Charles University in Prague Faculty of Mathematics and Physics

BACHELOR THESIS



Petr Mánek

Genetic programming in Swift for human-competitive evolution

Department of Software and Computer Science Education

Supervisor of the bachelor thesis: RNDr. František Mráz, CSc.

Study programme: Computer Science

Study branch: General Computer Science

| cited sources, literature and other pro I understand that my work relates to No. 121/2000 Sb., the Copyright Act, Charles University in Prague has the | for thesis independently, and only with the fessional sources. The rights and obligations under the Act as amended, in particular the fact that the right to conclude a license agreement on pursuant to Section 60 subsection 1 of the |
|---|--|
| In date | signature of the author |

Title: Genetic programming in Swift for human-competitive evolution

Author: Petr Mánek

Department: Department of Software and Computer Science Education

Supervisor: RNDr. František Mráz, CSc., Department of Software and Computer Science Education

Abstract: Imitating the process of natural selection, evolutionary algorithms have shown to be efficient search techniques for optimization and machine learning in poorly understood and irregular spaces. In this thesis, we implement a library containing essential implementation of such algorithms in recently unveiled programming language Swift. The result is a lightweight framework compatible with Linux-based computing clusters as well as mobile devices. Such wide range of supported platforms allows for successful application even in situations, where signals from various sensors have to be acquired and processed independently of other devices. In addition, thanks to Swift's minimalistic and functional syntax, the implementation of bundled algorithms and their sample usage clearly demonstrates fundamentals of genetic programming, making the work usable in teaching and quick prototyping of evolutionary algorithms.

Keywords: genetic programming artificial evolution

Dedication.

Contents

| 1.1 Entropy Generators 4 1.2 Data Structures 4 1.2.1 Randomizable Interface 4 1.2.2 Discrete Interface 4 1.3 Genetic Operators 4 1.3.1 Operator Life Cycle 5 1.3.2 Custom Interfaces 5 1.4 Selections 5 1.5 Algorithms 5 2 Library Implementation 6 2.1 Data Structures 6 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.2 Tree Structures 6 2.1.2 Tree Structures 6 2.1.2 Tree Structures 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 T | In | trod_{1} | uction | 3 | | |
|---|---------------|---------------------------|------------------------------|----|--|--|
| 1.1 Entropy Generators 4 1.2 Data Structures 4 1.2.1 Randomizable Interface 4 1.2.2 Discrete Interface 4 1.3 Genetic Operators 4 1.3.1 Operator Life Cycle 5 1.3.2 Custom Interfaces 5 1.4 Selections 5 1.5 Algorithms 5 2 Library Implementation 6 2.1 Data Structures 6 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.3 Program Interpretation 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.4 Algorithms 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 | 1 | Obi | ect-oriented Design | 4 | | |
| 1.2 Data Structures 4 1.2.1 Randomizable Interface 4 1.2.2 Discrete Interface 4 1.3 Genetic Operators 4 1.3.1 Operator Life Cycle 5 1.3.2 Custom Interfaces 5 1.4 Selections 5 1.5 Algorithms 5 2 Library Implementation 6 2.1 Data Structures 6 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.3 Program Interpretation 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of | | • | | 4 | | |
| 1.2.1 Randomizable Interface 4 1.2.2 Discrete Interface 4 1.3 Genetic Operators 4 1.3.1 Operator Life Cycle 5 1.3.2 Custom Interfaces 5 1.4 Selections 5 1.5 Algorithms 5 2 Library Implementation 6 2.1 Data Structures 6 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.3 Program Interpretation 6 2.2 Genetic Operators 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 | | 1.2 | 10 | 4 | | |
| 1.2.2 Discrete Interface 4 1.3 Genetic Operators 4 1.3.1 Operator Life Cycle 5 1.3.2 Custom Interfaces 5 1.4 Selections 5 1.5 Algorithms 5 2 Library Implementation 6 2.1 Data Structures 6 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.3 Program Interpretation 6 2.2.1 Reproduction 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 8 3.2 Self-driving Car Simulation 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 List of Tables 12 | | | | 4 | | |
| 1.3 Genetic Operators 4 1.3.1 Operator Life Cycle 5 1.3.2 Custom Interfaces 5 1.4 Selections 5 1.5 Algorithms 5 2 Library Implementation 6 2.1 Data Structures 6 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.3 Program Interpretation 6 2.2.1 Reproduction 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 List of Tables 12 | | | 1.2.2 Discrete Interface | 4 | | |
| 1.3.1 Operator Life Cycle 5 1.3.2 Custom Interfaces 5 1.4 Selections 5 1.5 Algorithms 5 1.5 Algorithms 5 2 Library Implementation 6 2.1 Data Structures 6 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.3 Program Interpretation 6 2.2 Genetic Operators 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 List of Tables | | 1.3 | | 4 | | |
| 1.3.2 Custom Interfaces 5 1.4 Selections 5 1.5 Algorithms 5 2 Library Implementation 6 2.1 Data Structures 6 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.3 Program Interpretation 6 2.2.1 Reproduction 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Tables 12 | | | - | 5 | | |
| 1.4 Selections 5 1.5 Algorithms 5 2 Library Implementation 6 2.1 Data Structures 6 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.3 Program Interpretation 6 2.2 Genetic Operators 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Tables 12 | | | - v | 5 | | |
| 2 Library Implementation 6 2.1 Data Structures 6 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.3 Program Interpretation 6 2.2 Genetic Operators 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 List of Figures 11 List of Tables 12 | | 1.4 | | 5 | | |
| 2.1 Data Structures 6 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.3 Program Interpretation 6 2.2 Genetic Operators 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Tables 12 | | 1.5 | Algorithms | 5 | | |
| 2.1 Data Structures 6 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.3 Program Interpretation 6 2.2 Genetic Operators 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Tables 12 | 2 | Libi | rary Implementation | 6 | | |
| 2.1.1 Common Types 6 2.1.2 Tree Structures 6 2.1.3 Program Interpretation 6 2.2 Genetic Operators 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Tables 12 | | | | 6 | | |
| 2.1.3 Program Interpretation 6 2.2 Genetic Operators 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | | | 6 | | |
| 2.2.2 Genetic Operators 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | | v - | 6 | | |
| 2.2 Genetic Operators 6 2.2.1 Reproduction 6 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | | 2.1.3 Program Interpretation | 6 | | |
| 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | 2.2 | | 6 | | |
| 2.2.2 Mutation 6 2.2.3 Crossover 6 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | | 2.2.1 Reproduction | 6 | | |
| 2.3 Selections 6 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 7 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | | - | 6 | | |
| 2.3.1 Roulette Selection 6 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | | 2.2.3 Crossover | 6 | | |
| 2.3.2 Rank Selection 6 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | 2.3 | Selections | 6 | | |
| 2.3.3 Tournament Selection 7 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | | 2.3.1 Roulette Selection | 6 | | |
| 2.3.4 Extensions 7 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | | 2.3.2 Rank Selection | 6 | | |
| 2.3.5 Optimizations 7 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | | 2.3.3 Tournament Selection | 7 | | |
| 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | | 2.3.4 Extensions | 7 | | |
| 2.4 Algorithms 7 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | | 2.3.5 Optimizations | 7 | | |
| 2.5 Event-driven Approach 7 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | 2.4 | | 7 | | |
| 2.6 Extensions 7 3 Usage Demonstration 8 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | 2.5 | | 7 | | |
| 3.1 Trivial Examples 8 3.2 Self-driving Car Simulation 8 3.3 QWOP Player 8 Conclusion 9 Bibliography 10 List of Figures 11 List of Tables 12 | | 2.6 | | 7 | | |
| 3.2 Self-driving Car Simulation 3.3 QWOP Player | 3 | Usage Demonstration | | | | |
| 3.3 QWOP Player | | 3.1 | Trivial Examples | 8 | | |
| 3.3 QWOP Player | | 3.2 | Self-driving Car Simulation | 8 | | |
| Bibliography 10 List of Figures 11 List of Tables 12 | | 3.3 | QWOP Player | 8 | | |
| List of Figures 11 List of Tables 12 | Co | onclu | asion | 9 | | |
| List of Tables 12 | Bi | bliog | graphy | 10 | | |
| List of Tables 12 | \mathbf{Li} | st of | Figures | 11 | | |
| | | | | | | |
| List of Abbreviations 13 | | List of Abbreviations | | | | |

Attachments 14

Introduction

Evolutionary Algorithms

TODO

Genetic Programming

TODO

The Swift Language

TODO

Practical Application

TODO

Structure of This Document

1. Object-oriented Design

In this chapter, the high-level design of individual components of the library is described.

Anděl [2007] **TODO**

1.1 Entropy Generators

Randomness plays a crucial role in evolutionary algorithms. Since the properties of pseudo-random generators impact the quality of produced solutions significantly, the library gives users full control over the algorithm, which is used to produce random sequences. In object design, this is achieved by simple abstraction.

The functionality of a random number generator is facilitated by an entropy generator object. In runtime, only a single instance of such object is created. This instance is then passed on to other components of the library, which require its capabilities, accessing it by reference. Users are responsible for instantiating this object, and can thus specify a seed for the generator or choose an algorithm particularly suitable for their application.

For the sake of minimality, entropy generators are only required to produce positive floating-point decimals from the [0; 1] interval. In spite of that, other components of the library can use them to generate random values of almost any type, provided that [0; 1] decimals can be mapped onto the type while achieving uniform distribution of generated values. This is further discussed in section 1.2.

1.2 Data Structures

Every individual in a generation is repesented by a separate instance of a class. The primary responsibility of such object is to store genetic information, which defines the individual. This information does not need to be held in a homogeneous data structure. In fact, it can be stored in any type suitable for the application. The only requirement on such type is that it can be generated randomly.

TODO

1.2.1 Randomizable Interface

TODO

1.2.2 Discrete Interface

TODO

1.3 Genetic Operators

1.3.1 Operator Life Cycle

TODO

1.3.2 Custom Interfaces

TODO

1.4 Selections

TODO

1.5 Algorithms

2. Library Implementation

TODO

2.1 Data Structures

TODO

2.1.1 Common Types

TODO

2.1.2 Tree Structures

TODO

2.1.3 Program Interpretation

TODO

2.2 Genetic Operators

TODO

2.2.1 Reproduction

TODO

2.2.2 Mutation

TODO

2.2.3 Crossover

TODO

2.3 Selections

TODO

2.3.1 Roulette Selection

TODO

2.3.2 Rank Selection

2.3.3 Tournament Selection

TODO

2.3.4 Extensions

TODO

2.3.5 Optimizations

TODO

2.4 Algorithms

TODO

2.5 Event-driven Approach

TODO

2.6 Extensions

3. Usage Demonstration

TODO

3.1 Trivial Examples

TODO

3.2 Self-driving Car Simulation

TODO

3.3 QWOP Player

Conclusion

Deployment

TODO

Teaching

TODO

Applications

Bibliography

J. Anděl. Základy matematické statistiky. Druhé opravené vydání. Matfyzpress, Praha, 2007. ISBN 80-7378-001-1.

List of Figures

List of Tables

List of Abbreviations

Attachments