**Module Interface Specification For The Input Module**

Brian Alexander

# Introduction

The Input module is responsible for opening sound files, interpreting their contents, and providing a stream of PCM data.

# Requirements

## Services Provided By this Module to Its Users

|  |  |  |
| --- | --- | --- |
| Service | Provided By | Tested By |
| 1. Open a sound file located at a given file path. | Open, Close |  |
| 1. Seek to a given location in a sound file as specified in seconds. | Seek |  |
| 1. Provide a stream of PCM data decoded from an audio file | GetFrames |  |

## Requirements for Services Needed By The Module

None

# Exceptions

|  |  |  |
| --- | --- | --- |
| Exception Name | Assumption | Tested By |
| NoFileOpen | GetFrames and Seek will not be called before a file has been opened |  |
| InvalidFilePath | The file path given to the Open method is not valid |  |

# Access Method Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Exceptions (for all access methods in this table): InvalidFilePath** | | | | | |
| **Access Methods** | **Parameter name** | **Parameter type** | **Parameter Info** | **Performance**  **Best - Worst** | **–Mapping to Services Provided** |
| Open | filepath | string | Relative or Absolute file path to sound file |  | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Exceptions (for all access methods in this table): NoFileOpen** | | | | | |
| **Access Methods** | **Parameter name** | **Parameter type** | **Parameter Info** | **Performance**  **Best - Worst** | **–Mapping to Services Provided** |
| Seek | Seconds | int | The location to seek to specified in number of seconds after the beginning |  | 2 |
| GetFrames | numFramesToRead | int | The number of frames to return in a stream |  | 3 |
| Close |  |  |  |  | 1 |

# Local Dictionary

## Parameter Terms

|  |  |
| --- | --- |
| **Parameter Term** | **Meaning** |
|  |  |
|  |  |

## Local Terms

|  |  |
| --- | --- |
| **Term** | **Meaning** |
|  |  |
|  |  |

## Local Conditions

|  |  |
| --- | --- |
| **Term** | **Meaning** |
|  |  |
|  |  |

## Local Data Types

|  |  |
| --- | --- |
| **Type** | **Value Space** |
|  |  |
|  |  |
|  |  |

# Access Methods Effects

List of Legal Traces: Open

Open.Seek

Open.Seek.GetFrames

Open.GetFrames

Open.GetFrames.Seek

|  |  |
| --- | --- |
| **Trace** | **Equivalence** |
| GetFrames.GetFrames | GetFrames |
| Seek.Seek | Seek |
| Seek.GetFrames.Seek | Seek |
| GetFrames.Seek.GetFrames | GetFrames |

# Events Signaled

None

# Externally Visible States of the Module

None

# Module Configuration Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter Name** | **Type** | **Default Value** | **Meaning** | **Parameter of Variation** | **Binding Time** |
| SupportedFileFormats | List<String> | {“.wav”,”.raw”} | What kinds of files can be played by the suite | P1 | Specification |

# Interface Design Issues

1, Should the file path be checked before it is passed to the Input module?

A1: Yes. The only parameters to the Open method will be valid file paths.

A2: No. The Input module will check the validity of the file path and throw an exception if it is invalid.

Resolution: Either method can be used. The Input can perform path checking, in which case the Open call should be in a “try” block. Or, the file path can be checked before the Open call is made.

# Implementation Notes.

# Efficiency Guide

# Test Cases

# Questions for Reviewers

Reviewers of this module interface specification should answer the following questions as they think about the specification.

## Requirements Validity

1. For each service provided by the module, is the service valid for all expected uses of this module? If not, give an example of a use where the service is not valid. (Systems Engineer)
2. For each service provided by the module, is the service valid for all expected configurations and versions of this module? If not, give an example of a needed configuration or version where the service is not valid. (Systems Engineer)
3. For each service needed described in this specification, is a module (or set of modules) identified that this module is allowed to use to satisfy the need? (Architect, Tester)
4. Are there cases where the interface specification could not be satisfied or was incomplete+? If so, how should it be changed? (Architect)

## Requirements Sufficiency

1. Does the set of services provided specify all of the services that will be needed by users of this module? Are there any services defined that are not identified in the requirements? (Systems Engineer)
2. Does the set of services needed specify all of the services that this module will need from other modules in order to operate correctly? What services are needed that are not identified in the requirements? (Architect)

## Consistency Between Services Provided and Access Programs

1. For each Services Provided described in this specification, which access program(s) can be used to satisfy the service? (Architect , Developer, Tester)
2. For each access program and signal specified in sections and which Service Provided is satisfied by the access programs? (Architect , Developer, Tester)

## Access Program Adequacy

1. Is the set of access programs and signals, including exceptions, sufficient to satisfy the uses needs of modules that are allowed to use this module? (Architect)
2. Are there access programs that should be combined into one access program? (Developer)
3. Are there single access programs that should be refactored into several different access programs? (Developer)
4. Are the performance requirements adequate for the uses that will be made of this module? (Tester)

## Implementation of Variability

1. Which variability (or variabilities) does this module implement? (Systems Engineer)
2. Are all values in the range defined in the parameters of variation accounted for in implementing each variability? Which values are not? (Developer)
3. Can the variabilities be bound at the time specified in the commonality analysis for the variabilities? (Architect)