## System Operation Contracts

|  |  |
| --- | --- |
| Name | **GetFunctions(CodeGenerationSettings)**  **GetProblematicFunctions(CodeGenerationSettings)** |
| Task | Gets the functions or problematic functions. A function is problematic if one of the parameters or the return type is marked as problematic or if an old function is attached. |
| Preconditions | - |
| Postconditions | - |
| Exceptions | - |
| Type | System |
| Links | UC “Change Code Generation Settings”, SSD “Change Code Generation Settings“ |
| Result | Function[\*] |
| Notes | - |

|  |  |
| --- | --- |
| Name | **SetTypeConversion(Type, TypeConversion)** |
| Task | Replaces the given Type’s current TypeConversion with the given one. |
| Preconditions | - |
| Postconditions | * The Type’s association to the original TypeConversion was deleted. * The original TypeConversion was deleted. * The Type was associated to the given TypeConversion. * The Type’s UserModified attribute was set to true. |
| Exceptions | No Type or TypeConversion given. |
| Type | System |
| Links | UC “Change Code Generation Settings”, SSD “Change Code Generation Settings“ |
| Result | void |
| Notes | Since the implementation is trivial, there is no sequence diagram in the design model. |

|  |  |
| --- | --- |
| Name | **SetProblemIsResolved(Type)** |
| Task | Sets the given Type’s ProblemIsResolved attribute to true. |
| Preconditions | - |
| Postconditions | The given Type’s ProblemIsResolved attribute was set to true. |
| Exceptions | - |
| Type | System |
| Links | UC “Change Code Generation Settings”, SSD “Change Code Generation Settings“ |
| Result | void |
| Notes | Since the implementation is trivial, there is no sequence diagram in the design model. |

|  |  |
| --- | --- |
| Name | **SetUpdateProblemIsResolved(Function)** |
| Task | Sets the given Function’s update problem as resolved. |
| Preconditions | - |
| Postconditions | The given Function’s attachedTo association to another Function was removed. |
| Exceptions | - |
| Type | System |
| Links | UC “Change Code Generation Settings”, SSD “Change Code Generation Settings“ |
| Result | void |
| Notes | Since the implementation is trivial, there is no sequence diagram in the design model. |

|  |  |
| --- | --- |
| Name | **GenerateCode(CodeGenerationSettings)** |
| Task | Uses the current CodeGenerationSettings to generate the code. |
| Preconditions | - |
| Postconditions | The code was generated and written to the C++ and C# code files. |
| Exceptions | - |
| Type | System |
| Links | UC “Generate Code”, SSD “Generate Code“ |
| Result | void |
| Notes | The code that needs to be generated is:   * C++: The Horde3DNoProfilingProxy and Horde3DProfilingProxy classes. * C#: The Horde3DCall class. |

|  |  |
| --- | --- |
| Name | **LoadCodeGenerationSettings(filePath : string)** |
| Task | Loads a CodeGenerationSettings object and all associated functions (and their associated TypeConversion) from the given file. |
| Preconditions | - |
| Postconditions | A CodeGenerationSetttings object was deserialized from the given file and returned. |
| Exceptions | The specified file could not be read or does not contain a serialized CodeGenerationSettings object. |
| Type | System |
| Links | UC “Load Code Generation Settings”, SSD “Load Code Generation Settings“ |
| Result | CodeGenerationSettings |
| Notes | The CodeGenerationSettings objects are deserialized using the standard .NET xml to objects deserialization. Since the implementation is trivial, there is no sequence diagram in the design model. |

|  |  |
| --- | --- |
| Name | **SaveCodeGenerationSettings(CodeGenerationSettings,** **filePath : string)** |
| Task | Stores the given CodeGenerationSettings object and all associated functions (and their associated TypeConversion) in the given file. If no filePath was given, the previously used filePath is used again. |
| Preconditions | There is a current CodeGenerationSettings object. |
| Postconditions | The CodeGenerationSettings object was stored in the specified file and can later be loaded again using LoadCodeGenerationSettings(filePath). |
| Exceptions | The specified file could not be created or overwritten. No file was specified and the CodeGenerationSettings have never been saved before. |
| Type | System |
| Links | UC “Save Code Generation Settings”, SSD “Save Code Generation Settings“ |
| Result | void |
| Notes | The CodeGenerationSettings objects are serialized using the standard .NET objects to xml serialization. Since the implementation is trivial, there is no sequence diagram in the design model. |

|  |  |
| --- | --- |
| Name | **ParseHorde3DHeaderFile(CodeGenerationSettings, filePath : string)** |
| Task | The system parses the file and creates default code generation settings for each Horde3D function. Type conversions of the functions' return types and parameter types are automatically attempted. If such an attempt seems problematic, the parameter or return type is marked as problematic.  If there are already any functions associated to the current CodeGenerationSettings, the old functions are compared to the new ones. For each function in the old function list that was modified by the user the new function settings are looked up. The changes made by the user are applied to the new function object, if possible. If the new function's parameters and return type were problematic but are now no longer problematic, their code generation problems are marked as resolved. If the changes made by the user could not be applied, the old function is attached to the function and its code generation problems are marked as unresolved.  All functions associated to the current CodeGenerationSettings are returned. |
| Preconditions | - |
| Postconditions | * The file was parsed and for each Horde3D function found in the file a Function object was created. * For all parameters and the return type a Type object was created. * For all created Type objects an automatic type conversion was attempted: A TypeConversion object associated to the Type. * All created Function objects were associated to the current CodeGenerationSettings. |
| Exceptions | No file given or the given file is not a Horde3D header file. |
| Type | System |
| Links | UC “Parse Horde3D Header File”, SSD “Parse Horde3D Header File“ |
| Result | void |
| Notes | - |