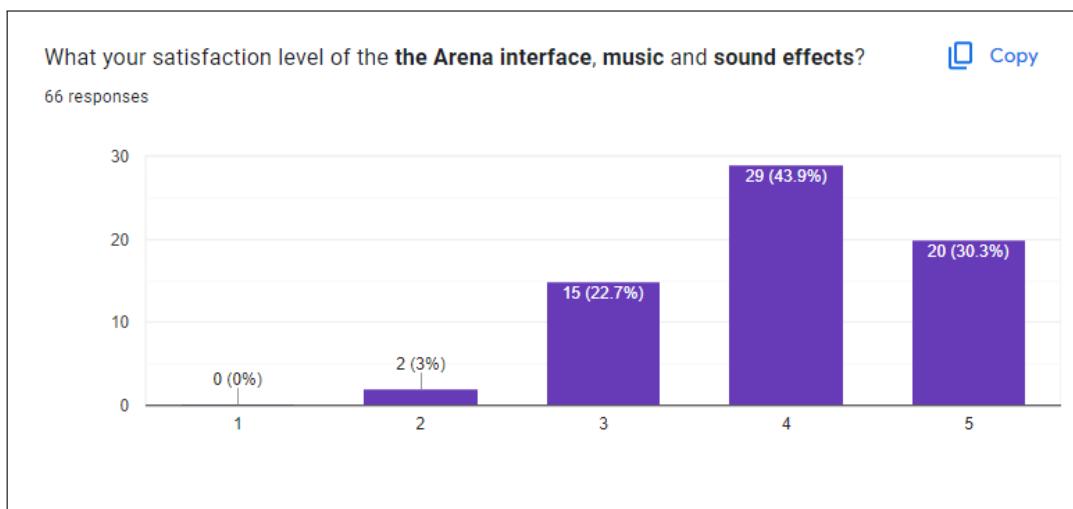


Figure G.17: Response to the satisfaction question 20

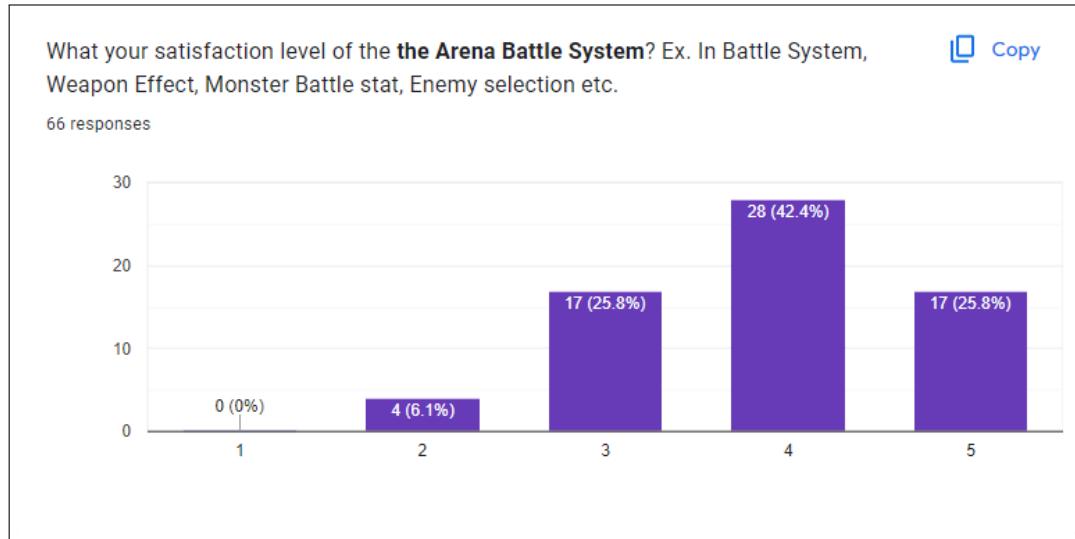


Figure G.18: Response to the satisfaction question 21

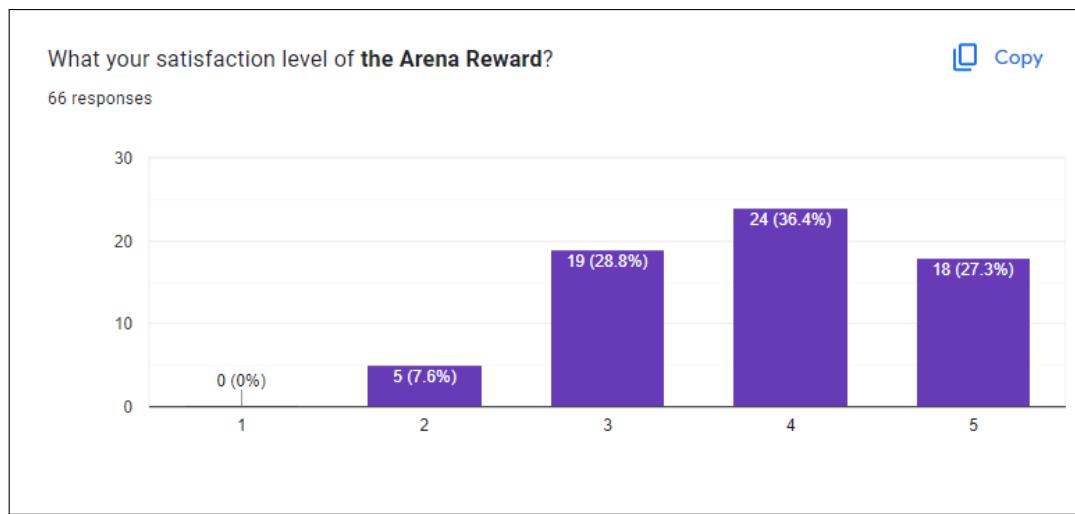


Figure G.19: Response to the satisfaction question 22

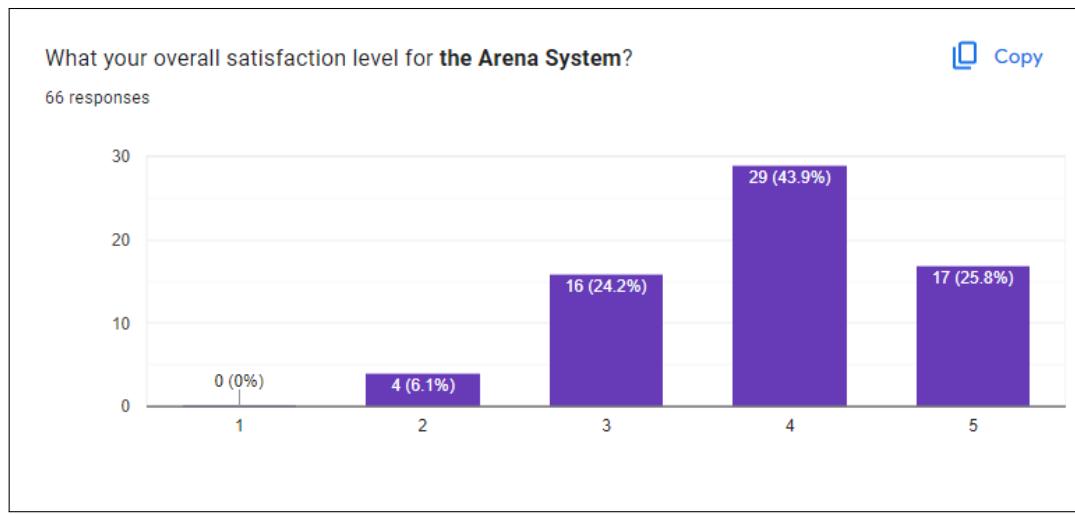


Figure G.20: Response to the satisfaction question 23

Do you have any suggestion for **the Arena System** of the game?

28 responses

-

Should be a tutorial battle first to demonstrate how the arena works.

Somehow the character refuse to do anything and the blue gauge is dead while the skills gauge is alive and do nothing upon activating it.

I can't play it because I can't get any monster I buy 6 monster but I lose my money for nothing but my friend really fun with arena

Instruction very unclear hard yo understand

more skills please

The fighting animation is slow for me.

found UI bug in the web player

no

ว้าสุดยอด

nothing

Figure G.21: Response to the satisfaction question 24

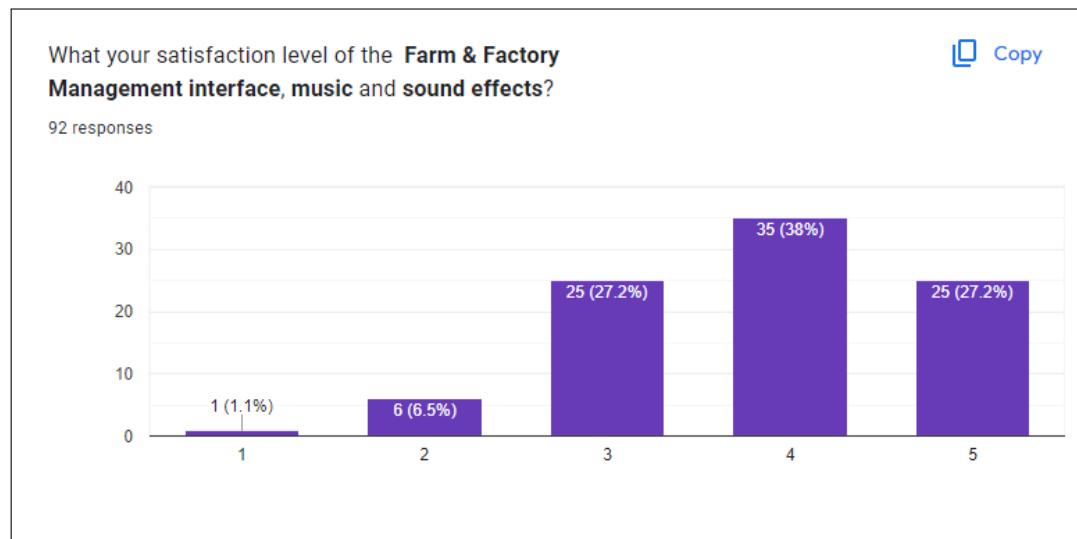


Figure G.22: Response to the satisfaction question 25

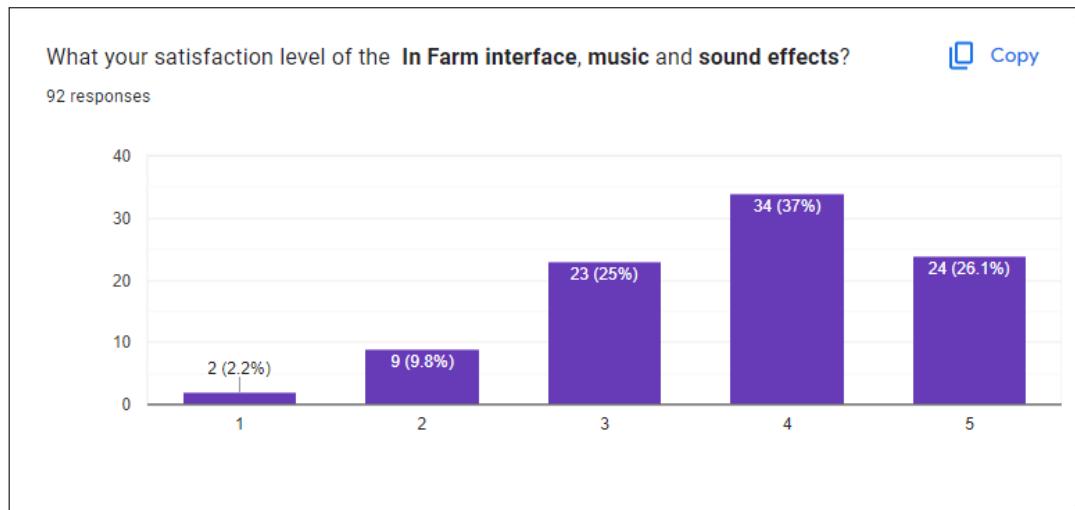


Figure G.23: Response to the satisfaction question 26

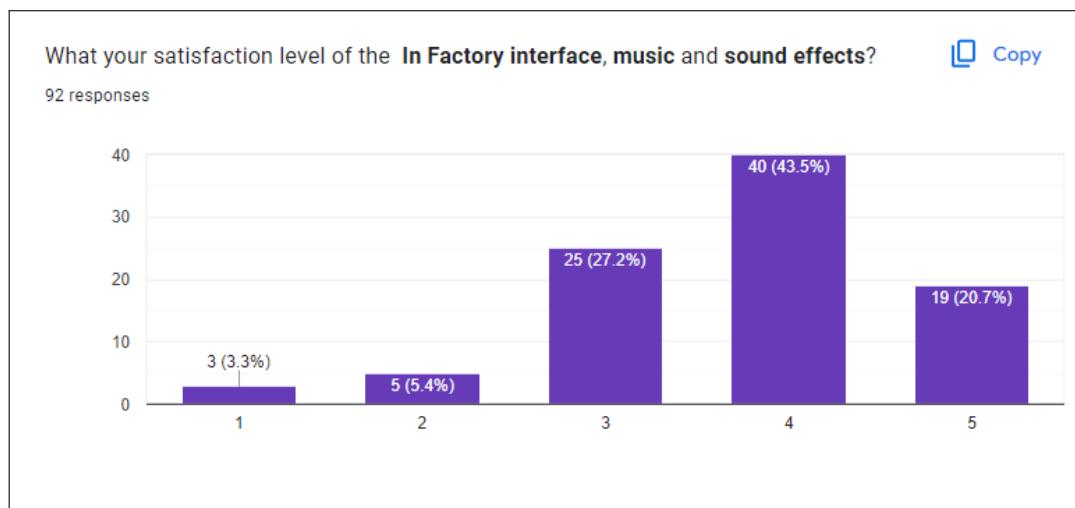


Figure G.24: Response to the satisfaction question 27

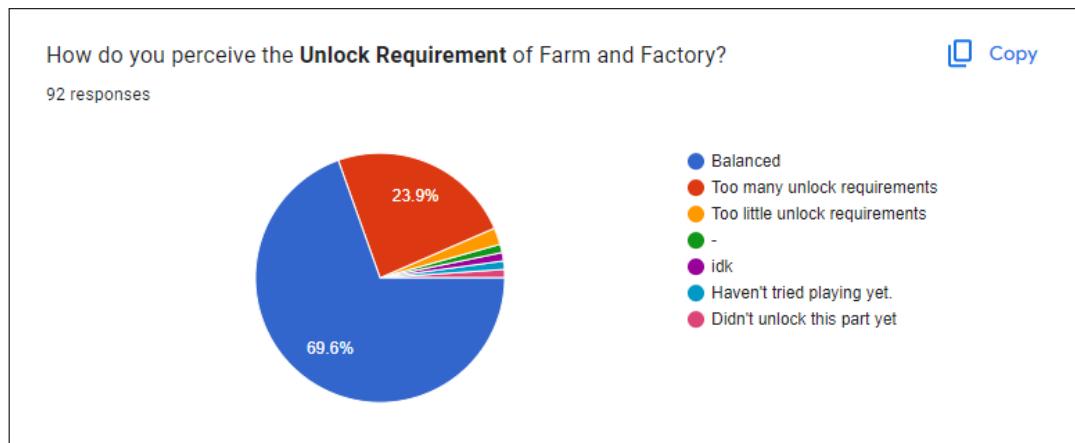


Figure G.25: Response to the satisfaction question 28

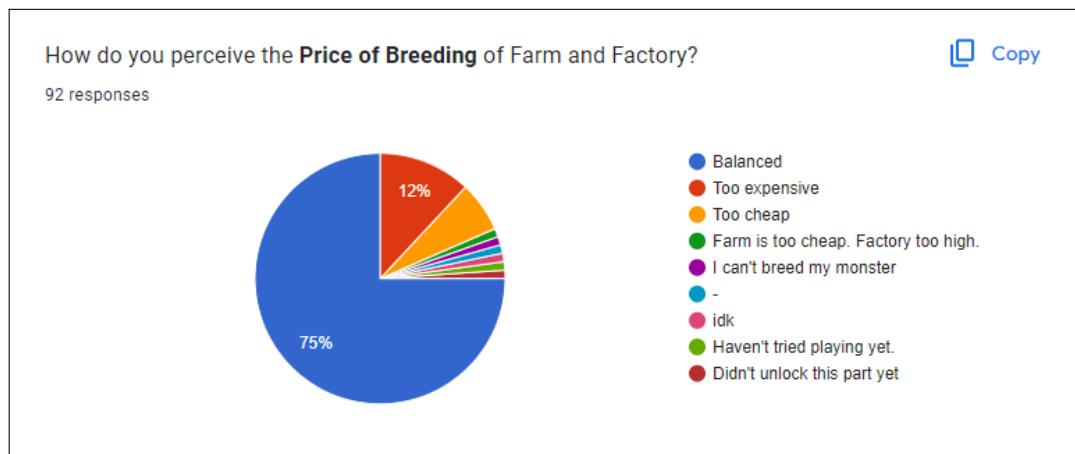


Figure G.26: Response to the satisfaction question 29

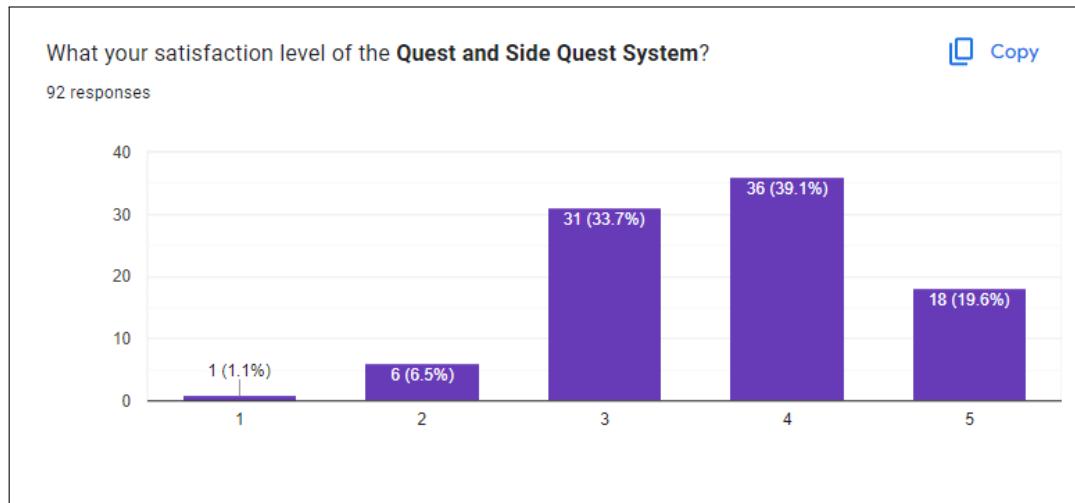


Figure G.27: Response to the satisfaction question 30

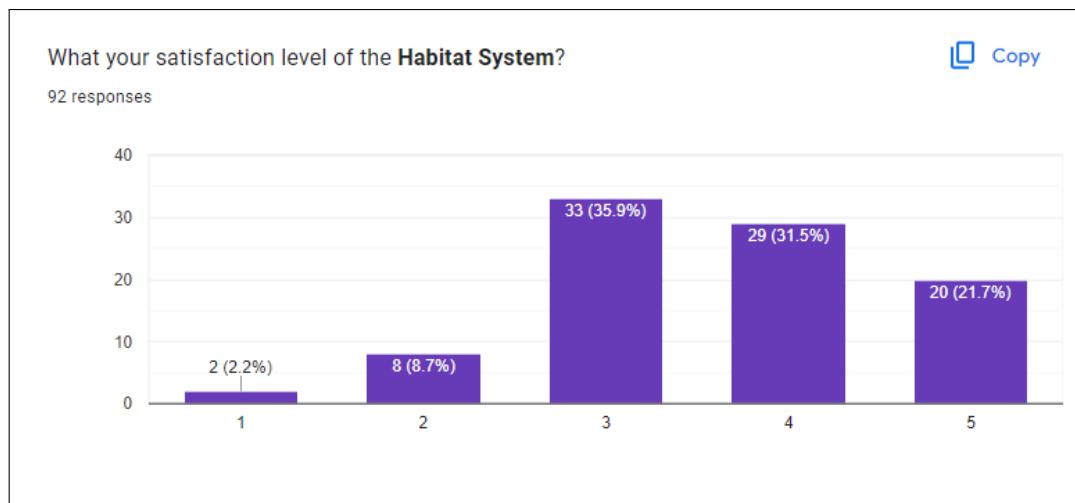


Figure G.28: Response to the satisfaction question 31

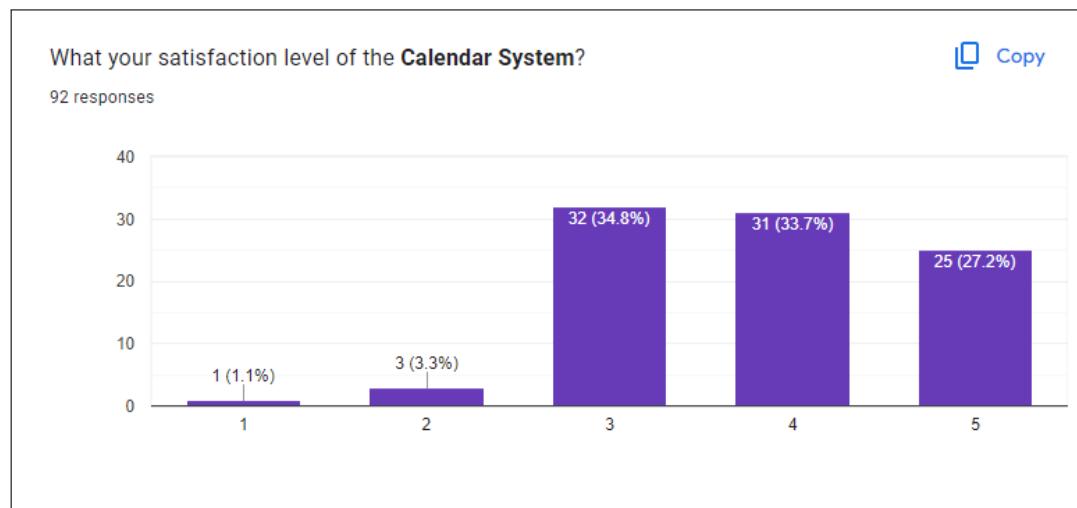


Figure G.29: Response to the satisfaction question 32

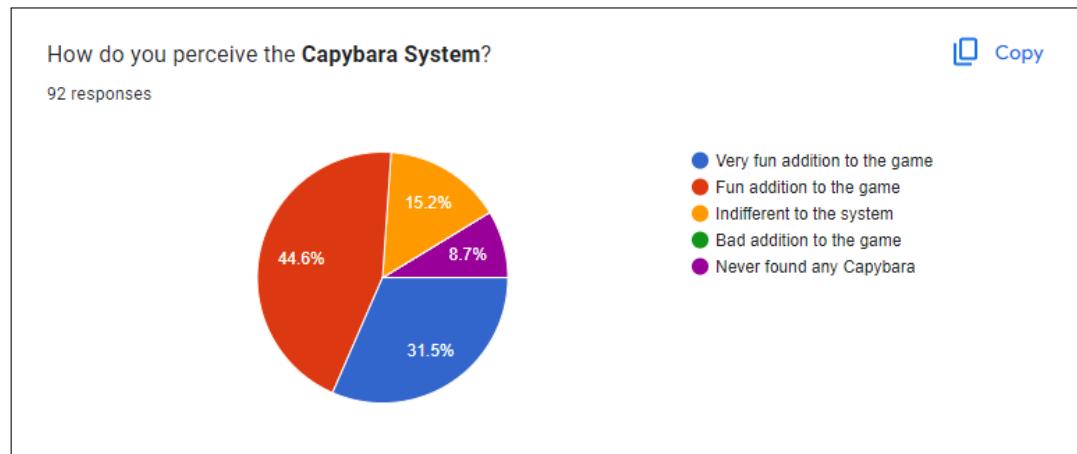


Figure G.30: Response to the satisfaction question 33

Do you have any suggestion/problem for The Game side system of the game?		
38 responses		
-	I love capybara	
should have guide for each system because there are no indicator to each system and player have no idea what to do next.	First time, I didn't know I needed to continuously click to obtain Capybara.	
หลักที่ UI แห้งและต้องซ่างขายา ให้รู้เวลาในการทำความเข้าใจ	Bug in Hall of fame The fourth one	
I think the quest need to explain more clearly that the player can only receive one quest in each day.	ไม่เลื่อน	
Factory and farm is hard to understand but this game got 100/10 score form me because the capybara	-	
My Calendar bug out and it stay at 6/01/2700 I think it because I do all Hall of Intelligence than do the quest	ui bug and stop running in the web player	
How can I breed my monster without error? I want to train my dragon. I wanna be Hiccup. Please debug this problem so I can be a dragon trainer.	โค้ดรวมไว้แล้วจะรีบวนท่าให้ลืมกันแน่นใจรึปั้น	
Please give this game gacha, so everyone can be a whale and maybe dating with Catbot. The gacha can be presents for increasing relationship rank with Catbot and we can have full-CG with Catbot being cute <3	no	
I can't interact with the application after the third day like it, my browser stop working.	This game has some bugs. And I can't skip the 6th day. Didn't know that you need to go to the next day for the next quest to pop up	

Figure G.31: Response to the satisfaction question 34

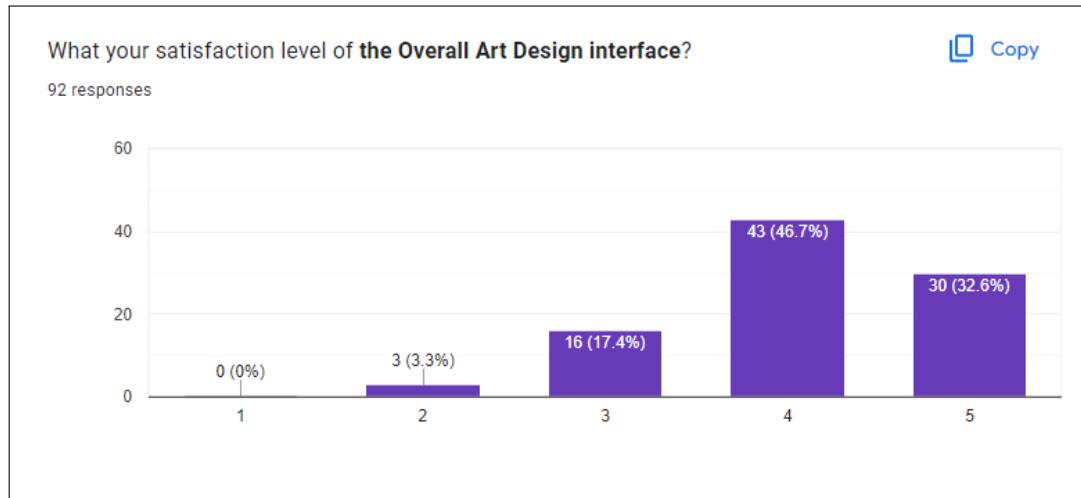


Figure G.32: Response to the satisfaction question 35

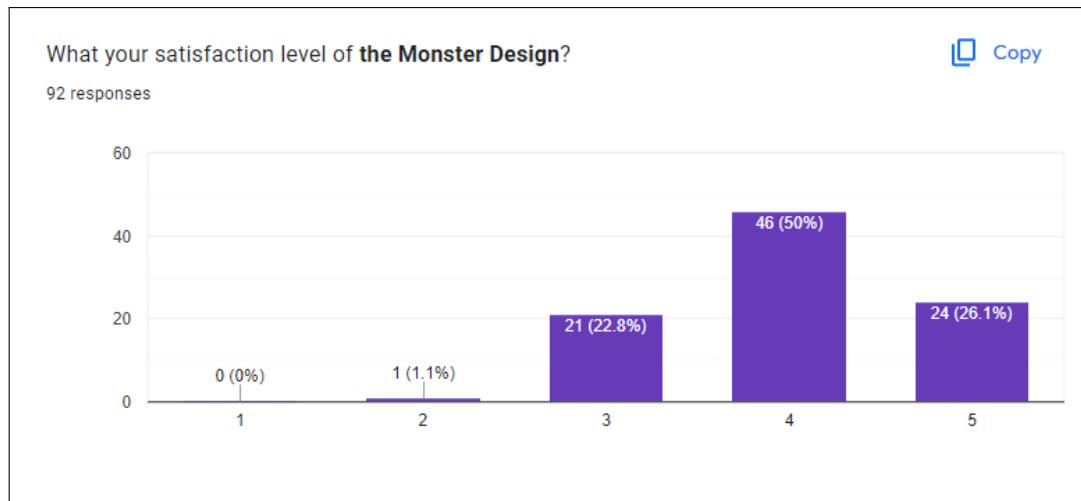


Figure G.33: Response to the satisfaction question 36

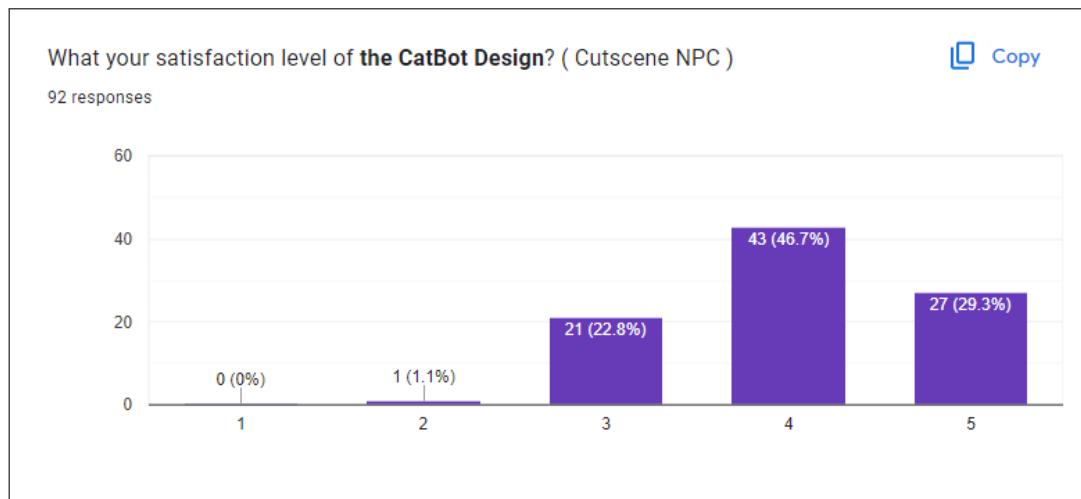


Figure G.34: Response to the satisfaction question 37

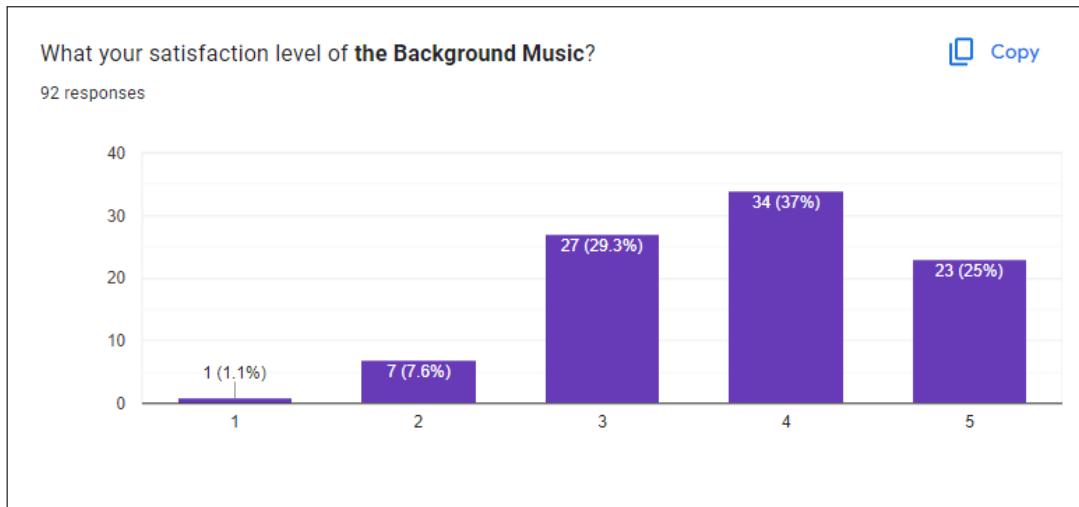


Figure G.35: Response to the satisfaction question 38

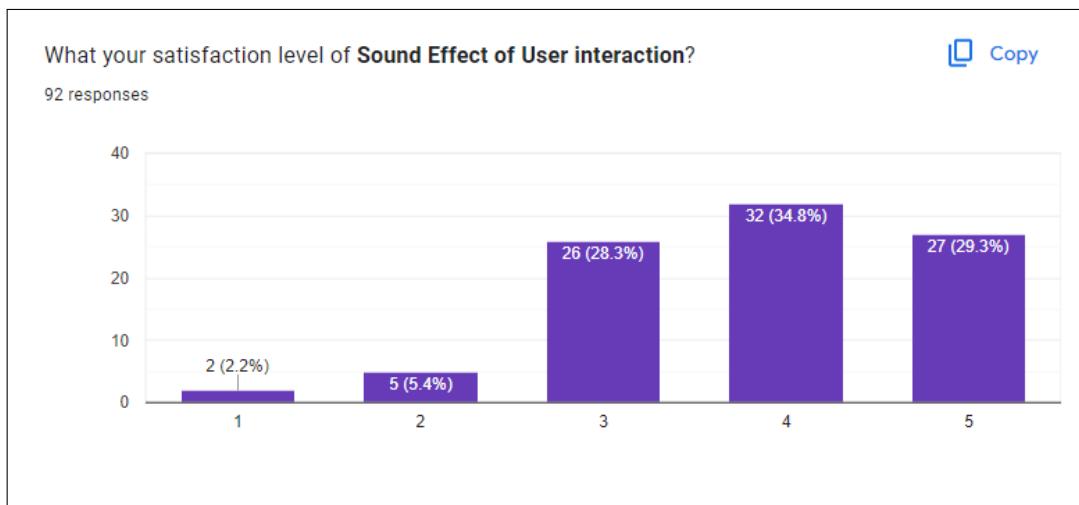


Figure G.36: Response to the satisfaction question 39

Do you have any suggestion/problem for The Art and Media of the game?

35 responses

-	Sound is too loud
art is unique but hard to grasp what its mean in the game. Music is super annoying because it's too loud and there is no option to low the volume.	Love capybara
มา พิเศษที่ไม่มากวนะ	Good
Music too loud. (For me)	ไม่มีคือรุน
The theme of this game is already attach in my opinion, but the catBot in cutscene need to have more action and background.	เมื่อจะจากเป็นภาพเบนรัมบิล โอลด์รัมบินเมื่อไรภาพก็ต้องมีความน่าคราฟและต้องใช้เวลาอ่านเพื่อเข้าใจมาก
PLS add volumen button	หัน
I like CatBot.	ดีๆ
I love capybara	That's great. I love 8 bit model.
	Full-CG with CatBot is needed, he needs to see the outside world and have a beach party with player. (Joking aside, Pixel artists can try other palettes color to make the games more refined and cute. Other environments and relationship between characters are needed to make the game look less forced.)

Figure G.37: Response to the satisfaction question 40

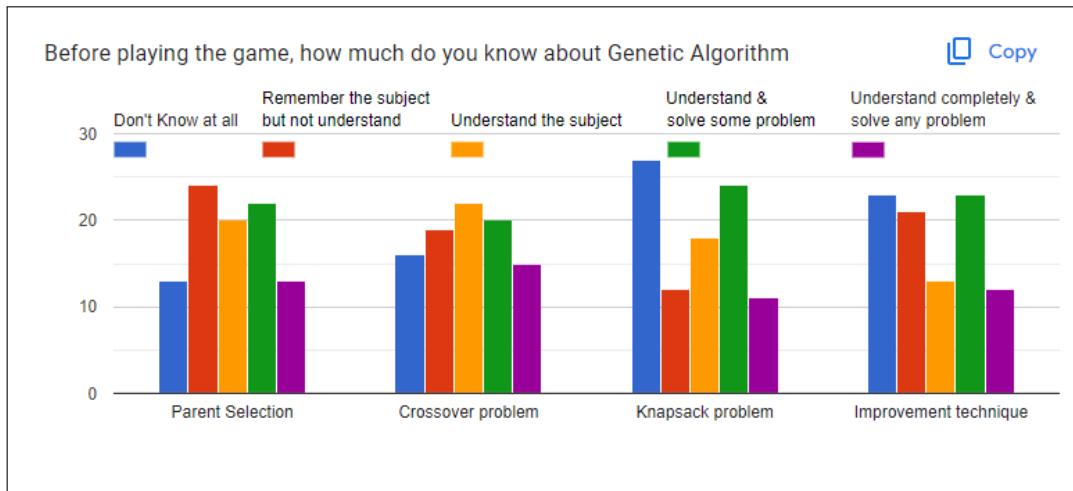


Figure G.38: Response to the satisfaction question 41

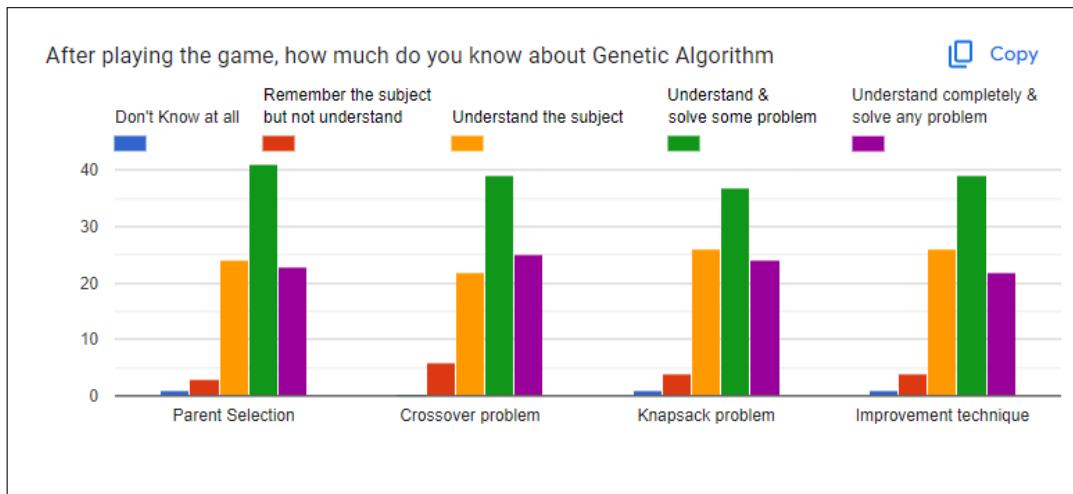


Figure G.39: Response to the satisfaction question 42

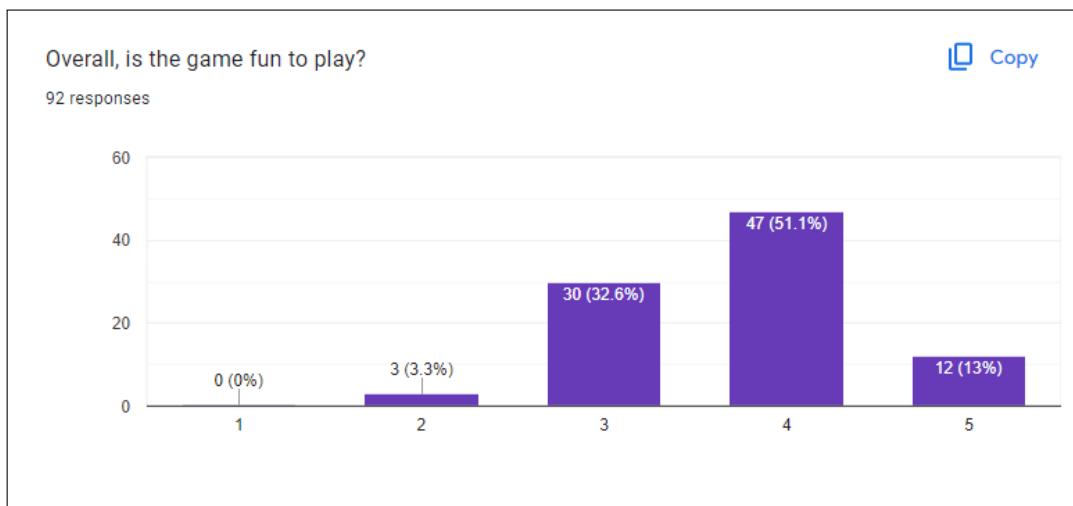


Figure G.40: Response to the satisfaction question 43

Do you encountered any problem while playing the game?
39 responses

-	A few bug in the game
Factory ที่มีการ Breed และใช้งานยาก ดู (i) และต้องใช้เวลาจากว่างและเสียเวลา resize the window. text is unreadable at some point. stuck and don't know where to go. can't low the volume or turn off.	Nothing
"UI ชื่อน	"ไม่มีปัญหา
Knapsack retrying.	There are some few bug.
yes, I encounter the tournament-based selection the most because I miss a concept in grouping. (I played it 5 times)	the ui not scale with the screen when not in full screen
BUG THE BIG BUG (I want to have my own Monster like my friend)	The game stopped running in the web player (mac/microsoft edge browser)
Calendar has stop working but I can still end day	no
Just about breeding my monster	I can't skip the 6th day.
Unsolvable problem (Random values didn't match the problems), can't move on to the next day (stuck at 07/01/2700), can't provide an answer (No ticking box)	Just misunderstanding the instruction but still working just fine.
	Monster didn't appear
	I don't know why I can't skip or end the day in calendar.

Figure G.41: Response to the satisfaction question 44

Do you have any suggestion for the game?
40 responses

-	I like the Hall of Intelligence in the puzzle part. It proves my understanding and is more fun than the question and answer.
ขอรวมเป็นเกมแนวไฟรวมรุ่นในงานเป็น ค่าความถูกไม่ถูกและไม่ถูกจันกินไป หากไม่คิดแล้วไม่รู้สึกว่าเป็น และไม่รู้สึกว่าเป็น	Can you make it to 3D game?
read above	ไม่มีอะไร(สนับสนุน)
ระบบจะดี	เมื่อเกมเข้ามานานมีมาก เพราะไม่เน้นเรื่องไข่ในการเล่นจะกระโดดไปเมืองfactory ที่แมกนัฟินทำให้เราเข้ามาบ่อ กิลล์factory อีก และเวลาให้เราออกตามท่าเดียวให้หัวหน้าบ่อคือหัวใจให้เราต้องเครียดตัวพื้นที่อยู่ข้างหลังบ่อ hall of intelligence มากไปกว่าที่จะรับมือไม่ไหวแล้วครับ
Create a game that name Capybara Island that allow me to catch Capybara Everyday UWU	
As fellow unity game dev, I suggest that you should move more of tweening (to make UI animation more fluid)	no
I don't know where to click and many instruction are hard to understand	Bug fix, please.
I want to date with Catbot. He's kinda cute. Please adding more story about Catbot.	No you good.
Need more stories, characters, CG, Capybaras, and date with CatBot. (Joking aside, fixing bug would be appreciated.)	Fun game But too bored for action lovers.
อยากให้ Hall of int. มาอยู่ชั้นนอกครับ อาจทำประมวลผลเป็นชุมชนคล้ายๆ community center แบบ stardew ที่มีอยู่ ครบ	game is ok but it's very tired to solve puzzle
	Fun game But too bored for action lovers.

Figure G.42: Response to the satisfaction question 45