

AdventureLearn

Generated by Doxygen 1.8.17

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 Class Documentation	7
3.1 Assignment Class Reference	7
3.1.1 Detailed Description	7
3.2 AssignmentBL Class Reference	7
3.2.1 Detailed Description	8
3.2.2 Member Function Documentation	8
3.2.2.1 GetAssignment()	8
3.2.2.2 GetAssignmentMonster()	8
3.2.2.3 GetStudentAssignment()	8
3.3 AssignmentDaolmpl Class Reference	10
3.3.1 Detailed Description	10
3.3.2 Member Function Documentation	10
3.3.2.1 GetAssignment()	10
3.3.2.2 GetAssignmentMonster()	11
3.3.2.3 GetStudentAssignment()	11
3.4 AssignmentScore Class Reference	11
3.4.1 Detailed Description	12
3.5 AssignmentScoreBL Class Reference	12
3.5.1 Detailed Description	12
3.5.2 Member Function Documentation	12
3.5.2.1 GetStudentCompletedAssignment()	12
3.5.2.2 InsertAssignmentScore()	13
3.6 AssignmentScoreDaolmpl Class Reference	13
3.6.1 Detailed Description	13
3.6.2 Member Function Documentation	13
3.6.2.1 GetStudentCompletedAssignment()	13
3.6.2.2 InsertAssignmentScore()	14
3.7 AssignmentScreen Class Reference	14
3.7.1 Detailed Description	14
3.7.2 Member Function Documentation	15
3.7.2.1 _Ready()	15
3.8 BackButton Class Reference	15
3.8.1 Detailed Description	15
3.9 BaseDaolmpl< T > Class Template Reference	15
3.9.1 Detailed Description	15
3.9.2 Member Function Documentation	16
3.9.2.1 ExecuteQuery() [1/2]	16

3.9.2.2 ExecuteQuery() [2/2]	16
3.9.2.3 ExecuteScalar() [1/2]	17
3.9.2.4 ExecuteScalar() [2/2]	17
3.9.2.5 RetrieveQuery()	17
3.10 BelongClass Class Reference	18
3.10.1 Detailed Description	18
3.11 Campaign Class Reference	18
3.11.1 Detailed Description	18
3.11.2 Member Function Documentation	18
3.11.2.1 _Ready()	18
3.12 CampaignBL Class Reference	19
3.12.1 Detailed Description	19
3.12.2 Member Function Documentation	19
3.12.2.1 GetLevel()	19
3.13 CampaignDaoImpl Class Reference	19
3.13.1 Detailed Description	20
3.13.2 Member Function Documentation	20
3.13.2.1 GetLevel()	20
3.14 CampaignQuestion Class Reference	20
3.14.1 Detailed Description	20
3.15 Character Class Reference	20
3.15.1 Detailed Description	21
3.16 CharacterBL Class Reference	21
3.16.1 Detailed Description	21
3.16.2 Member Function Documentation	21
3.16.2.1 GetAllCharacters()	21
3.16.2.2 GetAllMonsters()	22
3.16.2.3 GetCharacter()	22
3.17 CharacterDaoImpl Class Reference	22
3.17.1 Detailed Description	22
3.17.2 Member Function Documentation	23
3.17.2.1 GetAllCharacters()	23
3.17.2.2 GetAllMonsters()	23
3.17.2.3 GetCharacter()	23
3.18 CharSelect Class Reference	23
3.18.1 Detailed Description	24
3.18.2 Member Function Documentation	24
3.18.2.1 _Ready()	24
3.19 Chart Class Reference	24
3.19.1 Detailed Description	24
3.19.2 Member Function Documentation	24
3.19.2.1 _Ready()	24

3.20 ClassGroup Class Reference	25
3.20.1 Detailed Description	25
3.21 ClassGroupDaoImpl Class Reference	25
3.21.1 Detailed Description	25
3.21.2 Member Function Documentation	25
3.21.2.1 GetClassGroup()	25
3.22 CreateLevel Class Reference	26
3.22.1 Detailed Description	26
3.22.2 Member Function Documentation	26
3.22.2.1 _Ready()	26
3.22.2.2 SetLevelInitInfo()	26
3.23 CreateLevelBL Class Reference	27
3.23.1 Detailed Description	27
3.23.2 Member Function Documentation	28
3.23.2.1 CheckDuplicationOptions()	28
3.23.2.2 CheckEmptyFieldsExist()	28
3.23.2.3 CheckValidLevelName()	28
3.23.2.4 CreateLevel()	29
3.23.2.5 GetQuestion()	29
3.23.2.6 getTempQuestionList()	29
3.23.2.7 InitializeQuestions()	30
3.23.2.8 ListQuestions()	30
3.23.2.9 ReloadTempQuestionList()	30
3.23.2.10 SaveQuestion()	30
3.24 CreateLevelDAOImpl Class Reference	31
3.24.1 Detailed Description	31
3.24.2 Member Function Documentation	31
3.24.2.1 CheckValidLevelName()	31
3.24.2.2 InsertCustomLevel()	31
3.24.2.3 InsertQuestion()	32
3.25 CreateLevelInit Class Reference	32
3.25.1 Detailed Description	33
3.25.2 Member Function Documentation	33
3.25.2.1 _Ready()	33
3.26 Credits Class Reference	33
3.26.1 Detailed Description	33
3.26.2 Member Function Documentation	33
3.26.2.1 _Ready()	33
3.27 CustomLevel Class Reference	33
3.27.1 Detailed Description	34
3.28 CustomLevelBL Class Reference	34
3.28.1 Detailed Description	34

3.28.2 Member Function Documentation	34
3.28.2.1 DeleteCustomLevel()	34
3.28.2.2 GetClearedCustomLevels()	35
3.28.2.3 GetCustomLevel()	35
3.28.2.4 GetCustomLevelMonster()	36
3.28.2.5 GetCustomLevels()	36
3.28.2.6 GetStudentCustomLevel()	36
3.29 CustomLevelDaoImpl Class Reference	36
3.29.1 Detailed Description	37
3.29.2 Member Function Documentation	37
3.29.2.1 DeleteCustomLevel()	37
3.29.2.2 GetClearedCustomLevels()	37
3.29.2.3 GetCustomLevel()	38
3.29.2.4 GetCustomLevelMonster()	38
3.29.2.5 GetCustomLevels()	39
3.29.2.6 GetStudentCustomLevel()	39
3.30 CustomLevelScore Class Reference	39
3.30.1 Detailed Description	39
3.31 CustomLevelScoreBL Class Reference	40
3.31.1 Detailed Description	40
3.31.2 Member Function Documentation	40
3.31.2.1 InsertCustomLevelScore()	40
3.32 CustomLevelScoreDaoImpl Class Reference	40
3.32.1 Detailed Description	41
3.32.2 Member Function Documentation	41
3.32.2.1 InsertCustomLevelScore()	41
3.33 CustomLevelScreen Class Reference	41
3.33.1 Detailed Description	41
3.33.2 Member Function Documentation	42
3.33.2.1 _Ready()	42
3.34 DefaultSound Class Reference	42
3.34.1 Detailed Description	42
3.34.2 Member Function Documentation	42
3.34.2.1 _Ready()	43
3.34.2.2 disableSound()	43
3.34.2.3 enableSound()	43
3.34.2.4 playSound()	43
3.35 EditLevel Class Reference	43
3.35.1 Detailed Description	44
3.35.2 Member Function Documentation	44
3.35.2.1 _Ready()	44
3.35.2.2 SetLevelInitInfo()	44

3.36 EditLevelBL Class Reference	44
3.36.1 Detailed Description	45
3.36.2 Member Function Documentation	45
3.36.2.1 CheckDuplicationOptions()	45
3.36.2.2 CheckEmptyFieldsExist()	46
3.36.2.3 CheckValidLevelName()	46
3.36.2.4 GetOriginalQuestionList()	46
3.36.2.5 GetQuestion()	46
3.36.2.6 GetTempQuestionList()	47
3.36.2.7 LoadCustomLevelInfo()	47
3.36.2.8 ReloadTempQuestionList()	47
3.36.2.9 SaveQuestion()	47
3.36.2.10 UpdateLevel()	48
3.36.2.11 UpdateLevelInitInfo()	48
3.37 EditLevelDaoImpl Class Reference	48
3.37.1 Detailed Description	49
3.37.2 Member Function Documentation	49
3.37.2.1 CheckValidLevelName()	49
3.37.2.2 GetLevelInfo()	49
3.37.2.3 UpdateLevelInitInfo()	49
3.37.2.4 UpdateQuestion()	50
3.38 EditLevelInit Class Reference	50
3.38.1 Detailed Description	51
3.38.2 Member Function Documentation	51
3.38.2.1 _Ready()	51
3.39 GamePlay Class Reference	51
3.39.1 Detailed Description	52
3.39.2 Member Function Documentation	52
3.39.2.1 _Process()	52
3.39.2.2 _Ready()	52
3.39.2.3 CheckCorrectAnswer()	52
3.39.2.4 DisplayCharSprite()	53
3.39.2.5 DisplayMonsterSprite()	53
3.39.2.6 DisplayNextQuestion()	53
3.39.2.7 DisplayQuestion()	53
3.39.2.8 DisplayStars()	54
3.39.2.9 GetTimeLeft()	54
3.39.2.10 LoadStart()	54
3.39.2.11 SetBg()	54
3.39.2.12 SetCharacter()	55
3.39.2.13 SetGameType()	55
3.39.2.14 SetLevelTitle()	55

3.39.2.15 SetQuestionList()	55
3.39.2.16 SetQuestionNum()	56
3.39.2.17 SetTimeLimit()	56
3.40 Global Class Reference	56
3.40.1 Detailed Description	58
3.40.2 Member Function Documentation	58
3.40.2.1 CalculateScore()	58
3.40.2.2 GetFirstLoggedIn()	58
3.40.2.3 GetStudentName()	58
3.40.2.4 LoadSprite()	59
3.40.2.5 SetFbLoggedIn()	59
3.40.2.6 SetFirstLoggedIn()	59
3.40.2.7 SetGoogleLoggedIn()	59
3.40.2.8 SetStudentId()	60
3.40.2.9 SetStudentName()	61
3.40.3 Member Data Documentation	61
3.40.3.1 csb	61
3.40.4 Property Documentation	61
3.40.4.1 StudentName	61
3.41 Leaderboard Class Reference	62
3.41.1 Detailed Description	62
3.42 LeaderboardBL Class Reference	62
3.42.1 Detailed Description	62
3.42.2 Member Function Documentation	62
3.42.2.1 GetLeaderboards()	62
3.42.2.2 GetWorldLeaderboard()	62
3.42.2.3 GetWorlds()	63
3.43 LeaderboardDaoImpl Class Reference	63
3.43.1 Detailed Description	63
3.43.2 Member Function Documentation	63
3.43.2.1 GetLeaderboardScore() [1/2]	63
3.43.2.2 GetLeaderboardScore() [2/2]	63
3.44 LeaderboardScreen Class Reference	64
3.44.1 Detailed Description	64
3.44.2 Member Function Documentation	64
3.44.2.1 _Ready()	64
3.45 Level Class Reference	64
3.45.1 Detailed Description	65
3.46 LevelSelection Class Reference	65
3.46.1 Detailed Description	65
3.46.2 Member Function Documentation	65
3.46.2.1 _Ready()	65

3.47 Login Class Reference	65
3.47.1 Detailed Description	66
3.47.2 Member Function Documentation	66
3.47.2.1 _Ready()	66
3.48 MainMenu Class Reference	66
3.48.1 Detailed Description	66
3.48.2 Member Function Documentation	66
3.48.2.1 _Ready()	66
3.49 Monster Class Reference	67
3.49.1 Detailed Description	67
3.50 NotificationPopup Class Reference	67
3.50.1 Detailed Description	67
3.50.2 Member Function Documentation	67
3.50.2.1 _Ready()	68
3.50.2.2 DisplayPopup()	68
3.51 PublishAssignmentScreen Class Reference	68
3.51.1 Detailed Description	68
3.51.2 Member Function Documentation	68
3.51.2.1 _Ready()	68
3.52 PublishedAssignment Class Reference	69
3.52.1 Detailed Description	69
3.53 PublishedAssignmentBL Class Reference	69
3.53.1 Detailed Description	69
3.53.2 Member Function Documentation	69
3.53.2.1 GetClassNotAssignment()	69
3.53.2.2 GetTeacherPublished()	70
3.53.2.3 InsertPublishedAssignment()	70
3.54 PublishedAssignmentDaoImpl Class Reference	71
3.54.1 Detailed Description	71
3.54.2 Member Function Documentation	71
3.54.2.1 GetClassNotAssignment()	71
3.54.2.2 GetTeacherPublished()	72
3.54.2.3 InsertPublishedAssignment()	72
3.55 Question Class Reference	72
3.55.1 Detailed Description	73
3.56 Section Class Reference	73
3.56.1 Detailed Description	73
3.57 SectionBL Class Reference	73
3.57.1 Detailed Description	74
3.57.2 Member Function Documentation	74
3.57.2.1 CheckSectionCleared()	74
3.57.2.2 GetSectionLevels()	74

3.57.2.3 GetWorldSections()	75
3.58 SectionDaoImpl Class Reference	75
3.58.1 Detailed Description	75
3.58.2 Member Function Documentation	75
3.58.2.1 CheckSectionCleared()	75
3.58.2.2 GetSectionLevels()	76
3.58.2.3 GetWorldSections()	76
3.59 Settings Class Reference	77
3.59.1 Detailed Description	77
3.59.2 Member Function Documentation	77
3.59.2.1 _Ready()	77
3.60 StatisticsBL Class Reference	77
3.60.1 Detailed Description	79
3.60.2 Member Function Documentation	79
3.60.2.1 GetAssignmentRanking()	79
3.60.2.2 GetAssignments()	79
3.60.2.3 GetAvgAssignmentScore() [1/2]	80
3.60.2.4 GetAvgAssignmentScore() [2/2]	80
3.60.2.5 GetAvgSectionScores()	80
3.60.2.6 GetAvgSectionScoresAvg()	81
3.60.2.7 GetAvgSectionScoresMax()	81
3.60.2.8 GetAvgSectionScoresMin()	81
3.60.2.9 GetAvgStudentAssignmentScores()	82
3.60.2.10 GetAvgStudentScore()	82
3.60.2.11 GetAvgWorldScores()	82
3.60.2.12 GetAvgWorldScoresAvg()	83
3.60.2.13 GetAvgWorldScoresMax()	83
3.60.2.14 GetAvgWorldScoresMin()	83
3.60.2.15 GetCampaignRanking()	83
3.60.2.16 GetMaxAssignmentScore()	84
3.60.2.17 GetMaxStudentAssignmentScores()	84
3.60.2.18 GetMaxStudentScore()	85
3.60.2.19 GetMinAssignmentScore()	85
3.60.2.20 GetMinStudentAssignmentScores()	85
3.60.2.21 GetMinStudentScore()	86
3.60.2.22 GetSections()	86
3.60.2.23 GetStudentAssignmentScores()	87
3.60.2.24 GetStudentClass()	87
3.60.2.25 GetStudents()	87
3.60.2.26 GetStudentScores() [1/2]	88
3.60.2.27 GetStudentScores() [2/2]	88
3.60.2.28 GetWorlds()	88

3.60.2.29 GetWorldSections()	89
3.61 Student Class Reference	89
3.61.1 Detailed Description	89
3.62 StudentBL Class Reference	89
3.62.1 Detailed Description	90
3.62.2 Member Function Documentation	90
3.62.2.1 CheckFacebookCharExist()	90
3.62.2.2 CheckFacebookExist()	91
3.62.2.3 CheckGoogleCharExist()	91
3.62.2.4 CheckGoogleExist()	91
3.62.2.5 CheckStudentCharExist()	92
3.62.2.6 CheckStudentExist()	92
3.62.2.7 GetFacebookStudentId()	93
3.62.2.8 GetGoogleStudentId()	93
3.62.2.9 GetStudentCharacter()	93
3.62.2.10 InsertFacebookStudent()	94
3.62.2.11 InsertGoogleStudent()	94
3.62.2.12 UpdateStudentCharacter()	95
3.63 StudentCustomQuestion Class Reference	95
3.63.1 Detailed Description	95
3.64 StudentDaoImpl Class Reference	95
3.64.1 Detailed Description	96
3.64.2 Member Function Documentation	96
3.64.2.1 CheckFacebookCharExist()	96
3.64.2.2 CheckFacebookExist()	97
3.64.2.3 CheckGoogleCharExist()	97
3.64.2.4 CheckGoogleExist()	97
3.64.2.5 CheckStudentExist()	98
3.64.2.6 GetFacebookStudent()	98
3.64.2.7 GetGoogleStudent()	99
3.64.2.8 GetStudentCharacter()	99
3.64.2.9 InsertFacebookStudent()	99
3.64.2.10 InsertGoogleStudent()	100
3.64.2.11 UpdateStudentCharacter()	100
3.65 StudentScore Class Reference	100
3.65.1 Detailed Description	101
3.66 StudentScoreBL Class Reference	101
3.66.1 Detailed Description	101
3.66.2 Member Function Documentation	101
3.66.2.1 GetAvgWorldScores()	101
3.66.2.2 GetCampaignRanking()	102
3.66.2.3 GetStudentScores()	102

3.66.2.4 InsertStudentScore()	103
3.67 StudentScoreDaoImpl Class Reference	103
3.67.1 Detailed Description	103
3.67.2 Member Function Documentation	103
3.67.2.1 GetAvgWorldScores()	104
3.67.2.2 GetCampaignRanking()	105
3.67.2.3 GetStudentScores()	105
3.67.2.4 InsertStudentScore()	106
3.68 Teacher Class Reference	106
3.68.1 Detailed Description	106
3.69 TeacherAssignment Class Reference	106
3.69.1 Detailed Description	107
3.69.2 Member Function Documentation	107
3.69.2.1 _Ready()	107
3.70 TeacherBL Class Reference	107
3.70.1 Detailed Description	107
3.70.2 Member Function Documentation	108
3.70.2.1 CheckEmail()	108
3.70.2.2 CheckFacebookLogin()	108
3.70.2.3 CheckGoogleLogin()	108
3.70.2.4 GetFacebookTeacherId()	109
3.70.2.5 GetGoogleTeacherId()	109
3.70.2.6 UpdateFacebookId()	109
3.70.2.7 UpdateGoogleId()	110
3.71 TeacherCustomQuestion Class Reference	110
3.71.1 Detailed Description	110
3.72 TeacherDaoImpl Class Reference	111
3.72.1 Detailed Description	111
3.72.2 Member Function Documentation	111
3.72.2.1 CheckEmail()	111
3.72.2.2 CheckFacebookLogin()	112
3.72.2.3 CheckGoogleLogin()	112
3.72.2.4 GetFacebookTeacher()	112
3.72.2.5 GetGoogleTeacher()	113
3.72.2.6 UpdateFacebookId()	113
3.72.2.7 UpdateGoogleId()	113
3.73 TeacherLogin Class Reference	114
3.73.1 Detailed Description	114
3.73.2 Member Function Documentation	114
3.73.2.1 _Ready()	114
3.74 TeacherMainMenu Class Reference	114
3.74.1 Detailed Description	115

3.74.2 Member Function Documentation	115
3.74.2.1 _Ready()	115
3.75 UserCreatedQuestion Class Reference	115
3.75.1 Detailed Description	115
3.76 UserProfile Class Reference	115
3.76.1 Detailed Description	116
3.76.2 Member Function Documentation	116
3.76.2.1 _Ready()	116
3.77 ViewAssignment Class Reference	116
3.77.1 Detailed Description	116
3.77.2 Member Function Documentation	116
3.77.2.1 _Ready()	116
3.78 ViewCreatedLevels Class Reference	117
3.78.1 Detailed Description	117
3.78.2 Member Function Documentation	117
3.78.2.1 _Ready()	117
3.79 ViewCustomLevel Class Reference	117
3.79.1 Detailed Description	117
3.79.2 Member Function Documentation	117
3.79.2.1 _Ready()	118
3.80 World Class Reference	118
3.80.1 Detailed Description	118
3.81 WorldBL Class Reference	118
3.81.1 Detailed Description	118
3.81.2 Member Function Documentation	118
3.81.2.1 GetCompletedWorldCount()	119
3.81.2.2 GetTotalWorldCount()	119
3.82 WorldDaoImpl Class Reference	119
3.82.1 Detailed Description	119
3.82.2 Member Function Documentation	119
3.82.2.1 GetCompletedWorldCount()	120
3.82.2.2 GetWorlds()	120
3.83 WorldScreen Class Reference	120
3.83.1 Detailed Description	120
3.83.2 Member Function Documentation	120
3.83.2.1 _Ready()	120

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

AssignmentBL	7
AssignmentDaoImpl	10
AssignmentScoreDaoImpl	13
BaseDaoImpl< T >	15
CampaignBL	19
CampaignDaoImpl	19
CharacterBL	21
CharacterDaoImpl	22
ClassGroupDaoImpl	25
Control	
NotificationPopup	67
CreateLevelDAOImpl	31
CustomLevelBL	34
CustomLevelDaoImpl	36
CustomLevelScoreDaoImpl	40
EditLevelDaoImpl	48
LeaderboardBL	62
LeaderboardDaoImpl	63
Node	
Assignment	7
AssignmentScore	11
AssignmentScoreBL	12
BelongClass	18
CampaignQuestion	20
Character	20
ClassGroup	25
CreateLevelBL	27
CustomLevel	33
CustomLevelScore	39
CustomLevelScoreBL	40
EditLevelBL	44
Global	56
Leaderboard	62
LeaderboardScreen	64
Level	64

Monster	67
PublishedAssignment	69
Question	72
Section	73
StatisticsBL	77
Student	89
StudentBL	89
StudentCustomQuestion	95
StudentScore	100
Teacher	106
TeacherBL	107
TeacherCustomQuestion	110
UserCreatedQuestion	115
World	118
WorldBL	118
WorldScreen	120
Node2D	
AssignmentScreen	14
BackButton	15
Campaign	18
CharSelect	23
Chart	24
CreateLevel	26
CreateLevelInit	32
Credits	33
CustomLevelScreen	41
DefaultSound	42
EditLevel	43
EditLevelInit	50
GamePlay	51
LevelSelection	65
Login	65
MainMenu	66
PublishAssignmentScreen	68
Settings	77
TeacherAssignment	106
TeacherLogin	114
TeacherMainMenu	114
UserProfile	115
ViewAssignment	116
ViewCreatedLevels	117
ViewCustomLevel	117
PublishedAssignmentBL	69
PublishedAssignmentDaoImpl	71
SectionBL	73
SectionDaoImpl	75
StudentDaoImpl	95
StudentScoreBL	101
StudentScoreDaoImpl	103
TeacherDaoImpl	111
WorldDaoImpl	119

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Assignment	
DAO Object for Assignment	7
AssignmentBL	
Class to handle Business Logic for Assignment	7
AssignmentDaoImpl	
Class to handle DAO operations for Assignment	10
AssignmentScore	
DAO Object for AssignmentScore	11
AssignmentScoreBL	
Class to handle Business Logic for AssignmentScore	12
AssignmentScoreDaoImpl	
Class to handle DAO operations for AssignmentScore	13
AssignmentScreen	
Class to handle Presentation for AssignmentScreen (gameplay)	14
BackButton	
Class to handle Presentation for BackButton	15
BaseDaoImpl< T >	
Class to handle DAO operations for Generic methods	15
BelongClass	
DAO Object for BelongClass	18
Campaign	
Class to handle Presentation for Campaign	18
CampaignBL	
Class to handle Business Logic for Campaign	19
CampaignDaoImpl	
Class to handle DAO operations for Campaign	19
CampaignQuestion	
DAO Object for CampaignQuestion	20
Character	
DAO Object for Character	20
CharacterBL	
Class to handle Business Logic for Character	21
CharacterDaoImpl	
Class to handle DAO operations for Character	22
CharSelect	
Class to handle Presentation for Character Select	23

Chart	
Class to handle Presentation for Section	24
ClassGroup	
DAO Object for ClassGroup	25
ClassGroupDaoImpl	
Class to handle DAO operations for ClassGroup	25
CreateLevel	
Class to handle Presentation for CreateLevel	26
CreateLevelBL	
Class to handle Business Logic for CreateLevel	27
CreateLevelDAOImpl	
Class to handle DAO operations for CreateLevel	31
CreateLevelInit	
Class to handle Presentation for CreateLevelInit	32
Credits	
Class to handle Presentation for Credits	33
CustomLevel	
DAO Object for CustomLevel	33
CustomLevelBL	
Class to handle Business Logic for CustomLevel	34
CustomLevelDaoImpl	
Class to handle DAO operations for CustomLevel	36
CustomLevelScore	
DAO Object for CustomLevelScore	39
CustomLevelScoreBL	
Class to handle Business Logic for CustomLevelScore	40
CustomLevelScoreDaoImpl	
Class to handle DAO operations for CustomLevelScore	40
CustomLevelScreen	
Class to handle Presentation for CustomLevelScreen (gameplay)	41
DefaultSound	
Class to handle Presentation for DefaultSound	42
EditLevel	
Class to handle Presentation for EditLevel	43
EditLevelBL	
Class to handle Business Logic for EditLevel	44
EditLevelDaoImpl	
Class to handle DAO operations for EditLevel	48
EditLevelInit	
Class to handle Presentation for EditLevelInit	50
GamePlay	
Class to handle Presentation for GamePlay	51
Global	
Global class to store Global variables	56
Leaderboard	
DAO Object for Leaderboard	62
LeaderboardBL	
Class to handle Business Logic for Leaderboard	62
LeaderboardDaoImpl	
Class to handle DAO operations for Leaderboard	63
LeaderboardScreen	
Class to handle Presentation for LeaderboardScreen	64
Level	
DAO Object for Level	64
LevelSelection	
Class to handle Presentation for LevelSelection	65
Login	
Class to handle Presentation for Login	65

MainMenu		
Class to handle Presentation for Section	66	
Monster		
DAO Object for Monster	67	
NotificationPopup		
Class to handle Presentation for NotificationPopup	67	
PublishAssignmentScreen		
Class to handle Presentation for PublishAssignment	68	
PublishedAssignment		
DAO Object for PublishedAssignment	69	
PublishedAssignmentBL		
Class to handle Business Logic for PublishedAssignment	69	
PublishedAssignmentDaoImpl		
Class to handle DAO operations for PublishedAssignment	71	
Question		
DAO Object for Question	72	
Section		
DAO Object for Section	73	
SectionBL		
Class to handle Business Logic for Section	73	
SectionDaoImpl		
Class to handle DAO operations for SectionDao	75	
Settings		
Class to handle Presentation for Settings	77	
StatisticsBL		
Class to handle Business Logic for Statistics	77	
Student		
DAO Object for Student	89	
StudentBL		
Class to handle Business Logic for Student	89	
StudentCustomQuestion		
DAO Object for StudentCustomQuestion	95	
StudentDaoImpl		
Class to handle DAO operations for Student	95	
StudentScore		
DAO Object for StudentScore	100	
StudentScoreBL		
Class to handle Business Logic for StudentScore	101	
StudentScoreDaoImpl		
Class to handle DAO operations for StudentScore	103	
Teacher		
DAO Object for Teacher	106	
TeacherAssignment		
Class to handle Presentation for AssignmentScreen (gameplay)	106	
TeacherBL		
Class to handle Business Logic for Teacher	107	
TeacherCustomQuestion		
DAO Object for TeacherCustomQuestion	110	
TeacherDaoImpl		
Class to handle DAO operations for Teacher	111	
TeacherLogin		
Class to handle Presentation for Login	114	
TeacherMainMenu		
Class to handle Presentation for Section	114	
UserCreatedQuestion		
DAO Object for UserCreatedQuestion	115	
UserProfile		
Class to handle Presentation for UserProfile	115	

ViewAssignment	
Class to handle Presentation for ViewAssignment	116
ViewCreatedLevels	
Class to handle Presentation for ViewCreatedLevels	117
ViewCustomLevel	
Class to handle Presentation for ViewCustomLevel	117
World	
DAO Object for World	118
WorldBL	
Class to handle Business Logic for World	118
WorldDaoImpl	
Class to handle DAO operations for World	119
WorldScreen	
Class to handle Presentation for WorldScreen	120

Chapter 3

Class Documentation

3.1 Assignment Class Reference

DAO Object for [Assignment](#)

Inherits Node.

Properties

- int **AssignmentId** [get, set]
- [Teacher](#) **Teacher** [get, set]
- string **AssignmentName** [get, set]
- [Monster](#) **Monster** [get, set]
- int **TimeLimit** [get, set]
- List< [Question](#) > **Question** [get, set]

3.1.1 Detailed Description

DAO Object for [Assignment](#)

3.2 AssignmentBL Class Reference

Class to handle Business Logic for [Assignment](#)

Public Member Functions

- [Assignment](#) **GetAssignment** (int assignmentId)
Get selected [Assignment](#)
- List< [PublishedAssignment](#) > **GetStudentAssignment** (int studentId)
Get all [Student](#)'s published [Assignments](#)
- [Monster](#) **GetAssignmentMonster** (int assignmentId)
Get [Monster](#) that belongs to selected [Assignment](#)

3.2.1 Detailed Description

Class to handle Business Logic for [Assignment](#)

3.2.2 Member Function Documentation

3.2.2.1 GetAssignment()

```
Assignment AssignmentBL.GetAssignment (
    int assignmentId )
```

Get selected [Assignment](#)

Parameters

<i>assignmentId</i>	
---------------------	--

Returns

Return [Assignment](#) object

3.2.2.2 GetAssignmentMonster()

```
Monster AssignmentBL.GetAssignmentMonster (
    int assignmentId )
```

Get [Monster](#) that belongs to selected [Assignment](#)

Parameters

<i>assignmentId</i>	
---------------------	--

Returns

Return [Monster](#) object

3.2.2.3 GetStudentAssignment()

```
List<PublishedAssignment> AssignmentBL.GetStudentAssignment (
    int studentId )
```

Get all [Student](#)'s published Assignments

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Return list of [PublishedAssignment](#) object

3.3 AssignmentDaoImpl Class Reference

Class to handle DAO operations for [Assignment](#)

Public Member Functions

- [Assignment](#) [GetAssignment](#) (int assignmentId)
Get selected [Assignment](#)
- List< [PublishedAssignment](#) > [GetStudentAssignment](#) (int studentId)
Get all published assignments
- [Monster](#) [GetAssignmentMonster](#) (int assignmentId)
Get [Monster](#) which belongs to the assignment

3.3.1 Detailed Description

Class to handle DAO operations for [Assignment](#)

3.3.2 Member Function Documentation

3.3.2.1 GetAssignment()

```
Assignment AssignmentDaoImpl.GetAssignment (
    int assignmentId )
```

Get selected [Assignment](#)

Parameters

<i>assignment↔ Id</i>	
---------------------------	--

Returns

Return [Assignment](#) object containing [Monster](#) object and [Question](#) object

3.3.2.2 GetAssignmentMonster()

```
Monster AssignmentDaoImpl.GetAssignmentMonster (
    int assignmentId )
```

Get [Monster](#) which belongs to the assignment

Parameters

<i>assignmentId</i>	
---------------------	--

Returns

Return [Monster](#) object based on assignmentId

3.3.2.3 GetStudentAssignment()

```
List<PublishedAssignment> AssignmentDaoImpl.GetStudentAssignment (
    int studentId )
```

Get all published assignments

Parameters

<i>studentId</i>	
------------------	--

Returns

Return list of published assignment object based on studentId

3.4 AssignmentScore Class Reference

DAO Object for [AssignmentScore](#)

Inherits Node.

Properties

- [Student](#) **Student** [get, set]
- int **Score** [get, set]
- [PublishedAssignment](#) **PublishedAssignment** [get, set]

3.4.1 Detailed Description

DAO Object for [AssignmentScore](#)

3.5 AssignmentScoreBL Class Reference

Class to handle Business Logic for [AssignmentScore](#)

Inherits Node.

Public Member Functions

- int [InsertAssignmentScore](#) (int studentId, int assignmentId, int timeRemaining, int timeLimit)
Insert [Student](#)'s score of completed [Assignment](#)
- List< [AssignmentScore](#) > [GetStudentCompletedAssignment](#) (int studentId)
Get all AssignmentScores of [Student](#)'s completed assignments

3.5.1 Detailed Description

Class to handle Business Logic for [AssignmentScore](#)

3.5.2 Member Function Documentation

3.5.2.1 GetStudentCompletedAssignment()

```
List<AssignmentScore> AssignmentScoreBL.GetStudentCompletedAssignment (
    int studentId )
```

Get all AssignmentScores of [Student](#)'s completed assignments

Parameters

<i>studentId</i>	
------------------	--

Returns

/Return list of [AssignmentScore](#) object

3.5.2.2 InsertAssignmentScore()

```
int AssignmentScoreBL.InsertAssignmentScore (
    int studentId,
    int assignmentId,
    int timeRemaining,
    int timeLimit )
```

Insert [Student](#)'s score of completed [Assignment](#)

Parameters

<i>studentId</i>	
<i>assignmentId</i>	
<i>timeRemaining</i>	
<i>timeLimit</i>	

Returns

Return int result 1 if InsertAssignmentScore has executed successfully

3.6 AssignmentScoreDaoImpl Class Reference

Class to handle DAO operations for [AssignmentScore](#)

Public Member Functions

- int [InsertAssignmentScore](#) (int studentId, int assignmentId, int assignmentScore)
Insert [Student](#)'s score of completed [Assignment](#)
- List< [AssignmentScore](#) > [GetStudentCompletedAssignment](#) (int studentId)
Get all [Student](#)'s completed [Assignment](#) scores

3.6.1 Detailed Description

Class to handle DAO operations for [AssignmentScore](#)

3.6.2 Member Function Documentation**3.6.2.1 GetStudentCompletedAssignment()**

```
List<AssignmentScore> AssignmentScoreDaoImpl.GetStudentCompletedAssignment (
    int studentId )
```

Get all [Student](#)'s completed [Assignment](#) scores

Parameters

<i>studentId</i>	
------------------	--

Returns

Return list of [AssignmentScore](#) object

3.6.2.2 InsertAssignmentScore()

```
int AssignmentScoreDaoImpl.InsertAssignmentScore (
    int studentId,
    int assignmentId,
    int assignmentScore )
```

Insert [Student](#)'s score of completed [Assignment](#)

Parameters

<i>studentId</i>	
<i>assignmentId</i>	
<i>assignmentScore</i>	

Returns

Return int 1 if insertion query executed successfully

3.7 AssignmentScreen Class Reference

Class to handle Presentation for [AssignmentScreen](#) (gameplay)

Inherits [Node2D](#).

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.7.1 Detailed Description

Class to handle Presentation for [AssignmentScreen](#) (gameplay)

3.7.2 Member Function Documentation

3.7.2.1 _Ready()

```
override void AssignmentScreen._Ready ( )
```

Initialization

3.8 BackButton Class Reference

Class to handle Presentation for [BackButton](#)

Inherits Node2D.

3.8.1 Detailed Description

Class to handle Presentation for [BackButton](#)

3.9 BaseDaoImpl< T > Class Template Reference

Class to handle DAO operations for Generic methods

Public Member Functions

- int [ExecuteQuery](#) (string query, Object obj)
Generic method to query execution using obj parameter
- int [ExecuteQuery](#) (string query)
Generic method for query execution
- T [RetrieveQuery](#) (string query)
Generic method for query sql command
- T [ExecuteScalar](#) (string query, Object obj)
Generic method to execute sql command
- T [ExecuteScalar](#) (string query)
Generic method to execute scalar sql command

3.9.1 Detailed Description

Class to handle DAO operations for Generic methods

Template Parameters

<i>T</i>	
----------	--

3.9.2 Member Function Documentation

3.9.2.1 ExecuteQuery() [1/2]

```
int BaseDaoImpl< T >.ExecuteQuery (
    string query )
```

Generic method for query execution

Parameters

<i>query</i>	
--------------	--

Returns

Return 1 if query has executed successfully

3.9.2.2 ExecuteQuery() [2/2]

```
int BaseDaoImpl< T >.ExecuteQuery (
    string query,
    Object obj )
```

Generic method to query execution using obj parameter

Parameters

<i>query</i>	
<i>t</i>	

Returns

Return 1 if query has executed successfully

3.9.2.3 ExecuteScalar() [1/2]

```
T BaseDaoImpl< T >.ExecuteScalar (
    string query )
```

Generic method to execute scalar sql command

Parameters

<i>query</i>	
--------------	--

Returns

Return generic object if query has executed successfully

3.9.2.4 ExecuteScalar() [2/2]

```
T BaseDaoImpl< T >.ExecuteScalar (
    string query,
    Object obj )
```

Generic method to execute sql command

Parameters

<i>query</i>	
<i>t</i>	

Returns

Return int result 1 if successful for scalar query with object passed into query

3.9.2.5 RetrieveQuery()

```
T BaseDaoImpl< T >.RetrieveQuery (
    string query )
```

Generic method for query sql command

Parameters

<i>query</i>	
--------------	--

Returns

Return generic object T if query has executed successfully

3.10 BelongClass Class Reference

DAO Object for [BelongClass](#)

Inherits Node.

Properties

- int **ClassId** [get, set]
- List< [Student](#) > **Student** [get, set]

3.10.1 Detailed Description

DAO Object for [BelongClass](#)

3.11 Campaign Class Reference

Class to handle Presentation for [Campaign](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.11.1 Detailed Description

Class to handle Presentation for [Campaign](#)

3.11.2 Member Function Documentation

3.11.2.1 [_Ready\(\)](#)

```
override void Campaign._Ready ( )
```

Initialization

3.12 CampaignBL Class Reference

Class to handle Business Logic for [Campaign](#)

Public Member Functions

- [Level](#) [GetLevel](#) (int worldId, int sectionId, int levelId)
Get [Level](#) of selected [World](#) and [Section](#)

3.12.1 Detailed Description

Class to handle Business Logic for [Campaign](#)

3.12.2 Member Function Documentation

3.12.2.1 GetLevel()

```
Level CampaignBL.GetLevel (
    int worldId,
    int sectionId,
    int levelId )
```

Get [Level](#) of selected [World](#) and [Section](#)

Parameters

<i>worldId</i>	
<i>sectionId</i>	
<i>levelId</i>	

Returns

Return [Level](#) object

3.13 CampaignDaoImpl Class Reference

Class to handle DAO operations for [Campaign](#)

Public Member Functions

- [Level](#) [GetLevel](#) (int worldId, int sectionId, int levelId)
Get level which belongs to selected [World](#) and [Section](#)

3.13.1 Detailed Description

Class to handle DAO operations for [Campaign](#)

3.13.2 Member Function Documentation

3.13.2.1 GetLevel()

```
Level CampaignDaoImpl.GetLevel (
    int worldId,
    int sectionId,
    int levelId )
```

Get level which belongs to selected [World](#) and [Section](#)

Parameters

<i>worldId</i>	
<i>sectionId</i>	
<i>levelId</i>	

Returns

Return [Level](#) object containing monster and question object

3.14 CampaignQuestion Class Reference

DAO Object for [CampaignQuestion](#)

Inherits Node.

Properties

- [Question](#) **Question** [get, set]

3.14.1 Detailed Description

DAO Object for [CampaignQuestion](#)

3.15 Character Class Reference

DAO Object for [Character](#)

Inherits Node.

Properties

- int **CharId** [get, set]
- string **CharName** [get, set]
- string **CharSkill** [get, set]
- string **SkillDescription** [get, set]

3.15.1 Detailed Description

DAO Object for [Character](#)

3.16 CharacterBL Class Reference

Class to handle Business Logic for [Character](#)

Public Member Functions

- [Character](#) **GetCharacter** (int studentId)
Get [Student's](#) [Character](#)
- List< [Character](#) > **GetAllCharacters** ()
Get all of the Characters
- List< [Monster](#) > **GetAllMonsters** ()
Get all of the Monsters

3.16.1 Detailed Description

Class to handle Business Logic for [Character](#)

3.16.2 Member Function Documentation

3.16.2.1 GetAllCharacters()

```
List<Character> CharacterBL.GetAllCharacters ( )
```

Get all of the Characters

Returns

Return list of [Character](#) object

3.16.2.2 GetAllMonsters()

```
List<Monster> CharacterBL.GetAllMonsters ( )
```

Get all of the Monsters

Returns

Return list of [Monster](#) object

3.16.2.3 GetCharacter()

```
Character CharacterBL.GetCharacter (
    int studentId )
```

Get [Student's Character](#)

Parameters

<i>studentId</i>	
------------------	--

Returns

Return [Character](#) object

3.17 CharacterDaoImpl Class Reference

Class to handle DAO operations for [Character](#)

Public Member Functions

- [Character GetCharacter](#) (int studentId)
Get [Student's](#) character
- List< [Character](#) > [GetAllCharacters](#) ()
Get all Characters
- List< [Monster](#) > [GetAllMonsters](#) ()
Get all Monsters

3.17.1 Detailed Description

Class to handle DAO operations for [Character](#)

3.17.2 Member Function Documentation

3.17.2.1 GetAllCharacters()

```
List<Character> CharacterDaoImpl.GetAllCharacters ( )
```

Get all Characters

Returns

Return list of [Character](#) object

3.17.2.2 GetAllMonsters()

```
List<Monster> CharacterDaoImpl.GetAllMonsters ( )
```

Get all Monsters

Returns

Return list of [Monster](#) object

3.17.2.3 GetCharacter()

```
Character CharacterDaoImpl.GetCharacter (
    int studentId )
```

Get [Student](#)'s character

Parameters

<i>studentId</i>	
------------------	--

Returns

Return [Character](#) object

3.18 CharSelect Class Reference

Class to handle Presentation for [Character](#) Select

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.18.1 Detailed Description

Class to handle Presentation for [Character](#) Select

3.18.2 Member Function Documentation

3.18.2.1 [_Ready\(\)](#)

```
override void CharSelect._Ready ( )
```

Initialization

3.19 Chart Class Reference

Class to handle Presentation for [Section](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.19.1 Detailed Description

Class to handle Presentation for [Section](#)

3.19.2 Member Function Documentation

3.19.2.1 [_Ready\(\)](#)

```
override void Chart._Ready ( )
```

Initialization

3.20 ClassGroup Class Reference

DAO Object for [ClassGroup](#)

Inherits Node.

Properties

- int **ClassId** [get, set]
- [Teacher](#) **Teacher** [get, set]

3.20.1 Detailed Description

DAO Object for [ClassGroup](#)

3.21 ClassGroupDaoImpl Class Reference

Class to handle DAO operations for [ClassGroup](#)

Public Member Functions

- List< [ClassGroup](#) > [GetClassGroup](#) (int teacherId)
Get all classes that belong to selected [Teacher](#)

3.21.1 Detailed Description

Class to handle DAO operations for [ClassGroup](#)

3.21.2 Member Function Documentation

3.21.2.1 GetClassGroup()

```
List<ClassGroup> ClassGroupDaoImpl.GetClassGroup (
    int teacherId )
```

Get all classes that belong to selected [Teacher](#)

Parameters

<i>teacherId</i>	
------------------	--

Returns

Return list of [ClassGroup](#) object

3.22 CreateLevel Class Reference

Class to handle Presentation for [CreateLevel](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()

Initialization

Static Public Member Functions

- static void [SetLevelInitInfo](#) (string name, int id, int time)

Get levelName, monsterId, and timeLimit from [CreateLevelInit](#)

3.22.1 Detailed Description

Class to handle Presentation for [CreateLevel](#)

3.22.2 Member Function Documentation

3.22.2.1 [_Ready\(\)](#)

```
override void CreateLevel._Ready ( )
```

Initialization

3.22.2.2 [SetLevelInitInfo\(\)](#)

```
static void CreateLevel.SetLevelInitInfo (
    string name,
    int id,
    int time ) [static]
```

Get levelName, monsterId, and timeLimit from [CreateLevelInit](#)

Parameters

<i>name</i>	
<i>id</i>	
<i>time</i>	

3.23 CreateLevelBL Class Reference

Class to handle Business Logic for [CreateLevel](#)

Inherits Node.

Public Member Functions

- List< [UserCreatedQuestion](#) > [getTempQuestionList](#) ()
Get the Questions that are saved temporarily
- void [ReloadTempQuestionList](#) ()
Reload the Questions that are save temporarily previously
- void [InitializeQuestions](#) ()
Intialize 5 [Question](#) objects
- void [SaveQuestion](#) (int questionId, string option1, string option2, string option3, string option4, int correct←
Option, string questionTitle)
Save current question to a List as temporary storage
- void [CreateLevel](#) (string levelName, int monsterId, int timeLimit)
Insert new level and all questions associated with it into database through DAO
- [UserCreatedQuestion](#) [GetQuestion](#) (int questionNumber)
Get [Question](#) object based on question number
- int [CheckEmptyFieldsExist](#) ()
Find [Question](#) with empty fields
- int [CheckDuplicationOptions](#) ()
Find question with duplication options
- void [ListQuestions](#) ()
List all questions in List
- void [AutoGenerateQuestions](#) (string levelName)

Static Public Member Functions

- static int [CheckValidLevelName](#) (string levelName)
Check if LevelName already exists

3.23.1 Detailed Description

Class to handle Business Logic for [CreateLevel](#)

3.23.2 Member Function Documentation

3.23.2.1 CheckDuplicationOptions()

```
int CreateLevelBL.CheckDuplicationOptions ( )
```

Find question with duplication options

Returns

Return the [Question](#) Number with duplicate options, else return -1 if no duplicate fields are found

3.23.2.2 CheckEmptyFieldsExist()

```
int CreateLevelBL.CheckEmptyFieldsExist ( )
```

Find [Question](#) with empty fields

Returns

Return the [Question](#) Number with empty fields, else return -1 if no empty fields are found

3.23.2.3 CheckValidLevelName()

```
static int CreateLevelBL.CheckValidLevelName (
    string levelName ) [static]
```

Check if LevelName already exists

Parameters

<i>levelName</i>	
------------------	--

Returns

Return 1 if there are no existing LevelName, else return -1 if there is an existing LevelName

3.23.2.4 CreateLevel()

```
void CreateLevelBL.CreateLevel (
    string levelName,
    int monsterId,
    int timeLimit )
```

Insert new level and all questions associated with it into database through DAO

Parameters

<i>levelName</i>	
<i>monsterId</i>	
<i>timeLimit</i>	

3.23.2.5 GetQuestion()

```
UserCreatedQuestion CreateLevelBL.GetQuestion (
    int questionNumber )
```

Get [Question](#) object based on question number

Parameters

<i>questionNumber</i>	
-----------------------	--

Returns

Return the acquired [Question](#) if it exists, else return null if it does not exist

3.23.2.6 getTempQuestionList()

```
List<UserCreatedQuestion> CreateLevelBL.getTempQuestionList ( )
```

Get the Questions that are saved temporarily

Returns

Return the list of Questions that are saved temporarily

3.23.2.7 InitializeQuestions()

```
void CreateLevelBL.InitializeQuestions ( )
```

Intialize 5 [Question](#) objects

3.23.2.8 ListQuestions()

```
void CreateLevelBL.ListQuestions ( )
```

List all questions in List

3.23.2.9 ReloadTempQuestionList()

```
void CreateLevelBL.ReloadTempQuestionList ( )
```

Reload the Questions that are save temporarily previously

3.23.2.10 SaveQuestion()

```
void CreateLevelBL.SaveQuestion (
    int questionId,
    string option1,
    string option2,
    string option3,
    string option4,
    int correctOption,
    string questionTitle )
```

Save current question to a List as temporary storage

Parameters

<i>questionId</i>	
<i>option1</i>	
<i>option2</i>	
<i>option3</i>	
<i>option4</i>	
<i>correctOption</i>	
<i>questionTitle</i>	

3.24 CreateLevelDAOImpl Class Reference

Class to handle DAO operations for [CreateLevel](#)

Public Member Functions

- void [InsertQuestion](#) (string option1, string option2, string option3, string option4, int correctOptionInt, string questionTitle)
Insert [Question](#) into database
- void [InsertCustomLevel](#) (string levelName, int monsterId, int timeLimit)
Insert new custom level into database

Static Public Member Functions

- static int [CheckValidLevelName](#) (string name)
Check database for existing [Level](#) Name

3.24.1 Detailed Description

Class to handle DAO operations for [CreateLevel](#)

3.24.2 Member Function Documentation

3.24.2.1 CheckValidLevelName()

```
static int CreateLevelDAOImpl::CheckValidLevelName (  
    string name ) [static]
```

Check database for existing [Level](#) Name

Parameters

<i>name</i>	
-------------	--

Returns

Return -1 if there is existing level name, else return 1

3.24.2.2 InsertCustomLevel()

```
void CreateLevelDAOImpl::InsertCustomLevel (  
    string levelName,
```

```
int monsterId,
int timeLimit )
```

Insert new custom level into database

Parameters

<i>levelName</i>	
<i>monsterId</i>	
<i>timeLimit</i>	

3.24.2.3 InsertQuestion()

```
void CreateLevelDAOImpl.InsertQuestion (
    string option1,
    string option2,
    string option3,
    string option4,
    int correctOptionInt,
    string questionTitle )
```

Insert [Question](#) into database

Parameters

<i>option1</i>	
<i>option2</i>	
<i>option3</i>	
<i>option4</i>	
<i>correctOptionInt</i>	
<i>questionTitle</i>	

3.25 CreateLevelInit Class Reference

Class to handle Presentation for [CreateLevelInit](#)

Inherits [Node2D](#).

Public Member Functions

- override void [_Ready](#) ()
Initialization

Static Public Attributes

- static int **updated** = 0

3.25.1 Detailed Description

Class to handle Presentation for [CreateLevelInit](#)

3.25.2 Member Function Documentation

3.25.2.1 `_Ready()`

```
override void CreateLevelInit._Ready ( )
```

Initialization

3.26 Credits Class Reference

Class to handle Presentation for [Credits](#)

Inherits [Node2D](#).

Public Member Functions

- `override void _Ready ()`
Initialization

3.26.1 Detailed Description

Class to handle Presentation for [Credits](#)

3.26.2 Member Function Documentation

3.26.2.1 `_Ready()`

```
override void Credits._Ready ( )
```

Initialization

3.27 CustomLevel Class Reference

DAO Object for [CustomLevel](#)

Inherits [Node](#).

Properties

- int **CustomLevelId** [get, set]
- [Student](#) **Student** [get, set]
- string **CustomLevelName** [get, set]
- [Monster](#) **Monster** [get, set]
- int **TimeLimit** [get, set]
- List< [Question](#) > **Question** [get, set]

3.27.1 Detailed Description

DAO Object for [CustomLevel](#)

3.28 CustomLevelBL Class Reference

Class to handle Business Logic for [CustomLevel](#)

Public Member Functions

- [CustomLevel](#) **GetCustomLevel** (int customLevelId)
Get selected [CustomLevel](#)
- List< [CustomLevel](#) > **GetCustomLevels** ()
Get all [CustomLevels](#)
- List< [CustomLevel](#) > **GetStudentCustomLevel** (int studentId)
Get all [Student](#)'s created [CustomLevels](#)
- List< [CustomLevelScore](#) > **GetClearedCustomLevels** (int studentId)
Get all of [Student](#)'s [CustomLevelScore](#)
- int **DeleteCustomLevel** (int customLevelId)
Delete [CustomLevel](#)
- [Monster](#) **GetCustomLevelMonster** (int customLevelId)
Get [Monster](#) that belongs to selected [CustomLevel](#)

3.28.1 Detailed Description

Class to handle Business Logic for [CustomLevel](#)

3.28.2 Member Function Documentation

3.28.2.1 DeleteCustomLevel()

```
int CustomLevelBL.DeleteCustomLevel (
    int customLevelId )
```

Delete [CustomLevel](#)

Parameters

<i>custom↔ LevelId</i>	
----------------------------	--

Returns

Return 1 if delete query has executed successfully

3.28.2.2 GetClearedCustomLevels()

```
List<CustomLevelScore> CustomLevelBL.GetClearedCustomLevels (
    int studentId )
```

Get all of Student's CustomLevelScore

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Return list of CustomLevelScore

3.28.2.3 GetCustomLevel()

```
CustomLevel CustomLevelBL.GetCustomLevel (
    int customLevelId )
```

Get selected CustomLevel

Parameters

<i>custom↔ LevelId</i>	
----------------------------	--

Returns

Return CustomLevel object

3.28.2.4 GetCustomLevelMonster()

```
Monster CustomLevelBL.GetCustomLevelMonster (
    int customLevelId )
```

Get [Monster](#) that belongs to selected [CustomLevel](#)

Parameters

<i>custom↔ LevelId</i>	
----------------------------	--

Returns

Return [Monster](#) object

3.28.2.5 GetCustomLevels()

```
List<CustomLevel> CustomLevelBL.GetCustomLevels ( )
```

Get all CustomLevels

Returns

Return list of [CustomLevel](#) object

3.28.2.6 GetStudentCustomLevel()

```
List<CustomLevel> CustomLevelBL.GetStudentCustomLevel (
    int studentId )
```

Get all [Student](#)'s created CustomLevels

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Return list of [CustomLevel](#) object

3.29 CustomLevelDaoImpl Class Reference

Class to handle DAO operations for [CustomLevel](#)

Public Member Functions

- [CustomLevel GetCustomLevel](#) (int customLevelId)
Get selected [CustomLevel](#)
- List< [CustomLevel](#) > [GetCustomLevels](#) ()
Get all custom levels
- List< [CustomLevelScore](#) > [GetClearedCustomLevels](#) (int studentId)
Get all custom levels which had been cleared by the [Student](#)
- List< [CustomLevel](#) > [GetStudentCustomLevel](#) (int studentId)
Get CustomLevels that the [Student](#) has created
- int [DeleteCustomLevel](#) (int customLevelId)
Delete selected [CustomLevel](#)
- [Monster GetCustomLevelMonster](#) (int customLevelId)
Get [Monster](#) that belongs to selected [CustomLevel](#)

3.29.1 Detailed Description

Class to handle DAO operations for [CustomLevel](#)

3.29.2 Member Function Documentation

3.29.2.1 DeleteCustomLevel()

```
int CustomLevelDaoImpl.DeleteCustomLevel (
    int customLevelId )
```

Delete selected [CustomLevel](#)

Parameters

<i>customLevelId</i>	
----------------------	--

Returns

Return 1 if delete query has executed successfully

3.29.2.2 GetClearedCustomLevels()

```
List<CustomLevelScore> CustomLevelDaoImpl.GetClearedCustomLevels (
    int studentId )
```

Get all custom levels which had been cleared by the [Student](#)

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Return list of [CustomLevelScore](#) object

3.29.2.3 GetCustomLevel()

```
CustomLevel CustomLevelDaoImpl.GetCustomLevel (
    int customLevelId )
```

Get selected [CustomLevel](#)

Parameters

<i>custom↔ LevelId</i>	
----------------------------	--

Returns

Return [CustomLevel](#) object containing monster and question object

3.29.2.4 GetCustomLevelMonster()

```
Monster CustomLevelDaoImpl.GetCustomLevelMonster (
    int customLevelId )
```

Get [Monster](#) that belongs to selected [CustomLevel](#)

Parameters

<i>custom↔ LevelId</i>	
----------------------------	--

Returns

Return [Monster](#) object

3.29.2.5 GetCustomLevels()

```
List<CustomLevel> CustomLevelDaoImpl.GetCustomLevels ( )
```

Get all custom levels

Returns

Return list of [CustomLevel](#) object

3.29.2.6 GetStudentCustomLevel()

```
List<CustomLevel> CustomLevelDaoImpl.GetStudentCustomLevel (
    int studentId )
```

Get CustomLevels that the [Student](#) has created

Parameters

<i>studentId</i>	
------------------	--

Returns

Return [CustomLevel](#) object

3.30 CustomLevelScore Class Reference

DAO Object for [CustomLevelScore](#)

Inherits Node.

Properties

- [Student](#) **Student** [get, set]
- [CustomLevel](#) **CustomLevel** [get, set]
- int **LevelScore** [get, set]

3.30.1 Detailed Description

DAO Object for [CustomLevelScore](#)

3.31 CustomLevelScoreBL Class Reference

Class to handle Business Logic for [CustomLevelScore](#)

Inherits Node.

Public Member Functions

- int [InsertCustomLevelScore](#) (int studentId, int customLevelId, int timeRemaining, int timeLimit)
Insert score of cleared [CustomLevel](#)

3.31.1 Detailed Description

Class to handle Business Logic for [CustomLevelScore](#)

3.31.2 Member Function Documentation

3.31.2.1 InsertCustomLevelScore()

```
int CustomLevelScoreBL.InsertCustomLevelScore (
    int studentId,
    int customLevelId,
    int timeRemaining,
    int timeLimit )
```

Insert score of cleared [CustomLevel](#)

Parameters

<i>studentId</i>	
<i>customLevelId</i>	
<i>timeRemaining</i>	
<i>timeLimit</i>	

Returns

Return 1 if insert query has executed successfully

3.32 CustomLevelScoreDaoImpl Class Reference

Class to handle DAO operations for [CustomLevelScore](#)

Public Member Functions

- int [InsertCustomLevelScore](#) (int studentId, int customLevelId, int levelScore)
Insert [Student's CustomLevelScore](#) of cleared [CustomLevel](#)

3.32.1 Detailed Description

Class to handle DAO operations for [CustomLevelScore](#)

3.32.2 Member Function Documentation

3.32.2.1 InsertCustomLevelScore()

```
int CustomLevelScoreDaoImpl.InsertCustomLevelScore (
    int studentId,
    int customLevelId,
    int levelScore )
```

Insert [Student's CustomLevelScore](#) of cleared [CustomLevel](#)

Parameters

<i>studentId</i>	
<i>custom↔ LevelId</i>	
<i>levelScore</i>	

<return>Return 1 if query has executed successfully

3.33 CustomLevelScreen Class Reference

Class to handle Presentation for [CustomLevelScreen](#) (gameplay)

Inherits [Node2D](#).

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.33.1 Detailed Description

Class to handle Presentation for [CustomLevelScreen](#) (gameplay)

3.33.2 Member Function Documentation

3.33.2.1 `_Ready()`

```
override void CustomLevelScreen._Ready ( )
```

Initialization

3.34 DefaultSound Class Reference

Class to handle Presentation for [DefaultSound](#)

Inherits [Node2D](#).

Public Member Functions

- override void [_Ready](#) ()
Initialization

Static Public Member Functions

- static void [disableSound](#) ()
Disable the sound
- static void [enableSound](#) ()
Enable the sound
- static void [playSound](#) (AudioStream audioStream)
Play the sound

Static Public Attributes

- static AudioStreamPlayer **audioPlayer**

3.34.1 Detailed Description

Class to handle Presentation for [DefaultSound](#)

3.34.2 Member Function Documentation

3.34.2.1 Ready()

```
override void DefaultSound._Ready ( )
```

Initialization

3.34.2.2 disableSound()

```
static void DefaultSound.disableSound ( ) [static]
```

Disable the sound

3.34.2.3 enableSound()

```
static void DefaultSound.enableSound ( ) [static]
```

Enable the sound

3.34.2.4 playSound()

```
static void DefaultSound.playSound (
    AudioStream audioStream ) [static]
```

Play the sound

Parameters

<i>audioStream</i>	
--------------------	--

3.35 EditLevel Class Reference

Class to handle Presentation for [EditLevel](#)

Inherits Node2D.

Public Member Functions

- [override void _Ready \(\)](#)
Initialization

Static Public Member Functions

- static void [SetLevelInitInfo](#) (string name, int id, int time)
Get levelName, monsterId, and timeLimit from [EditLevelInit](#)

3.35.1 Detailed Description

Class to handle Presentation for [EditLevel](#)

3.35.2 Member Function Documentation

3.35.2.1 `_Ready()`

```
override void EditLevel._Ready ( )
```

Initialization

3.35.2.2 `SetLevelInitInfo()`

```
static void EditLevel.SetLevelInitInfo (
    string name,
    int id,
    int time ) [static]
```

Get levelName, monsterId, and timeLimit from [EditLevelInit](#)

Parameters

<i>name</i>	
<i>id</i>	
<i>time</i>	

3.36 EditLevelBL Class Reference

Class to handle Business Logic for [EditLevel](#)

Inherits [Node](#).

Public Member Functions

- List< [UserCreatedQuestion](#) > [GetTempQuestionList](#) ()

- Get the Questions that are saved temporarily*

 - List< [UserCreatedQuestion](#) > [GetOriginalQuestionList](#) ()

Get the original Questions that have not been edited
- void [ReloadTempQuestionList](#) ()

Reload the Questions that are save temporarily previously
- void [SaveQuestion](#) (int questionId, string option1, string option2, string option3, string option4, int correct←Option, string questionTitle)

Save current [Question](#) to a List as temporary storage
- void [UpdateLevel](#) ()

Update questions in the selected [CustomLevel](#) into database through DAO
- [UserCreatedQuestion](#) [GetQuestion](#) (int questionNumber)

Get [Question](#) object based on [Question](#) Number
- int [CheckEmptyFieldsExist](#) ()

Find [Question](#) with empty fields
- [CustomLevel](#) [LoadCustomLevelInfo](#) ()

Load selected [CustomLevel](#) information from DAO
- int [CheckDuplicationOptions](#) ()

Find question with duplication options
- void [UpdateLevelInitInfo](#) (string levelName, int monsterId, int timeLimit)

Updates the LevelName, [Monster](#), and TimeLimit

Static Public Member Functions

- static int [CheckValidLevelName](#) (string oldName, string newName)

Check if LevelName already exists

3.36.1 Detailed Description

Class to handle Business Logic for [EditLevel](#)

3.36.2 Member Function Documentation

3.36.2.1 CheckDuplicationOptions()

```
int EditLevelBL.CheckDuplicationOptions ( )
```

Find question with duplication options

Returns

Return the [Question](#) Number with duplicate options, else return -1 if no duplicate fields are found

3.36.2.2 CheckEmptyFieldsExist()

```
int EditLevelBL.CheckEmptyFieldsExist ( )
```

Find [Question](#) with empty fields

Returns

Return the [Question](#) Number with empty fields, else return -1 if no empty fields are found

3.36.2.3 CheckValidLevelName()

```
static int EditLevelBL.CheckValidLevelName (
    string oldName,
    string newName ) [static]
```

Check if LevelName already exists

Parameters

<i>oldName</i>	
<i>newName</i>	

Returns

Return 1 if there are no existing LevelName, else return -1 if there is an existing LevelName

3.36.2.4 GetOriginalQuestionList()

```
List<UserCreatedQuestion> EditLevelBL.GetOriginalQuestionList ( )
```

Get the original Questions that have not been edited

Returns

Return the original Questions in a List

3.36.2.5 GetQuestion()

```
UserCreatedQuestion EditLevelBL.GetQuestion (
    int questionNumber )
```

Get [Question](#) object based on [Question](#) Number

Parameters

<i>questionNumber</i>	
-----------------------	--

Returns

Return the acquired [Question](#) if it exists, else return null if it does not exist

3.36.2.6 GetTempQuestionList()

```
List<UserCreatedQuestion> EditLevelBL.GetTempQuestionList ( )
```

Get the Questions that are saved temporarily

Returns

Return the list of Questions that are saved temporarily

3.36.2.7 LoadCustomLevelInfo()

```
CustomLevel EditLevelBL.LoadCustomLevelInfo ( )
```

Load selected [CustomLevel](#) information from DAO

Returns

Return the acquired information in a [CustomLevel](#) object

3.36.2.8 ReloadTempQuestionList()

```
void EditLevelBL.ReloadTempQuestionList ( )
```

Reload the Questions that are save temporarily previously

3.36.2.9 SaveQuestion()

```
void EditLevelBL.SaveQuestion (
    int questionId,
    string option1,
    string option2,
    string option3,
    string option4,
    int correctOption,
    string questionTitle )
```

Save current [Question](#) to a List as temporary storage

Parameters

<i>questionId</i>	
<i>option1</i>	
<i>option2</i>	
<i>option3</i>	
<i>option4</i>	
<i>correctOption</i>	
<i>questionTitle</i>	

3.36.2.10 UpdateLevel()

```
void EditLevelBL.UpdateLevel ( )
```

Update questions in the selected [CustomLevel](#) into database through DAO

3.36.2.11 UpdateLevelInitInfo()

```
void EditLevelBL.UpdateLevelInitInfo (
    string levelName,
    int monsterId,
    int timeLimit )
```

Updates the LevelName, [Monster](#), and TimeLimit

Parameters

<i>levelName</i>	
<i>monsterId</i>	
<i>timeLimit</i>	

3.37 EditLevelDaoImpl Class Reference

Class to handle DAO operations for [EditLevel](#)

Public Member Functions

- void [UpdateQuestion](#) (string option1, string option2, string option3, string option4, int correctOptionInt, string questionTitle, int questionId)
Update [Question](#) into database
- [CustomLevel GetLevelInfo](#) ()
Load all information on the [CustomLevel](#) from database
- void [UpdateLevelInitInfo](#) (string levelName, int monsterId, int timeLimit)
Updates the level name, monster, and time limit

Static Public Member Functions

- static int [CheckValidLevelName](#) (string oldName, string newName)
Check database for existing level name

3.37.1 Detailed Description

Class to handle DAO operations for [EditLevel](#)

3.37.2 Member Function Documentation

3.37.2.1 CheckValidLevelName()

```
static int EditLevelDaoImpl.CheckValidLevelName (  
    string oldName,  
    string newName ) [static]
```

Check database for existing level name

Parameters

<i>oldName</i>	
<i>newName</i>	

Returns

Return -1 if there is existing level name, else return 1

3.37.2.2 GetLevelInfo()

```
CustomLevel EditLevelDaoImpl.GetLevelInfo ( )
```

Load all information on the [CustomLevel](#) from database

Returns

Return the acquired information in a [CustomLevel](#) object

3.37.2.3 UpdateLevelInitInfo()

```
void EditLevelDaoImpl.UpdateLevelInitInfo (  
    string levelName,  
    int monsterId,  
    int timeLimit )
```

Updates the level name, monster, and time limit

Parameters

<i>levelName</i>	
<i>monsterId</i>	
<i>timeLimit</i>	

3.37.2.4 UpdateQuestion()

```
void EditLevelDaoImpl.UpdateQuestion (
    string option1,
    string option2,
    string option3,
    string option4,
    int correctOptionInt,
    string questionTitle,
    int questionId )
```

Update [Question](#) into database

Parameters

<i>option1</i>	
<i>option2</i>	
<i>option3</i>	
<i>option4</i>	
<i>correctOptionInt</i>	
<i>questionTitle</i>	
<i>questionId</i>	

3.38 EditLevelInit Class Reference

Class to handle Presentation for [EditLevelInit](#)

Inherits [Node2D](#).

Public Member Functions

- override void [_Ready](#) ()
Initialization

Static Public Attributes

- static int **updated** = 0

3.38.1 Detailed Description

Class to handle Presentation for [EditLevelInit](#)

3.38.2 Member Function Documentation

3.38.2.1 `_Ready()`

```
override void EditLevelInit._Ready ( )
```

Initialization

3.39 GamePlay Class Reference

Class to handle Presentation for [GamePlay](#)

Inherits [Node2D](#).

Public Member Functions

- override void [_Ready](#) ()
Initialization
- override void [_Process](#) (float delta)
Handles the countdown logic
- void [SetLevelTitle](#) (string title)
Set [Level](#) Title
- void [SetQuestionNum](#) ()
Set [Question](#) Number
- void [SetTimeLimit](#) (int timeLimit)
Set time limit
- void [SetQuestionList](#) (List< [Question](#) > questionList)
Set [Question](#) list
- void [DisplayQuestion](#) ()
Display the [Question](#)
- void [DisplayCharSprite](#) ([Character](#) character)
Display the [Student](#)'s character sprite
- void [DisplayMonsterSprite](#) ([Monster](#) monster)
Display the [Monster](#) sprite
- void [DisplayNextQuestion](#) ()
Display next [Question](#)
- bool [CheckCorrectAnswer](#) (string option)
Check if the answer is correct and play attack animations for Character/Monster and sound effects
- void [SetGameType](#) (string gameType)
Set the game type

- int [GetTimeLeft](#) ()
Get remaining time left
- void [SetCharacter](#) ([Character](#) character)
Set [Character](#)
- void [DisplayStars](#) (int star)
Display the stars
- void [SetBg](#) ()
Set the background
- void [LoadStart](#) ([Character](#) character, [Monster](#) monster)
Load the required images for [Character](#) and [Monster](#) for the VS preview that plays before the level actually starts

3.39.1 Detailed Description

Class to handle Presentation for [GamePlay](#)

3.39.2 Member Function Documentation

3.39.2.1 [_Process\(\)](#)

```
override void Gameplay._Process (
    float delta )
```

Handles the countdown logic

Parameters

<i>delta</i>	
--------------	--

3.39.2.2 [_Ready\(\)](#)

```
override void Gameplay._Ready ( )
```

Initialization

3.39.2.3 [CheckCorrectAnswer\(\)](#)

```
bool Gameplay.CheckCorrectAnswer (
    string option )
```

Check if the answer is correct and play attack animations for Character/Monster and sound effects

Parameters

<i>option</i>	
---------------	--

Returns

Return true if the answer, else return false if the answer is wrong

3.39.2.4 DisplayCharSprite()

```
void GamePlay.DisplayCharSprite (
    Character character )
```

Display the [Student](#)'s character sprite

Parameters

<i>character</i>	
------------------	--

3.39.2.5 DisplayMonsterSprite()

```
void GamePlay.DisplayMonsterSprite (
    Monster monster )
```

Display the [Monster](#) sprite

Parameters

<i>monster</i>	
----------------	--

3.39.2.6 DisplayNextQuestion()

```
void GamePlay.DisplayNextQuestion ( )
```

Display next [Question](#)

3.39.2.7 DisplayQuestion()

```
void GamePlay.DisplayQuestion ( )
```

Display the [Question](#)

3.39.2.8 DisplayStars()

```
void Gameplay.DisplayStars (
    int star )
```

Display the stars

Parameters

<i>star</i>	
-------------	--

3.39.2.9 GetTimeLeft()

```
int Gameplay.GetTimeLeft ( )
```

Get remaining time left

Returns

3.39.2.10 LoadStart()

```
void Gameplay.LoadStart (
    Character character,
    Monster monster )
```

Load the required images for [Character](#) and [Monster](#) for the VS preview that plays before the level actually starts

Parameters

<i>character</i>	
<i>monster</i>	

3.39.2.11 SetBg()

```
void Gameplay.SetBg ( )
```

Set the background

3.39.2.12 SetCharacter()

```
void GamePlay.SetCharacter (
    Character character )
```

Set [Character](#)

Parameters

<i>character</i>	
------------------	--

3.39.2.13 SetGameType()

```
void GamePlay.SetGameType (
    string gameType )
```

Set the game type

Parameters

<i>gameType</i>	
-----------------	--

3.39.2.14 SetLevelTitle()

```
void GamePlay.SetLevelTitle (
    string title )
```

Set [Level](#) Title

Parameters

<i>title</i>	
--------------	--

3.39.2.15 SetQuestionList()

```
void GamePlay.SetQuestionList (
    List< Question > questionList )
```

Set [Question](#) list

Parameters

<i>questionList</i>	
---------------------	--

3.39.2.16 SetQuestionNum()

```
void GamePlay.SetQuestionNum ( )
```

Set [Question](#) Number**3.39.2.17 SetTimeLimit()**

```
void GamePlay.SetTimeLimit (
    int timeLimit )
```

Set time limit

Parameters

<i>timeLimit</i>	
------------------	--

3.40 Global Class Reference[Global](#) class to store [Global](#) variables

Inherits Node.

Static Public Member Functions

- static int [CalculateScore](#) (int timeRemaining, int timeLimit)
Calculate Gameplay Score

Parameters

int timeRemaining	
int timeLimit	

- static int [GetFirstLoggedIn](#) ()
Get state of [Student](#)'s first logged in
- static void [SetFirstLoggedIn](#) (int status)
Store state of [Student](#)'s first logged in
- static void [SetStudentId](#) (int id)

Store StudentId

Parameters

int id	
--------	--

- static void [SetStudentName](#) (string name)

Store [Student](#) Name

Parameters

string name	
-------------	--

- static string [GetStudentName](#) ()
Get [Student](#) Name
- static void [SetGoogleLoggedIn](#) ()
Indicate that the user has logged in using Google Account
- static void [SetFbLoggedIn](#) ()
Indicate that the user has logged in using Facebook Account
- static void [LoadSprite](#) (string spritePath, AnimatedSprite animatedSprite, List< string > animationList)
Load the set of PNGs into SpriteFrames needed for the AnimatedSprite

Parameters

string spritePath	
AnimatedSprite animatedSprite	
List<string> animationList	

Static Public Attributes

- static MySqlConnectionStringBuilder [csb](#)
[Global](#) connection string

Properties

- static string [StudentName](#) [get, set]
[Global](#) variables
- static int **StudentId** [get, set]
- static int **WorldId** [get, set]
- static int **SectionId** [get, set]
- static int **LevelId** [get, set]
- static int **CustomLevelId** [get, set]
- static int **AssignmentId** [get, set]
- static string **TeacherName** [get, set]
- static int **TeacherId** [get, set]
- static string **CustomLevelName** [get, set]
- static string **AssignmentName** [get, set]
- static int **MonsterId** [get, set]
- static int **TimeLimit** [get, set]
- static bool **GoogleLoggedIn** [get, set]
- static bool **FbLoggedIn** [get, set]

- static List< [UserCreatedQuestion](#) > **QuestionList** [get, set]
- static float **BgmVol** [get, set]
- static float **BattleBgmVol** [get, set]
- static float **SfxVol** [get, set]
- static int **FirstLoggedIn** [get, set]

3.40.1 Detailed Description

[Global](#) class to store [Global](#) variables

3.40.2 Member Function Documentation

3.40.2.1 CalculateScore()

```
static int Global.CalculateScore (
    int timeRemaining,
    int timeLimit ) [static]
```

Calculate Gameplay Score

Parameters

<i>int timeRemaining</i>	
<i>int timeLimit</i>	

3.40.2.2 GetFirstLoggedIn()

```
static int Global.GetFirstLoggedIn ( ) [static]
```

Get state of [Student](#)'s first logged in

Returns

Return 1 if user has first logged in

3.40.2.3 GetStudentName()

```
static string Global.GetStudentName ( ) [static]
```

Get [Student](#) Name

Returns

Return StudentName

3.40.2.4 LoadSprite()

```
static void Global.LoadSprite (
    string spritePath,
    AnimatedSprite animatedSprite,
    List< string > animationList ) [static]
```

Load the set of PNGs into SpriteFrames needed for the AnimatedSprite

Parameters

<i>string spritePath</i>	
<i>AnimatedSprite animatedSprite</i>	
<i>List<string> animationList</i>	

3.40.2.5 SetFbLoggedIn()

```
static void Global.SetFbLoggedIn ( ) [static]
```

Indicate that the user has logged in using Facebook Account

3.40.2.6 SetFirstLoggedIn()

```
static void Global.SetFirstLoggedIn (
    int status ) [static]
```

Store state of [Student](#)'s first logged in

Parameters

<i>status</i>	
---------------	--

3.40.2.7 SetGoogleLoggedIn()

```
static void Global.SetGoogleLoggedIn ( ) [static]
```

Indicate that the user has logged in using Google Account

3.40.2.8 SetStudentId()

```
static void Global.SetStudentId (  
    int id ) [static]
```

Store StudentId

Parameters

<i>int id</i>	
---------------	--

3.40.2.9 SetStudentName()

```
static void Global.SetStudentName (  
    string name ) [static]
```

Store [Student](#) Name

Parameters

<i>string name</i>	
--------------------	--

3.40.3 Member Data Documentation**3.40.3.1 csb**

```
MySqlConnectionStringBuilder Global.csb [static]
```

Initial value:

```
= new MySqlConnectionStringBuilder  
{  
    Server = "35.198.238.34",  
    UserID = "root",  
    Password = "MpiPkr9y04xmg1lh",  
    Database = "AdventureLearn",  
    SslMode = MySqlSslMode.None,  
}
```

[Global](#) connection string

3.40.4 Property Documentation**3.40.4.1 StudentName**

```
string Global.StudentName [static], [get], [set]
```

[Global](#) variables

3.41 Leaderboard Class Reference

DAO Object for [Leaderboard](#)

Inherits Node.

Properties

- string **StudentName** [get, set]
- string **CharName** [get, set]
- int **TotalScore** [get, set]

3.41.1 Detailed Description

DAO Object for [Leaderboard](#)

3.42 LeaderboardBL Class Reference

Class to handle Business Logic for [Leaderboard](#)

Public Member Functions

- List< [Leaderboard](#) > [GetWorldLeaderboard](#) (int worldId)
Get all [Leaderboard](#) scores of selected [World](#)
- List< [Leaderboard](#) > [GetLeaderboards](#) ()
Get all [Leaderboard](#) score of all Worlds
- List< [World](#) > [GetWorlds](#) ()
Get all Worlds

3.42.1 Detailed Description

Class to handle Business Logic for [Leaderboard](#)

3.42.2 Member Function Documentation

3.42.2.1 GetLeaderboards()

```
List<Leaderboard> LeaderboardBL.GetLeaderboards ( )
```

Get all [Leaderboard](#) score of all Worlds

Returns

Return list of [Leaderboard](#) object

3.42.2.2 GetWorldLeaderboard()

```
List<Leaderboard> LeaderboardBL.GetWorldLeaderboard (
    int worldId )
```

Get all [Leaderboard](#) scores of selected [World](#)

Parameters

<i>worldId</i>	
<i>Id</i>	

Returns

Return list of [Leaderboard](#) object

3.42.2.3 GetWorlds()

```
List<World> LeaderboardBL.GetWorlds ( )
```

Get all Worlds

Returns

Return list of [World](#) object

3.43 LeaderboardDaoImpl Class Reference

Class to handle DAO operations for [Leaderboard](#)

Public Member Functions

- List< [Leaderboard](#) > [GetLeaderboardScore](#) ()
Get all [Leaderboard](#) scores
- List< [Leaderboard](#) > [GetLeaderboardScore](#) (int worldId)
Get all [Leaderboard](#) score on selected [World](#)

3.43.1 Detailed Description

Class to handle DAO operations for [Leaderboard](#)

3.43.2 Member Function Documentation

3.43.2.1 GetLeaderboardScore() [1/2]

```
List<Leaderboard> LeaderboardDaoImpl.GetLeaderboardScore ( )
```

Get all [Leaderboard](#) scores

Returns

Return list of [Leaderboard](#) object

3.43.2.2 GetLeaderboardScore() [2/2]

```
List<Leaderboard> LeaderboardDaoImpl.GetLeaderboardScore (
    int worldId )
```

Get all [Leaderboard](#) score on selected [World](#)

Parameters

<i>world</i> ↔	
<i>Id</i>	

Returns

Return list of leaderboard object

3.44 LeaderboardScreen Class Reference

Class to handle Presentation for [LeaderboardScreen](#)

Inherits Node.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.44.1 Detailed Description

Class to handle Presentation for [LeaderboardScreen](#)

3.44.2 Member Function Documentation

3.44.2.1 [_Ready\(\)](#)

```
override void LeaderboardScreen._Ready ( )
```

Initialization

3.45 Level Class Reference

DAO Object for [Level](#)

Inherits Node.

Properties

- int **LevelId** [get, set]
- **Monster** **Monster** [get, set]
- int **TimeLimit** [get, set]
- List< **Question** > **Question** [get, set]

3.45.1 Detailed Description

DAO Object for [Level](#)

3.46 LevelSelection Class Reference

Class to handle Presentation for [LevelSelection](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.46.1 Detailed Description

Class to handle Presentation for [LevelSelection](#)

3.46.2 Member Function Documentation

3.46.2.1 [_Ready\(\)](#)

```
override void LevelSelection._Ready ( )
```

Initialization

3.47 Login Class Reference

Class to handle Presentation for [Login](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.47.1 Detailed Description

Class to handle Presentation for [Login](#)

3.47.2 Member Function Documentation

3.47.2.1 [_Ready\(\)](#)

```
override void Login._Ready ( )
```

Initialization

3.48 MainMenu Class Reference

Class to handle Presentation for [Section](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.48.1 Detailed Description

Class to handle Presentation for [Section](#)

3.48.2 Member Function Documentation

3.48.2.1 [_Ready\(\)](#)

```
override void MainMenu._Ready ( )
```

Initialization

3.49 Monster Class Reference

DAO Object for [Monster](#)

Inherits Node.

Properties

- int **MonsterId** [get, set]
- string **MonsterName** [get, set]

3.49.1 Detailed Description

DAO Object for [Monster](#)

3.50 NotificationPopup Class Reference

Class to handle Presentation for [NotificationPopup](#)

Inherits Control.

Public Member Functions

- override void [_Ready](#) ()
Initialization

Static Public Member Functions

- static void [DisplayPopup](#) (string message)
Display the message in the parameter as popup message

Static Public Attributes

- static Control **node**
- static Label **msg**
- static AnimationPlayer **animations**

3.50.1 Detailed Description

Class to handle Presentation for [NotificationPopup](#)

3.50.2 Member Function Documentation

3.50.2.1 `_Ready()`

```
override void NotificationPopup._Ready ( )
```

Initialization

3.50.2.2 `DisplayPopup()`

```
static void NotificationPopup.DisplayPopup (
    string message ) [static]
```

Display the message in the parameter as popup message

Parameters

<i>message</i>	
----------------	--

3.51 PublishAssignmentScreen Class Reference

Class to handle Presentation for PublishAssignment

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.51.1 Detailed Description

Class to handle Presentation for PublishAssignment

3.51.2 Member Function Documentation

3.51.2.1 `_Ready()`

```
override void PublishAssignmentScreen._Ready ( )
```

Initialization

3.52 PublishedAssignment Class Reference

DAO Object for [PublishedAssignment](#)

Inherits [Node](#).

Properties

- [Assignment](#) **Assignment** [get, set]
- [ClassGroup](#) **ClassGroup** [get, set]
- [DateTime](#) **DueDate** [get, set]

3.52.1 Detailed Description

DAO Object for [PublishedAssignment](#)

3.53 PublishedAssignmentBL Class Reference

Class to handle Business Logic for [PublishedAssignment](#)

Public Member Functions

- int [InsertPublishedAssignment](#) (int assignmentId, int classId, DateTime dt)
Insert query to publish [Assignment](#)
- List< [ClassGroup](#) > [GetTeacherPublished](#) (int assignmentId, int teacherId)
Get all of [ClassGroup](#) that has been added to selected [Teacher's Assignment](#)
- List< [ClassGroup](#) > [GetClassNotAssignment](#) (int assignmentId, int teacherId)
Get classes that have not been assigned to selected [Assignment](#)

3.53.1 Detailed Description

Class to handle Business Logic for [PublishedAssignment](#)

3.53.2 Member Function Documentation

3.53.2.1 GetClassNotAssignment()

```
List<ClassGroup> PublishedAssignmentBL.GetClassNotAssignment (
    int assignmentId,
    int teacherId )
```

Get classes that have not been assigned to selected [Assignment](#)

Parameters

<i>assignment↔ Id</i>	
<i>teacherId</i>	

Returns

Return list of [ClassGroup](#) object

3.53.2.2 GetTeacherPublished()

```
List<ClassGroup> PublishedAssignmentBL.GetTeacherPublished (
    int assignmentId,
    int teacherId )
```

Get all of [ClassGroup](#) that has been added to selected [Teacher's Assignment](#)

Parameters

<i>assignment↔ Id</i>	
<i>teacherId</i>	

Returns

Return list of [ClassGroup](#) object

3.53.2.3 InsertPublishedAssignment()

```
int PublishedAssignmentBL.InsertPublishedAssignment (
    int assignmentId,
    int classId,
    DateTime dt )
```

Insert query to publish [Assignment](#)

Parameters

<i>assignment↔ Id</i>	
<i>classId</i>	
<i>dt</i>	

Returns

Return 1 if insert query has executed successfully

3.54 PublishedAssignmentDaoImpl Class Reference

Class to handle DAO operations for [PublishedAssignment](#)

Public Member Functions

- int [InsertPublishedAssignment](#) (int assignmentId, int classId, DateTime dt)
Insert query to publish [Assignment](#)
- List< [ClassGroup](#) > [GetTeacherPublished](#) (int assignmentId, int teacherId)
Get all of [ClassGroup](#) that has been added to selected [Teacher's Assignment](#)
- List< [ClassGroup](#) > [GetClassNotAssignment](#) (int assignmentId, int teacherId)
Get classes that have not been assigned to selected [Assignment](#)

3.54.1 Detailed Description

Class to handle DAO operations for [PublishedAssignment](#)

3.54.2 Member Function Documentation

3.54.2.1 [GetClassNotAssignment\(\)](#)

```
List<ClassGroup> PublishedAssignmentDaoImpl.GetClassNotAssignment (
    int assignmentId,
    int teacherId )
```

Get classes that have not been assigned to selected [Assignment](#)

Parameters

<i>assignmentId</i>	
<i>teacherId</i>	

Returns

Return list of [ClassGroup](#) object

3.54.2.2 GetTeacherPublished()

```
List<ClassGroup> PublishedAssignmentDaoImpl.GetTeacherPublished (
    int assignmentId,
    int teacherId )
```

Get all of [ClassGroup](#) that has been added to selected [Teacher's Assignment](#)

Parameters

<i>assignmentId</i>	
<i>teacherId</i>	

Returns

Return list of [ClassGroup](#) object

3.54.2.3 InsertPublishedAssignment()

```
int PublishedAssignmentDaoImpl.InsertPublishedAssignment (
    int assignmentId,
    int classId,
    DateTime dt )
```

Insert query to publish [Assignment](#)

Parameters

<i>assignmentId</i>	
<i>classId</i>	
<i>dt</i>	

Returns

Return 1 if insert query has executed successfully

3.55 Question Class Reference

DAO Object for [Question](#)

Inherits Node.

Public Member Functions

- **Question** (int QuestionId, string Option1, string Option2, string Option3, string CorrectOption, string QuestionTitle)
- **Question** (string Option1, string Option2, string Option3, string CorrectOption, string QuestionTitle)

Properties

- int **QuestionId** [get, set]
- string **Option1** [get, set]
- string **Option2** [get, set]
- string **Option3** [get, set]
- string **CorrectOption** [get, set]
- string **QuestionTitle** [get, set]

3.55.1 Detailed Description

DAO Object for [Question](#)

3.56 Section Class Reference

DAO Object for [Section](#)

Inherits Node.

Properties

- int **SectionId** [get, set]
- string **SectionName** [get, set]
- List< [Level](#) > **Level** [get, set]

3.56.1 Detailed Description

DAO Object for [Section](#)

3.57 SectionBL Class Reference

Class to handle Business Logic for [Section](#)

Public Member Functions

- List< [Section](#) > [GetWorldSections](#) (int worldId)
Get all [Section](#) that belongs to selected [World](#)
- [Section](#) [GetSectionLevels](#) (int worldId, int sectionId)
Get all [Levels](#) that belong to selected [Section](#)
- int [CheckSectionCleared](#) (int worldId, int sectionId, int studentId)
Check if [Student](#) has cleared the selected [World's](#) [Section](#)

3.57.1 Detailed Description

Class to handle Business Logic for [Section](#)

3.57.2 Member Function Documentation

3.57.2.1 CheckSectionCleared()

```
int SectionBL.CheckSectionCleared (
    int worldId,
    int sectionId,
    int studentId )
```

Check if [Student](#) has cleared the selected [World's Section](#)

Parameters

<i>worldId</i>	
<i>section↔ Id</i>	
<i>student↔ Id</i>	

Returns

Return int result 1 if [Student](#) has cleared the section

3.57.2.2 GetSectionLevels()

```
Section SectionBL.GetSectionLevels (
    int worldId,
    int sectionId )
```

Get all Levels that belong to selected [Section](#)

Parameters

<i>worldId</i>	
<i>section↔ Id</i>	

Returns

Return [Section](#) object containing list of [Level](#) object

3.57.2.3 GetWorldSections()

```
List<Section> SectionBL.GetWorldSections (
    int worldId )
```

Get all [Section](#) that belongs to selected [World](#)

Parameters

<i>worldId</i>	
<i>Id</i>	

Returns

Return list of [Section](#) object

3.58 SectionDaoImpl Class Reference

Class to handle DAO operations for SectionDao

Public Member Functions

- List< [Section](#) > [GetWorldSections](#) (int worldId)
Get all Sections of selected [World](#)
- [Section](#) [GetSectionLevels](#) (int worldId, int sectionId)
Get all levels that belongs to selected [Section](#)
- int [CheckSectionCleared](#) (int worldId, int sectionId, int studentId)
Check if [Student](#) has cleared the selected [World's](#) [Section](#)

3.58.1 Detailed Description

Class to handle DAO operations for SectionDao

3.58.2 Member Function Documentation

3.58.2.1 CheckSectionCleared()

```
int SectionDaoImpl.CheckSectionCleared (
    int worldId,
    int sectionId,
    int studentId )
```

Check if [Student](#) has cleared the selected [World's](#) [Section](#)

Parameters

<i>worldId</i>	
<i>section↔ Id</i>	
<i>student↔ Id</i>	

Returns

Return int result 1 if [Student](#) has cleared the section

3.58.2.2 GetSectionLevels()

```
Section SectionDaoImpl.GetSectionLevels (
    int worldId,
    int sectionId )
```

Get all levels that belongs to selected [Section](#)

Parameters

<i>worldId</i>	
<i>section↔ Id</i>	

Returns

Return [Section](#) object containing list of [Level](#) object

3.58.2.3 GetWorldSections()

```
List<Section> SectionDaoImpl.GetWorldSections (
    int worldId )
```

Get all Sections of selected [World](#)

Parameters

<i>world↔ Id</i>	
----------------------	--

Returns

Return list of [Section](#) objects

3.59 Settings Class Reference

Class to handle Presentation for [Settings](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()

Initialization

3.59.1 Detailed Description

Class to handle Presentation for [Settings](#)

3.59.2 Member Function Documentation

3.59.2.1 [_Ready\(\)](#)

```
override void Settings._Ready ( )
```

Initialization

3.60 StatisticsBL Class Reference

Class to handle Business Logic for Statistics

Inherits Node.

Public Member Functions

- List< [World](#) > [GetWorlds](#) ()
Get all Worlds
- List< [Section](#) > [GetSections](#) (int worldId)
Get all of selected [World](#)'s [Section](#)
- List< [Student](#) > [GetStudents](#) ()
Get all [Student](#)
- List< [StudentScore](#) > [GetStudentScores](#) (int worldId, int sectionId, int studentId)
Get all of level score of selected [Student](#)
- List< [StudentScore](#) > [GetAvgSectionScores](#) (int worldId)
Get average score in each [Section](#) of selected [World](#)
- [StudentScore](#) [GetMaxStudentScore](#) (int worldId, int sectionId, int studentId)
Get highest score of selected campaign for selected [Student](#)
- [StudentScore](#) [GetMinStudentScore](#) (int worldId, int sectionId, int studentId)
Get lowest score of selected campaign for selected [Student](#)
- [StudentScore](#) [GetAvgStudentScore](#) (int worldId, int sectionId, int studentId)
Get average score of selected campaign for selected [Student](#)
- [StudentScore](#) [GetAvgWorldScoresMax](#) ()
Get highest score in average score of each [World](#)
- [StudentScore](#) [GetAvgWorldScoresMin](#) ()
Get lowest score in average score of each [World](#)
- [StudentScore](#) [GetAvgWorldScoresAvg](#) ()
Get average score in average score of each [World](#)
- List< [StudentScore](#) > [GetAvgWorldScores](#) ()
Get average scores in each [World](#)
- [StudentScore](#) [GetAvgSectionScoresMax](#) (int worldId)
Get highest score in average score of selected [World](#)
- [StudentScore](#) [GetAvgSectionScoresMin](#) (int worldId)
Get lowest score in average score of selected [World](#)
- [StudentScore](#) [GetAvgSectionScoresAvg](#) (int worldId)
Get average score in average score of selected [World](#)
- List< [World](#) > [GetWorldSections](#) ()
Get all Worlds with their respective Sections
- List< [AssignmentScore](#) > [GetAvgAssignmentScore](#) ()
Get average of all [Assignment](#) scores
- [AssignmentScore](#) [GetAvgAssignmentScore](#) (int assignmentId)
Get average score of selected [Assignment](#)
- [AssignmentScore](#) [GetMaxAssignmentScore](#) (int assignmentId)
Get highest score of selected [Assignment](#)
- [AssignmentScore](#) [GetMinAssignmentScore](#) (int assignmentId)
Get lowest score of selected [Assignment](#)
- List< [Assignment](#) > [GetAssignments](#) ()
Get all Assignments
- List< [AssignmentScore](#) > [GetStudentAssignmentScores](#) (int studentId)
Get all of [Student](#)'s [Assignment](#) Score
- [AssignmentScore](#) [GetMinStudentAssignmentScores](#) (int studentId)
Get lowest score of selected [Student](#)'s [Assignment](#) scores
- [AssignmentScore](#) [GetMaxStudentAssignmentScores](#) (int studentId)
Get highest score of selected [Student](#)'s [Assignment](#) scores
- [AssignmentScore](#) [GetAvgStudentAssignmentScores](#) (int studentId)

- *Get average score of selected [Student](#)'s [Assignment](#) scores*
- `int GetStudentClass (int studentId)`
Get class of selected [Student](#)
- `List< StudentScore > GetStudentScores (int studentId)`
Get all [Student](#) Scores of selected [Student](#)
- `int GetCampaignRanking (int studentId)`
Get campaign ranking of selected [Student](#)
- `int GetAssignmentRanking (int studentId)`
Get assignment ranking of selected student

3.60.1 Detailed Description

Class to handle Business Logic for Statistics

3.60.2 Member Function Documentation

3.60.2.1 [GetAssignmentRanking\(\)](#)

```
int StatisticsBL.GetAssignmentRanking (
    int studentId )
```

Get assignment ranking of selected student

Parameters

<i>studentId</i>	
------------------	--

Returns

Return integer value of student's assignment rank

3.60.2.2 [GetAssignments\(\)](#)

```
List<Assignment> StatisticsBL.GetAssignments ( )
```

Get all Assignments

Returns

Return list of [Assignment](#) object

3.60.2.3 GetAvgAssignmentScore() [1/2]

```
List<AssignmentScore> StatisticsBL.GetAvgAssignmentScore ( )
```

Get average of all [Assignment](#) scores

Returns

Return list of [AssignmentScore](#) object

3.60.2.4 GetAvgAssignmentScore() [2/2]

```
AssignmentScore StatisticsBL.GetAvgAssignmentScore (
    int assignmentId )
```

Get average score of selected [Assignment](#)

Parameters

<i>assignmentId</i>	
---------------------	--

Returns

Return [AssignmentScore](#) object

3.60.2.5 GetAvgSectionScores()

```
List<StudentScore> StatisticsBL.GetAvgSectionScores (
    int worldId )
```

Get average score in each [Section](#) of selected [World](#)

Parameters

<i>worldId</i>	
----------------	--

Returns

Return list of [StudentScore](#) object

3.60.2.6 GetAvgSectionScoresAvg()

```
StudentScore StatisticsBL.GetAvgSectionScoresAvg (
    int worldId )
```

Get average score in average score of selected [World](#)

Parameters

<i>worldId</i>	
----------------	--

Returns

Return [StudentScore](#) object

3.60.2.7 GetAvgSectionScoresMax()

```
StudentScore StatisticsBL.GetAvgSectionScoresMax (
    int worldId )
```

Get highest score in average score of selected [World](#)

Parameters

<i>worldId</i>	
----------------	--

Returns

Return [StudentScore](#) object

3.60.2.8 GetAvgSectionScoresMin()

```
StudentScore StatisticsBL.GetAvgSectionScoresMin (
    int worldId )
```

Get lowest score in average score of selected [World](#)

Parameters

<i>worldId</i>	
----------------	--

Returns

Return [StudentScore](#) object

3.60.2.9 GetAvgStudentAssignmentScores()

```
AssignmentScore StatisticsBL.GetAvgStudentAssignmentScores (
    int studentId )
```

Get average score of selected [Student](#)'s [Assignment](#) scores

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Return [AssignmentScore](#) object

3.60.2.10 GetAvgStudentScore()

```
StudentScore StatisticsBL.GetAvgStudentScore (
    int worldId,
    int sectionId,
    int studentId )
```

Get average score of selected campaign for selected [Student](#)

Parameters

<i>worldId</i>	
<i>section↔ Id</i>	
<i>student↔ Id</i>	

Returns

Return [StudentScore](#) object

3.60.2.11 GetAvgWorldScores()

```
List<StudentScore> StatisticsBL.GetAvgWorldScores ( )
```

Get average scores in each [World](#)

Returns

Return list of [StudentScore](#)

3.60.2.12 GetAvgWorldScoresAvg()

```
StudentScore StatisticsBL.GetAvgWorldScoresAvg ( )
```

Get average score in average score of each [World](#)

Returns

Return [StudentScore](#) object

3.60.2.13 GetAvgWorldScoresMax()

```
StudentScore StatisticsBL.GetAvgWorldScoresMax ( )
```

Get highest score in average score of each [World](#)

Returns

Return [StudentScore](#) object

3.60.2.14 GetAvgWorldScoresMin()

```
StudentScore StatisticsBL.GetAvgWorldScoresMin ( )
```

Get lowest score in average score of each [World](#)

Returns

Return [StudentScore](#) object

3.60.2.15 GetCampaignRanking()

```
int StatisticsBL.GetCampaignRanking (
    int studentId )
```

Get campaign ranking of selected [Student](#)

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Return integer value of student's campaign rank

3.60.2.16 GetMaxAssignmentScore()

```
AssignmentScore StatisticsBL.GetMaxAssignmentScore (
    int assignmentId )
```

Get hightest score of selected [Assignment](#)

Parameters

<i>assignment↔ Id</i>	
---------------------------	--

Returns

Return [AssignmentScore](#) object

3.60.2.17 GetMaxStudentAssignmentScores()

```
AssignmentScore StatisticsBL.GetMaxStudentAssignmentScores (
    int studentId )
```

Get highest score of selected [Student's Assignment](#) scores

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Return [AssignmentScore](#) object

3.60.2.18 GetMaxStudentScore()

```
StudentScore StatisticsBL.GetMaxStudentScore (
    int worldId,
    int sectionId,
    int studentId )
```

Get highest score of selected campaign for selected [Student](#)

Parameters

<i>worldId</i>	
<i>sectionId</i>	
<i>studentId</i>	

Returns

Return [StudentScore](#) object

3.60.2.19 GetMinAssignmentScore()

```
AssignmentScore StatisticsBL.GetMinAssignmentScore (
    int assignmentId )
```

Get lowest score of selected [Assignment](#)

Parameters

<i>assignmentId</i>	
---------------------	--

Returns

Return [AssignmentScore](#) objec

3.60.2.20 GetMinStudentAssignmentScores()

```
AssignmentScore StatisticsBL.GetMinStudentAssignmentScores (
    int studentId )
```

Get lowest score of selected [Student](#)'s [Assignment](#) scores

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Return [AssignmentScore](#) object

3.60.2.21 GetMinStudentScore()

```
StudentScore StatisticsBL.GetMinStudentScore (
    int worldId,
    int sectionId,
    int studentId )
```

Get lowest score of selected campaign for selected [Student](#)

Parameters

<i>worldId</i>	
<i>section↔ Id</i>	
<i>student↔ Id</i>	

Returns

Return [StudentScore](#) object

3.60.2.22 GetSections()

```
List<Section> StatisticsBL.GetSections (
    int worldId )
```

Get all of selected [World's Section](#)

Parameters

<i>world↔ Id</i>	
----------------------	--

Returns

Return list of [Section](#) object

3.60.2.23 GetStudentAssignmentScores()

```
List<AssignmentScore> StatisticsBL.GetStudentAssignmentScores (
    int studentId )
```

Get all of [Student](#)'s [Assignment](#) Score

Parameters

<i>studentId</i>	
------------------	--

Returns

Return list of [AssignmentScore](#) object

3.60.2.24 GetStudentClass()

```
int StatisticsBL.GetStudentClass (
    int studentId )
```

Get class of selected [Student](#)

Parameters

<i>studentId</i>	
------------------	--

Returns

Return integer value of [Student](#)'s class

3.60.2.25 GetStudents()

```
List<Student> StatisticsBL.GetStudents ( )
```

Get all [Student](#)

Returns

Return list of [Student](#) object

3.60.2.26 GetStudentScores() [1/2]

```
List<StudentScore> StatisticsBL.GetStudentScores (
    int studentId )
```

Get all [Student](#) Scores of selected [Student](#)

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Get all [Student](#) Scores of selected [Student](#)

3.60.2.27 GetStudentScores() [2/2]

```
List<StudentScore> StatisticsBL.GetStudentScores (
    int worldId,
    int sectionId,
    int studentId )
```

Get all of level score of selected [Student](#)

Parameters

<i>worldId</i>	
<i>section↔ Id</i>	
<i>student↔ Id</i>	

Returns

Return list of [StudentScore](#) object

3.60.2.28 GetWorlds()

```
List<World> StatisticsBL.GetWorlds ( )
```

Get all Worlds

Returns

Return list of [World](#) object

3.60.2.29 GetWorldSections()

```
List<World> StatisticsBL.GetWorldSections ( )
```

Get all Worlds with their respective Sections

Returns

Return list of [World](#) object containing list of [Section](#) object

3.61 Student Class Reference

DAO Object for [Student](#)

Inherits Node.

Public Member Functions

- **Student** (string studentName, int charId, string studentEmail, string studentUsername, string studentPassword)

Properties

- int **StudentId** [get, set]
- string **StudentName** [get, set]
- int **CharId** [get, set]
- [Character](#) **Character** [get, set]
- string **StudentEmail** [get, set]
- string **StudentUserName** [get, set]
- string **StudentPassword** [get, set]
- string **Hash** [get, set]
- string **Salt** [get, set]
- List< [StudentScore](#) > **StudentScore** [get, set]

3.61.1 Detailed Description

DAO Object for [Student](#)

3.62 StudentBL Class Reference

Class to handle Business Logic for [Student](#)

Inherits Node.

Public Member Functions

- bool [CheckStudentExist](#) (int studentId)
Check if [Student](#) has an existing account
- bool [CheckStudentCharExist](#) (int studentId)
Check if [Student](#) has an existing account
- bool [CheckGoogleCharExist](#) (string googleId)
Check if [Student](#) who has logged in with Google account has an existing [Character](#)
- bool [CheckFacebookCharExist](#) (string fbId)
Check if [Student](#) who has logged in with Facebook account has an existing Chrracter
- bool [CheckGoogleExist](#) (string googleId)
Check if there is an existing [Student](#)'s Google account
- bool [CheckFacebookExist](#) (string googleId)
Check if there is an existing [Student](#)'s Facebook account
- int [InsertGoogleStudent](#) (string studentName, string studentEmail, string googleId)
Insert [Student](#) record who logged in through Google
- int [InsertFacebookStudent](#) (string studentName, string studentEmail, string fbId)
Insert [Student](#) record who logged in through Facebook
- int [GetGoogleStudentId](#) (string googleId)
Get existing [Student](#)'s Google account
- int [GetFacebookStudentId](#) (string fbId)
Get existing [Student](#)'s Facebook account
- int [UpdateStudentCharacter](#) (int charId, int studentId)
Updates [Student](#)'s [Character](#)
- [Student](#) [GetStudentCharacter](#) (int studentId)
Get [Character](#) that belongs to selected [Student](#)

3.62.1 Detailed Description

Class to handle Business Logic for [Student](#)

3.62.2 Member Function Documentation

3.62.2.1 CheckFacebookCharExist()

```
bool StudentBL.CheckFacebookCharExist (
    string fbId )
```

Check if [Student](#) who has logged in with Facebook account has an existing Chrracter

Parameters

$fb \leftrightarrow$	
Id	

Returns

Return true if [Student](#) has existing [Character](#)

3.62.2.2 CheckFacebookExist()

```
bool StudentBL.CheckFacebookExist (
    string googleId )
```

Check if there is an existing [Student](#)'s Facebook account

Parameters

<i>google↵ Id</i>	
-----------------------	--

Returns

Return true if [Student](#)'s Facebook account exists

3.62.2.3 CheckGoogleCharExist()

```
bool StudentBL.CheckGoogleCharExist (
    string googleId )
```

Check if [Student](#) who has logged in with Google account has an existing [Character](#)

Parameters

<i>google↵ Id</i>	
-----------------------	--

Returns

Return true if [Student](#) has existing [Character](#)

3.62.2.4 CheckGoogleExist()

```
bool StudentBL.CheckGoogleExist (
    string googleId )
```

Check if there is an existing [Student](#)'s Google account

Parameters

<i>google↔</i> <i>Id</i>	
-----------------------------	--

Returns

Return true if [Student](#)'s Google account exists

3.62.2.5 CheckStudentCharExist()

```
bool StudentBL.CheckStudentCharExist (
    int studentId )
```

Check if [Student](#) has an existing account

Parameters

<i>student↔</i> <i>Id</i>	
------------------------------	--

Returns

Return true if [Student](#)'s record exist

3.62.2.6 CheckStudentExist()

```
bool StudentBL.CheckStudentExist (
    int studentId )
```

Check if [Student](#) has an existing account

Parameters

<i>student↔</i> <i>Id</i>	
------------------------------	--

Returns

Return true if [Student](#)'s record exist

3.62.2.7 GetFacebookStudentId()

```
int StudentBL.GetFacebookStudentId (
    string fbId )
```

Get existing [Student](#)'s Facebook account

Parameters

<i>fb↔</i> <i>Id</i>	
-------------------------	--

Returns

Retrun [Student](#) Object

3.62.2.8 GetGoogleStudentId()

```
int StudentBL.GetGoogleStudentId (
    string googleId )
```

Get existing [Student](#)'s Google account

Parameters

<i>google↔</i> <i>Id</i>	
-----------------------------	--

Returns

Return [Student](#) Object

3.62.2.9 GetStudentCharacter()

```
Student StudentBL.GetStudentCharacter (
    int studentId )
```

Get [Character](#) that belongs to selected [Student](#)

Parameters

<i>student↔</i> <i>Id</i>	
------------------------------	--

Returns

Return [Student](#) object containing [Character](#) object

3.62.2.10 InsertFacebookStudent()

```
int StudentBL.InsertFacebookStudent (
    string studentName,
    string studentEmail,
    string fbId )
```

Insert [Student](#) record who logged in through Facebook

Parameters

<i>studentName</i>	
<i>studentEmail</i>	
<i>fbId</i>	

Returns

Return int result 1 if insertion query has executed successfully

3.62.2.11 InsertGoogleStudent()

```
int StudentBL.InsertGoogleStudent (
    string studentName,
    string studentEmail,
    string googleId )
```

Insert [Student](#) record who logged in through Google

Parameters

<i>studentName</i>	
<i>charId</i>	
<i>studentEmail</i>	
<i>studentUsername</i>	
<i>studentPassword</i>	

Returns

Return int result 1 if insertion query has executed successfully

3.62.2.12 UpdateStudentCharacter()

```
int StudentBL.UpdateStudentCharacter (
    int charId,
    int studentId )
```

Updates [Student's Character](#)

Parameters

<i>charId</i>	
<i>student↔ Id</i>	

Returns

Return 1 if update query has executed successfully

3.63 StudentCustomQuestion Class Reference

DAO Object for [StudentCustomQuestion](#)

Inherits Node.

Properties

- [Question Question](#) [get, set]

3.63.1 Detailed Description

DAO Object for [StudentCustomQuestion](#)

3.64 StudentDaoImpl Class Reference

Class to handle DAO operations for [Student](#)

Public Member Functions

- int [InsertGoogleStudent](#) (string studentName, string studentEmail, string googleId)
Insert [Student](#) record who logged in through Google
- int [InsertFacebookStudent](#) (string studentName, string studentEmail, string fbId)
Insert [Student](#) record who logged in through Facebook
- bool [CheckStudentExist](#) (int studentId)
Check if [Student](#) has an existing account
- bool [CheckGoogleCharExist](#) (string googleId)
Check if [Student](#) who has logged in with Google account has an existing [Character](#)
- bool [CheckFacebookCharExist](#) (string fbId)
Check if [Student](#) who has logged in with Facebook account has an existing Chrracter
- bool [CheckGoogleExist](#) (string googleId)
Check if there is an existing [Student](#)'s Google account
- bool [CheckFacebookExist](#) (string fbId)
Check if there is an existing [Student](#)'s Facebook account
- [Student](#) [GetFacebookStudent](#) (string fbId)
Get existing [Student](#)'s Facebook account
- [Student](#) [GetGoogleStudent](#) (string googleId)
Get existing [Student](#)'s Google account
- int [UpdateStudentCharacter](#) (int charId, int studentId)
Updates [Student](#)'s [Character](#)
- [Student](#) [GetStudentCharacter](#) (int studentId)
Get [Character](#) that belongs to selected [Student](#)

3.64.1 Detailed Description

Class to handle DAO operations for [Student](#)

3.64.2 Member Function Documentation

3.64.2.1 CheckFacebookCharExist()

```
bool StudentDaoImpl.CheckFacebookCharExist (
    string fbId )
```

Check if [Student](#) who has logged in with Facebook account has an existing Chrracter

Parameters

<i>fb↔</i>	
<i>Id</i>	

Returns

Return true if [Student](#) has existing [Character](#)

3.64.2.2 CheckFacebookExist()

```
bool StudentDaoImpl.CheckFacebookExist (
    string fbId )
```

Check if there is an existing [Student](#)'s Facebook account

Parameters

<i>fb↔</i>	
<i>Id</i>	

Returns

Return true if [Student](#)'s Facebook account exists

3.64.2.3 CheckGoogleCharExist()

```
bool StudentDaoImpl.CheckGoogleCharExist (
    string googleId )
```

Check if [Student](#) who has logged in with Google account has an existing [Character](#)

Parameters

<i>google↔</i>	
<i>Id</i>	

Returns

Return true if [Student](#) has existing [Character](#)

3.64.2.4 CheckGoogleExist()

```
bool StudentDaoImpl.CheckGoogleExist (
    string googleId )
```

Check if there is an existing [Student](#)'s Google account

Parameters

<i>google↔</i> <i>Id</i>	
-----------------------------	--

Returns

Return true if [Student](#)'s Google account exists

3.64.2.5 CheckStudentExist()

```
bool StudentDaoImpl.CheckStudentExist (
    int studentId )
```

Check if [Student](#) has an existing account

Parameters

<i>student↔</i> <i>Id</i>	
------------------------------	--

Returns

Return true if [Student](#)'s record exist

3.64.2.6 GetFacebookStudent()

```
Student StudentDaoImpl.GetFacebookStudent (
    string fbId )
```

Get existing [Student](#)'s Facebook account

Parameters

<i>fb↔</i> <i>Id</i>	
-------------------------	--

Returns

Retrun [Student](#) Object

3.64.2.7 GetGoogleStudent()

```
Student StudentDaoImpl.GetGoogleStudent (
    string googleId )
```

Get existing [Student](#)'s Google account

Parameters

<i>googleId</i>	
-----------------	--

Returns

Return [Student](#) Object

3.64.2.8 GetStudentCharacter()

```
Student StudentDaoImpl.GetStudentCharacter (
    int studentId )
```

Get [Character](#) that belongs to selected [Student](#)

Parameters

<i>studentId</i>	
------------------	--

Returns

Return [Student](#) object containing [Character](#) object

3.64.2.9 InsertFacebookStudent()

```
int StudentDaoImpl.InsertFacebookStudent (
    string studentName,
    string studentEmail,
    string fbId )
```

Insert [Student](#) record who logged in through Facebook

Parameters

<i>studentName</i>	
<i>studentEmail</i>	
<i>fbId</i>	

Returns

Return int result 1 if insertion query has executed successfully

3.64.2.10 InsertGoogleStudent()

```
int StudentDaoImpl.InsertGoogleStudent (
    string studentName,
    string studentEmail,
    string googleId )
```

Insert [Student](#) record who logged in through Google

Parameters

<i>studentName</i>	
<i>charId</i>	
<i>studentEmail</i>	
<i>studentUsername</i>	
<i>studentPassword</i>	

Returns

Return int result 1 if insertion query has executed successfully

3.64.2.11 UpdateStudentCharacter()

```
int StudentDaoImpl.UpdateStudentCharacter (
    int charId,
    int studentId )
```

Updates [Student's Character](#)

Parameters

<i>charId</i>	
<i>student↔ Id</i>	

Returns

Return 1 if update query has executed successfully

3.65 StudentScore Class Reference

DAO Object for [StudentScore](#)

Inherits Node.

Properties

- string **StudentId** [get, set]
- int **WorldId** [get, set]
- int **SectionId** [get, set]
- int **LevelId** [get, set]
- int **LevelScore** [get, set]

3.65.1 Detailed Description

DAO Object for [StudentScore](#)

3.66 StudentScoreBL Class Reference

Class to handle Business Logic for [StudentScore](#)

Public Member Functions

- [Student GetStudentScores](#) (int worldId, int sectionId, int studentId)
Get all [Student](#) scores in selected [World](#) and [Section](#)
- int [InsertStudentScore](#) (int studentId, int worldId, int sectionId, int levelId, int timeRemaining, int timeLimit)
Insert [StudentScore](#) of cleared level
- [StudentScore GetAvgWorldScores](#) (int studentId)
Get [Student](#)'s average score in all Worlds
- int [GetCampaignRanking](#) (int studentId)
Get [Student](#)'s campaign rank

3.66.1 Detailed Description

Class to handle Business Logic for [StudentScore](#)

3.66.2 Member Function Documentation

3.66.2.1 GetAvgWorldScores()

```
StudentScore StudentScoreBL.GetAvgWorldScores (
    int studentId )
```

Get [Student](#)'s average score in all Worlds

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Return list of [StudentScore](#) object

3.66.2.2 GetCampaignRanking()

```
int StudentScoreBL.GetCampaignRanking (
    int studentId )
```

Get [Student](#)'s campaign rank

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Return integer result of [Student](#)'s campaign rank

3.66.2.3 GetStudentScores()

```
Student StudentScoreBL.GetStudentScores (
    int worldId,
    int sectionId,
    int studentId )
```

Get all [Student](#) scores in selected [World](#) and [Section](#)

Parameters

<i>section↔ Id</i>	
<i>student↔ Id</i>	

Returns

Return [Student](#) object containing [StudentScore](#) object

3.66.2.4 InsertStudentScore()

```
int StudentScoreBL.InsertStudentScore (
    int studentId,
    int worldId,
    int sectionId,
    int levelId,
    int timeRemaining,
    int timeLimit )
```

Insert [StudentScore](#) of cleared level

Parameters

<i>studentId</i>	
<i>worldId</i>	
<i>sectionId</i>	
<i>levelId</i>	
<i>timeRemaining</i>	
<i>timeLimit</i>	

Returns

Return int result 1 if insertion query has executed successfully

3.67 StudentScoreDaoImpl Class Reference

Class to handle DAO operations for [StudentScore](#)

Public Member Functions

- [Student GetStudentScores](#) (int worldId, int sectionId, int studentId)
Get all [Student](#) scores in selected [World](#) and [Section](#)
- [StudentScore GetAvgWorldScores](#) (int studentId)
Get [Student](#)'s average score in all [Worlds](#)
- int [InsertStudentScore](#) (int studentId, int worldId, int sectionId, int levelId, int levelScore)
Insert [StudentScore](#) of cleared level
- int [GetCampaignRanking](#) (int studentId)
Get [Student](#)'s campaign rank

3.67.1 Detailed Description

Class to handle DAO operations for [StudentScore](#)

3.67.2 Member Function Documentation

3.67.2.1 GetAvgWorldScores()

```
StudentScore StudentScoreDaoImpl.GetAvgWorldScores (
    int studentId )
```

Get [Student](#)'s average score in all Worlds

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Return list of [StudentScore](#) object

3.67.2.2 GetCampaignRanking()

```
int StudentScoreDaoImpl.GetCampaignRanking (
    int studentId )
```

Get [Student](#)'s campaign rank

Parameters

<i>student↔ Id</i>	
------------------------	--

Returns

Return integer result of [Student](#)'s campaign rank

3.67.2.3 GetStudentScores()

```
Student StudentScoreDaoImpl.GetStudentScores (
    int worldId,
    int sectionId,
    int studentId )
```

Get all [Student](#) scores in selected [World](#) and [Section](#)

Parameters

<i>section↔ Id</i>	
<i>student↔ Id</i>	

Returns

Return [Student](#) object containing [StudentScore](#) object

3.67.2.4 InsertStudentScore()

```
int StudentScoreDaoImpl.InsertStudentScore (
    int studentId,
    int worldId,
    int sectionId,
    int levelId,
    int levelScore )
```

Insert [StudentScore](#) of cleared level

Parameters

<i>studentId</i>	
<i>worldId</i>	
<i>sectionId</i>	
<i>levelId</i>	
<i>levelScore</i>	

Returns

Return int result 1 if insertion query has executed successfully

3.68 Teacher Class Reference

DAO Object for [Teacher](#)

Inherits Node.

Properties

- int **TeacherId** [get, set]
- string **TeacherName** [get, set]
- string **TeacherEmail** [get, set]

3.68.1 Detailed Description

DAO Object for [Teacher](#)

3.69 TeacherAssignment Class Reference

Class to handle Presentation for [AssignmentScreen](#) (gameplay)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.69.1 Detailed Description

Class to handle Presentation for [AssignmentScreen](#) (gameplay)

3.69.2 Member Function Documentation

3.69.2.1 [_Ready\(\)](#)

```
override void TeacherAssignment._Ready ( )
```

Initialization

3.70 TeacherBL Class Reference

Class to handle Business Logic for [Teacher](#)

Inherits [Node](#).

Public Member Functions

- bool [CheckGoogleLogin](#) (string email, string id)
Check if there is an existing [Teacher](#)'s Google account
- bool [CheckFacebookLogin](#) (string email, string id)
Check if there is an existing [Teacher](#)'s Facebook account
- bool [CheckEmail](#) (string email)
Check if [Teacher](#)'s email exist
- int [UpdateFacebookId](#) (string email, string id)
Update [Teacher](#)'s FacebookId
- int [UpdateGoogleId](#) (string email, string id)
Update [Teacher](#)'s GoogleId
- int [GetFacebookTeacherId](#) (string id)
Get [Teacher](#) with selected FacebookId
- int [GetGoogleTeacherId](#) (string id)
Get [Teacher](#) with selected GoogleId

3.70.1 Detailed Description

Class to handle Business Logic for [Teacher](#)

3.70.2 Member Function Documentation

3.70.2.1 CheckEmail()

```
bool TeacherBL.CheckEmail (
    string email )
```

Check if [Teacher](#)'s email exist

Parameters

<i>email</i>	
--------------	--

Returns

Return true if [Teacher](#)'s email exist

3.70.2.2 CheckFacebookLogin()

```
bool TeacherBL.CheckFacebookLogin (
    string email,
    string id )
```

Check if there is an existing [Teacher](#)'s Facebook account

Parameters

<i>email</i>	
<i>id</i>	

Returns

Return true if [Teacher](#)'s Facebook account exist

3.70.2.3 CheckGoogleLogin()

```
bool TeacherBL.CheckGoogleLogin (
    string email,
    string id )
```

Check if there is an existing [Teacher](#)'s Google account

Parameters

<i>userName</i>	
<i>password</i>	

Returns

Return true if [Teacher](#)'s Google account exist

3.70.2.4 GetFacebookTeacherId()

```
int TeacherBL.GetFacebookTeacherId (  
    string id )
```

Get [Teacher](#) with selected FacebookId

Parameters

<i>id</i>	
-----------	--

Returns

Return [Teacher](#) object

3.70.2.5 GetGoogleTeacherId()

```
int TeacherBL.GetGoogleTeacherId (  
    string id )
```

Get [Teacher](#) with selected GoogleId

Parameters

<i>id</i>	
-----------	--

Returns

Return [Teacher](#) object

3.70.2.6 UpdateFacebookId()

```
int TeacherBL.UpdateFacebookId (  

```

```
string email,  
string id )
```

Update [Teacher](#)'s FacebookId

Parameters

<i>email</i>	
<i>id</i>	

Returns

Return 1 if update query has executed successfully

3.70.2.7 UpdateGoogleId()

```
int TeacherBL.UpdateGoogleId (  
    string email,  
    string id )
```

Update [Teacher](#)'s GoogleId

Parameters

<i>email</i>	
<i>id</i>	

Returns

Return 1 if update query has executed successfully

3.71 TeacherCustomQuestion Class Reference

DAO Object for [TeacherCustomQuestion](#)

Inherits Node.

Properties

- int **AssignmentId** [get, set]
- [Question](#) **Question** [get, set]

3.71.1 Detailed Description

DAO Object for [TeacherCustomQuestion](#)

3.72 TeacherDaoImpl Class Reference

Class to handle DAO operations for [Teacher](#)

Public Member Functions

- bool [CheckGoogleLogin](#) (string email, string id)
Check if there is an existing [Teacher](#)'s Google account
- bool [CheckFacebookLogin](#) (string email, string id)
Check if there is an existing [Teacher](#)'s Facebook account
- bool [CheckEmail](#) (string email)
Check if [Teacher](#)'s email exist
- int [UpdateFacebookId](#) (string email, string id)
Update [Teacher](#)'s FacebookId
- int [UpdateGoogleId](#) (string email, string id)
Update [Teacher](#)'s GoogleId
- [Teacher](#) [GetGoogleTeacher](#) (string id)
Get [Teacher](#) with selected GoogleId
- [Teacher](#) [GetFacebookTeacher](#) (string id)
Get [Teacher](#) with selected FacebookId

3.72.1 Detailed Description

Class to handle DAO operations for [Teacher](#)

3.72.2 Member Function Documentation

3.72.2.1 CheckEmail()

```
bool TeacherDaoImpl.CheckEmail (  
    string email )
```

Check if [Teacher](#)'s email exist

Parameters

<i>email</i>	
--------------	--

Returns

Return true if [Teacher](#)'s email exist

3.72.2.2 CheckFacebookLogin()

```
bool TeacherDaoImpl.CheckFacebookLogin (
    string email,
    string id )
```

Check if there is an existing [Teacher](#)'s Facebook account

Parameters

<i>email</i>	
<i>id</i>	

Returns

Return true if [Teacher](#)'s Facebook account exist

3.72.2.3 CheckGoogleLogin()

```
bool TeacherDaoImpl.CheckGoogleLogin (
    string email,
    string id )
```

Check if there is an existing [Teacher](#)'s Google account

Parameters

<i>userName</i>	
<i>password</i>	

Returns

Return true if [Teacher](#)'s Google account exist

3.72.2.4 GetFacebookTeacher()

```
Teacher TeacherDaoImpl.GetFacebookTeacher (
    string id )
```

Get [Teacher](#) with selected FacebookId

Parameters

<i>id</i>	
-----------	--

Returns

Return [Teacher](#) object

3.72.2.5 GetGoogleTeacher()

```
Teacher TeacherDaoImpl.GetGoogleTeacher (
    string id )
```

Get [Teacher](#) with selected GoogleId

Parameters

<i>id</i>	
-----------	--

Returns

Return [Teacher](#) object

3.72.2.6 UpdateFacebookId()

```
int TeacherDaoImpl.UpdateFacebookId (
    string email,
    string id )
```

Update [Teacher](#)'s FacebookId

Parameters

<i>email</i>	
<i>id</i>	

Returns

Return 1 if update query has executed successfully

3.72.2.7 UpdateGoogleId()

```
int TeacherDaoImpl.UpdateGoogleId (
    string email,
    string id )
```

Update [Teacher](#)'s GoogleId

Parameters

<i>email</i>	
<i>id</i>	

Returns

Return 1 if update query has executed successfully

3.73 TeacherLogin Class Reference

Class to handle Presentation for [Login](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.73.1 Detailed Description

Class to handle Presentation for [Login](#)

3.73.2 Member Function Documentation

3.73.2.1 [_Ready\(\)](#)

```
override void TeacherLogin._Ready ( )
```

Initialization

3.74 TeacherMainMenu Class Reference

Class to handle Presentation for [Section](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.74.1 Detailed Description

Class to handle Presentation for [Section](#)

3.74.2 Member Function Documentation

3.74.2.1 _Ready()

```
override void TeacherMainMenu._Ready ( )
```

Initialization

3.75 UserCreatedQuestion Class Reference

DAO Object for [UserCreatedQuestion](#)

Inherits Node.

Public Member Functions

- **UserCreatedQuestion** (int QuestionId, string Option1, string Option2, string Option3, string Option4, int CorrectOption, string QuestionTitle)
- **UserCreatedQuestion** (int QuestionId)

Properties

- int **QuestionId** [get, set]
- string **Option1** [get, set]
- string **Option2** [get, set]
- string **Option3** [get, set]
- string **Option4** [get, set]
- int **CorrectOption** [get, set]
- string **QuestionTitle** [get, set]

3.75.1 Detailed Description

DAO Object for [UserCreatedQuestion](#)

3.76 UserProfile Class Reference

Class to handle Presentation for [UserProfile](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.76.1 Detailed Description

Class to handle Presentation for [UserProfile](#)

3.76.2 Member Function Documentation

3.76.2.1 [_Ready\(\)](#)

```
override void UserProfile._Ready ( )
```

Initialization

3.77 ViewAssignment Class Reference

Class to handle Presentation for [ViewAssignment](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.77.1 Detailed Description

Class to handle Presentation for [ViewAssignment](#)

3.77.2 Member Function Documentation

3.77.2.1 [_Ready\(\)](#)

```
override void ViewAssignment._Ready ( )
```

Initialization

3.78 ViewCreatedLevels Class Reference

Class to handle Presentation for [ViewCreatedLevels](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()

Initialization

3.78.1 Detailed Description

Class to handle Presentation for [ViewCreatedLevels](#)

3.78.2 Member Function Documentation

3.78.2.1 [_Ready\(\)](#)

```
override void ViewCreatedLevels._Ready ( )
```

Initialization

3.79 ViewCustomLevel Class Reference

Class to handle Presentation for [ViewCustomLevel](#)

Inherits Node2D.

Public Member Functions

- override void [_Ready](#) ()

Initialization

3.79.1 Detailed Description

Class to handle Presentation for [ViewCustomLevel](#)

3.79.2 Member Function Documentation

3.79.2.1 `_Ready()`

```
override void ViewCustomLevel._Ready ( )
```

Initialization

3.80 World Class Reference

DAO Object for [World](#)

Inherits Node.

Properties

- int **WorldId** [get, set]
- string **WorldName** [get, set]
- List< [Section](#) > **Section** [get, set]

3.80.1 Detailed Description

DAO Object for [World](#)

3.81 WorldBL Class Reference

Class to handle Business Logic for [World](#)

Inherits Node.

Public Member Functions

- int [GetCompletedWorldCount](#) ()
Get the total number of CompletedWorld for the [Student](#)
- int [GetTotalWorldCount](#) ()
Get the number of Worlds

3.81.1 Detailed Description

Class to handle Business Logic for [World](#)

3.81.2 Member Function Documentation

3.81.2.1 GetCompletedWorldCount()

```
int WorldBL.GetCompletedWorldCount ( )
```

Get the total number of CompletedWorld for the [Student](#)

Returns

Return the total count of Completed Worlds

3.81.2.2 GetTotalWorldCount()

```
int WorldBL.GetTotalWorldCount ( )
```

Get the number of Worlds

Returns

Return integer value of number of Worlds

3.82 WorldDaoImpl Class Reference

Class to handle DAO operations for [World](#)

Public Member Functions

- List< [World](#) > [GetWorlds](#) ()
Get all Worlds
- int [GetCompletedWorldCount](#) ()
Get the total number of CompletedWorld for the [Student](#)

3.82.1 Detailed Description

Class to handle DAO operations for [World](#)

3.82.2 Member Function Documentation

3.82.2.1 GetCompletedWorldCount()

```
int WorldDaoImpl.GetCompletedWorldCount ( )
```

Get the total number of CompletedWorld for the [Student](#)

Returns

Return the total count of Completed Worlds

3.82.2.2 GetWorlds()

```
List<World> WorldDaoImpl.GetWorlds ( )
```

Get all Worlds

Returns

Return list of [World](#) object

3.83 WorldScreen Class Reference

Class to handle Presentation for [WorldScreen](#)

Inherits Node.

Public Member Functions

- override void [_Ready](#) ()
Initialization

3.83.1 Detailed Description

Class to handle Presentation for [WorldScreen](#)

3.83.2 Member Function Documentation

3.83.2.1 _Ready()

```
override void WorldScreen._Ready ( )
```

Initialization

Index

- [_Process](#)
 - [GamePlay, 52](#)
 - [_Ready](#)
 - [AssignmentScreen, 15](#)
 - [Campaign, 18](#)
 - [CharSelect, 24](#)
 - [Chart, 24](#)
 - [CreateLevel, 26](#)
 - [CreateLevelInit, 33](#)
 - [Credits, 33](#)
 - [CustomLevelScreen, 42](#)
 - [DefaultSound, 42](#)
 - [EditLevel, 44](#)
 - [EditLevelInit, 51](#)
 - [GamePlay, 52](#)
 - [LeaderboardScreen, 64](#)
 - [LevelSelection, 65](#)
 - [Login, 66](#)
 - [MainMenu, 66](#)
 - [NotificationPopup, 67](#)
 - [PublishAssignmentScreen, 68](#)
 - [Settings, 77](#)
 - [TeacherAssignment, 107](#)
 - [TeacherLogin, 114](#)
 - [TeacherMainMenu, 115](#)
 - [UserProfile, 116](#)
 - [ViewAssignment, 116](#)
 - [ViewCreatedLevels, 117](#)
 - [ViewCustomLevel, 117](#)
 - [WorldScreen, 120](#)
- [Assignment, 7](#)
- [AssignmentBL, 7](#)
 - [GetAssignment, 8](#)
 - [GetAssignmentMonster, 8](#)
 - [GetStudentAssignment, 8](#)
- [AssignmentDaoImpl, 10](#)
 - [GetAssignment, 10](#)
 - [GetAssignmentMonster, 11](#)
 - [GetStudentAssignment, 11](#)
- [AssignmentScore, 11](#)
- [AssignmentScoreBL, 12](#)
 - [GetStudentCompletedAssignment, 12](#)
 - [InsertAssignmentScore, 13](#)
- [AssignmentScoreDaoImpl, 13](#)
 - [GetStudentCompletedAssignment, 13](#)
 - [InsertAssignmentScore, 14](#)
- [AssignmentScreen, 14](#)
 - [_Ready, 15](#)
- [BackButton, 15](#)
- [BaseDaoImpl< T >, 15](#)
 - [ExecuteQuery, 16](#)
 - [ExecuteScalar, 16, 17](#)
 - [RetrieveQuery, 17](#)
- [BelongClass, 18](#)
- [CalculateScore](#)
 - [Global, 58](#)
- [Campaign, 18](#)
 - [_Ready, 18](#)
- [CampaignBL, 19](#)
 - [GetLevel, 19](#)
- [CampaignDaoImpl, 19](#)
 - [GetLevel, 20](#)
- [CampaignQuestion, 20](#)
- [Character, 20](#)
- [CharacterBL, 21](#)
 - [GetAllCharacters, 21](#)
 - [GetAllMonsters, 21](#)
 - [GetCharacter, 22](#)
- [CharacterDaoImpl, 22](#)
 - [GetAllCharacters, 23](#)
 - [GetAllMonsters, 23](#)
 - [GetCharacter, 23](#)
- [CharSelect, 23](#)
 - [_Ready, 24](#)
- [Chart, 24](#)
 - [_Ready, 24](#)
- [CheckCorrectAnswer](#)
 - [GamePlay, 52](#)
- [CheckDuplicationOptions](#)
 - [CreateLevelBL, 28](#)
 - [EditLevelBL, 45](#)
- [CheckEmail](#)
 - [TeacherBL, 108](#)
 - [TeacherDaoImpl, 111](#)
- [CheckEmptyFieldsExist](#)
 - [CreateLevelBL, 28](#)
 - [EditLevelBL, 45](#)
- [CheckFacebookCharExist](#)
 - [StudentBL, 90](#)
 - [StudentDaoImpl, 96](#)
- [CheckFacebookExist](#)
 - [StudentBL, 91](#)
 - [StudentDaoImpl, 97](#)
- [CheckFacebookLogin](#)
 - [TeacherBL, 108](#)
 - [TeacherDaoImpl, 111](#)
- [CheckGoogleCharExist](#)

- StudentBL, 91
- StudentDaoImpl, 97
- CheckGoogleExist
 - StudentBL, 91
 - StudentDaoImpl, 97
- CheckGoogleLogin
 - TeacherBL, 108
 - TeacherDaoImpl, 112
- CheckSectionCleared
 - SectionBL, 74
 - SectionDaoImpl, 75
- CheckStudentCharExist
 - StudentBL, 92
- CheckStudentExist
 - StudentBL, 92
 - StudentDaoImpl, 98
- CheckValidLevelName
 - CreateLevelBL, 28
 - CreateLevelDAOImpl, 31
 - EditLevelBL, 46
 - EditLevelDaoImpl, 49
- ClassGroup, 25
- ClassGroupDaoImpl, 25
 - GetClassGroup, 25
- CreateLevel, 26
 - _Ready, 26
 - CreateLevelBL, 28
 - SetLevelInitInfo, 26
- CreateLevelBL, 27
 - CheckDuplicationOptions, 28
 - CheckEmptyFieldsExist, 28
 - CheckValidLevelName, 28
 - CreateLevel, 28
 - GetQuestion, 29
 - getTempQuestionList, 29
 - InitializeQuestions, 29
 - ListQuestions, 30
 - ReloadTempQuestionList, 30
 - SaveQuestion, 30
- CreateLevelDAOImpl, 31
 - CheckValidLevelName, 31
 - InsertCustomLevel, 31
 - InsertQuestion, 32
- CreateLevelInit, 32
 - _Ready, 33
- Credits, 33
 - _Ready, 33
- csb
 - Global, 61
- CustomLevel, 33
- CustomLevelBL, 34
 - DeleteCustomLevel, 34
 - GetClearedCustomLevels, 35
 - GetCustomLevel, 35
 - GetCustomLevelMonster, 35
 - GetCustomLevels, 36
 - GetStudentCustomLevel, 36
- CustomLevelDaoImpl, 36
 - DeleteCustomLevel, 37
 - GetClearedCustomLevels, 37
 - GetCustomLevel, 38
 - GetCustomLevelMonster, 38
 - GetCustomLevels, 38
 - GetStudentCustomLevel, 39
- CustomLevelScore, 39
- CustomLevelScoreBL, 40
 - InsertCustomLevelScore, 40
- CustomLevelScoreDaoImpl, 40
 - InsertCustomLevelScore, 41
- CustomLevelScreen, 41
 - _Ready, 42
- DefaultSound, 42
 - _Ready, 42
 - disableSound, 43
 - enableSound, 43
 - playSound, 43
- DeleteCustomLevel
 - CustomLevelBL, 34
 - CustomLevelDaoImpl, 37
- disableSound
 - DefaultSound, 43
- DisplayCharSprite
 - GamePlay, 53
- DisplayMonsterSprite
 - GamePlay, 53
- DisplayNextQuestion
 - GamePlay, 53
- DisplayPopup
 - NotificationPopup, 68
- DisplayQuestion
 - GamePlay, 53
- DisplayStars
 - GamePlay, 53
- EditLevel, 43
 - _Ready, 44
 - SetLevelInitInfo, 44
- EditLevelBL, 44
 - CheckDuplicationOptions, 45
 - CheckEmptyFieldsExist, 45
 - CheckValidLevelName, 46
 - GetOriginalQuestionList, 46
 - GetQuestion, 46
 - getTempQuestionList, 47
 - LoadCustomLevelInfo, 47
 - ReloadTempQuestionList, 47
 - SaveQuestion, 47
 - UpdateLevel, 48
 - UpdateLevelInitInfo, 48
- EditLevelDaoImpl, 48
 - CheckValidLevelName, 49
 - GetLevelInfo, 49
 - UpdateLevelInitInfo, 49
 - UpdateQuestion, 50
- EditLevelInit, 50
 - _Ready, 51

- enableSound
 - DefaultSound, [43](#)
- ExecuteQuery
 - BaseDaoImpl< T >, [16](#)
- ExecuteScalar
 - BaseDaoImpl< T >, [16](#), [17](#)
- GamePlay, [51](#)
 - _Process, [52](#)
 - _Ready, [52](#)
 - CheckCorrectAnswer, [52](#)
 - DisplayCharSprite, [53](#)
 - DisplayMonsterSprite, [53](#)
 - DisplayNextQuestion, [53](#)
 - DisplayQuestion, [53](#)
 - DisplayStars, [53](#)
 - GetTimeLeft, [54](#)
 - LoadStart, [54](#)
 - SetBg, [54](#)
 - SetCharacter, [54](#)
 - SetGameType, [55](#)
 - SetLevelTitle, [55](#)
 - SetQuestionList, [55](#)
 - SetQuestionNum, [56](#)
 - SetTimeLimit, [56](#)
- GetAllCharacters
 - CharacterBL, [21](#)
 - CharacterDaoImpl, [23](#)
- GetAllMonsters
 - CharacterBL, [21](#)
 - CharacterDaoImpl, [23](#)
- GetAssignment
 - AssignmentBL, [8](#)
 - AssignmentDaoImpl, [10](#)
- GetAssignmentMonster
 - AssignmentBL, [8](#)
 - AssignmentDaoImpl, [11](#)
- GetAssignmentRanking
 - StatisticsBL, [79](#)
- GetAssignments
 - StatisticsBL, [79](#)
- GetAvgAssignmentScore
 - StatisticsBL, [79](#), [80](#)
- GetAvgSectionScores
 - StatisticsBL, [80](#)
- GetAvgSectionScoresAvg
 - StatisticsBL, [80](#)
- GetAvgSectionScoresMax
 - StatisticsBL, [81](#)
- GetAvgSectionScoresMin
 - StatisticsBL, [81](#)
- GetAvgStudentAssignmentScores
 - StatisticsBL, [82](#)
- GetAvgStudentScore
 - StatisticsBL, [82](#)
- GetAvgWorldScores
 - StatisticsBL, [82](#)
 - StudentScoreBL, [101](#)
 - StudentScoreDaoImpl, [103](#)
- GetAvgWorldScoresAvg
 - StatisticsBL, [83](#)
- GetAvgWorldScoresMax
 - StatisticsBL, [83](#)
- GetAvgWorldScoresMin
 - StatisticsBL, [83](#)
- GetCampaignRanking
 - StatisticsBL, [83](#)
 - StudentScoreBL, [102](#)
 - StudentScoreDaoImpl, [105](#)
- GetCharacter
 - CharacterBL, [22](#)
 - CharacterDaoImpl, [23](#)
- GetClassGroup
 - ClassGroupDaoImpl, [25](#)
- GetClassNotAssignment
 - PublishedAssignmentBL, [69](#)
 - PublishedAssignmentDaoImpl, [71](#)
- GetClearedCustomLevels
 - CustomLevelBL, [35](#)
 - CustomLevelDaoImpl, [37](#)
- GetCompletedWorldCount
 - WorldBL, [118](#)
 - WorldDaoImpl, [119](#)
- GetCustomLevel
 - CustomLevelBL, [35](#)
 - CustomLevelDaoImpl, [38](#)
- GetCustomLevelMonster
 - CustomLevelBL, [35](#)
 - CustomLevelDaoImpl, [38](#)
- GetCustomLevels
 - CustomLevelBL, [36](#)
 - CustomLevelDaoImpl, [38](#)
- GetFacebookStudent
 - StudentDaoImpl, [98](#)
- GetFacebookStudentId
 - StudentBL, [92](#)
- GetFacebookTeacher
 - TeacherDaoImpl, [112](#)
- GetFacebookTeacherId
 - TeacherBL, [109](#)
- GetFirstLoggedIn
 - Global, [58](#)
- GetGoogleStudent
 - StudentDaoImpl, [98](#)
- GetGoogleStudentId
 - StudentBL, [93](#)
- GetGoogleTeacher
 - TeacherDaoImpl, [113](#)
- GetGoogleTeacherId
 - TeacherBL, [109](#)
- GetLeaderboards
 - LeaderboardBL, [62](#)
- GetLeaderboardScore
 - LeaderboardDaoImpl, [63](#)
- GetLevel
 - CampaignBL, [19](#)
 - CampaignDaoImpl, [20](#)

- GetLevelInfo
 - EditLevelDaoImpl, 49
- GetMaxAssignmentScore
 - StatisticsBL, 84
- GetMaxStudentAssignmentScores
 - StatisticsBL, 84
- GetMaxStudentScore
 - StatisticsBL, 84
- GetMinAssignmentScore
 - StatisticsBL, 85
- GetMinStudentAssignmentScores
 - StatisticsBL, 85
- GetMinStudentScore
 - StatisticsBL, 86
- GetOriginalQuestionList
 - EditLevelBL, 46
- GetQuestion
 - CreateLevelBL, 29
 - EditLevelBL, 46
- GetSectionLevels
 - SectionBL, 74
 - SectionDaoImpl, 76
- GetSections
 - StatisticsBL, 86
- GetStudentAssignment
 - AssignmentBL, 8
 - AssignmentDaoImpl, 11
- GetStudentAssignmentScores
 - StatisticsBL, 87
- GetStudentCharacter
 - StudentBL, 93
 - StudentDaoImpl, 99
- GetStudentClass
 - StatisticsBL, 87
- GetStudentCompletedAssignment
 - AssignmentScoreBL, 12
 - AssignmentScoreDaoImpl, 13
- GetStudentCustomLevel
 - CustomLevelBL, 36
 - CustomLevelDaoImpl, 39
- GetStudentName
 - Global, 58
- GetStudents
 - StatisticsBL, 87
- GetStudentScores
 - StatisticsBL, 87, 88
 - StudentScoreBL, 102
 - StudentScoreDaoImpl, 105
- GetTeacherPublished
 - PublishedAssignmentBL, 70
 - PublishedAssignmentDaoImpl, 71
- GetTempQuestionList
 - EditLevelBL, 47
- getTempQuestionList
 - CreateLevelBL, 29
- GetTimeLeft
 - GamePlay, 54
- GetTotalWorldCount
 - WorldBL, 119
- GetWorldLeaderboard
 - LeaderboardBL, 62
- GetWorlds
 - LeaderboardBL, 63
 - StatisticsBL, 88
 - WorldDaoImpl, 120
- GetWorldSections
 - SectionBL, 75
 - SectionDaoImpl, 76
 - StatisticsBL, 88
- Global, 56
 - CalculateScore, 58
 - csb, 61
 - GetFirstLoggedIn, 58
 - GetStudentName, 58
 - LoadSprite, 58
 - SetFbLoggedIn, 59
 - SetFirstLoggedIn, 59
 - SetGoogleLoggedIn, 59
 - SetStudentId, 59
 - SetStudentName, 61
 - StudentName, 61
- InitializeQuestions
 - CreateLevelBL, 29
- InsertAssignmentScore
 - AssignmentScoreBL, 13
 - AssignmentScoreDaoImpl, 14
- InsertCustomLevel
 - CreateLevelDAOImpl, 31
- InsertCustomLevelScore
 - CustomLevelScoreBL, 40
 - CustomLevelScoreDaoImpl, 41
- InsertFacebookStudent
 - StudentBL, 94
 - StudentDaoImpl, 99
- InsertGoogleStudent
 - StudentBL, 94
 - StudentDaoImpl, 100
- InsertPublishedAssignment
 - PublishedAssignmentBL, 70
 - PublishedAssignmentDaoImpl, 72
- InsertQuestion
 - CreateLevelDAOImpl, 32
- InsertStudentScore
 - StudentScoreBL, 102
 - StudentScoreDaoImpl, 105
- Leaderboard, 62
 - LeaderboardBL, 62
 - GetLeaderboards, 62
 - GetWorldLeaderboard, 62
 - GetWorlds, 63
- LeaderboardDaoImpl, 63
 - GetLeaderboardScore, 63
- LeaderboardScreen, 64
 - _Ready, 64
- Level, 64

- LevelSelection, [65](#)
 - _Ready, [65](#)
- ListQuestions
 - CreateLevelBL, [30](#)
- LoadCustomLevelInfo
 - EditLevelBL, [47](#)
- LoadSprite
 - Global, [58](#)
- LoadStart
 - GamePlay, [54](#)
- Login, [65](#)
 - _Ready, [66](#)
- MainMenu, [66](#)
 - _Ready, [66](#)
- Monster, [67](#)
- NotificationPopup, [67](#)
 - _Ready, [67](#)
 - DisplayPopup, [68](#)
- playSound
 - DefaultSound, [43](#)
- PublishAssignmentScreen, [68](#)
 - _Ready, [68](#)
- PublishedAssignment, [69](#)
- PublishedAssignmentBL, [69](#)
 - GetClassNotAssignment, [69](#)
 - GetTeacherPublished, [70](#)
 - InsertPublishedAssignment, [70](#)
- PublishedAssignmentDaoImpl, [71](#)
 - GetClassNotAssignment, [71](#)
 - GetTeacherPublished, [71](#)
 - InsertPublishedAssignment, [72](#)
- Question, [72](#)
- ReloadTempQuestionList
 - CreateLevelBL, [30](#)
 - EditLevelBL, [47](#)
- RetrieveQuery
 - BaseDaoImpl< T >, [17](#)
- SaveQuestion
 - CreateLevelBL, [30](#)
 - EditLevelBL, [47](#)
- Section, [73](#)
- SectionBL, [73](#)
 - CheckSectionCleared, [74](#)
 - GetSectionLevels, [74](#)
 - GetWorldSections, [75](#)
- SectionDaoImpl, [75](#)
 - CheckSectionCleared, [75](#)
 - GetSectionLevels, [76](#)
 - GetWorldSections, [76](#)
- SetBg
 - GamePlay, [54](#)
- SetCharacter
 - GamePlay, [54](#)
- SetFbLoggedIn
 - Global, [59](#)
- SetFirstLoggedIn
 - Global, [59](#)
- SetGameType
 - GamePlay, [55](#)
- SetGoogleLoggedIn
 - Global, [59](#)
- SetLevelInitInfo
 - CreateLevel, [26](#)
 - EditLevel, [44](#)
- SetLevelTitle
 - GamePlay, [55](#)
- SetQuestionList
 - GamePlay, [55](#)
- SetQuestionNum
 - GamePlay, [56](#)
- SetStudentId
 - Global, [59](#)
- SetStudentName
 - Global, [61](#)
- SetTimeLimit
 - GamePlay, [56](#)
- Settings, [77](#)
 - _Ready, [77](#)
- StatisticsBL, [77](#)
 - GetAssignmentRanking, [79](#)
 - GetAssignments, [79](#)
 - GetAvgAssignmentScore, [79, 80](#)
 - GetAvgSectionScores, [80](#)
 - GetAvgSectionScoresAvg, [80](#)
 - GetAvgSectionScoresMax, [81](#)
 - GetAvgSectionScoresMin, [81](#)
 - GetAvgStudentAssignmentScores, [82](#)
 - GetAvgStudentScore, [82](#)
 - GetAvgWorldScores, [82](#)
 - GetAvgWorldScoresAvg, [83](#)
 - GetAvgWorldScoresMax, [83](#)
 - GetAvgWorldScoresMin, [83](#)
 - GetCampaignRanking, [83](#)
 - GetMaxAssignmentScore, [84](#)
 - GetMaxStudentAssignmentScores, [84](#)
 - GetMaxStudentScore, [84](#)
 - GetMinAssignmentScore, [85](#)
 - GetMinStudentAssignmentScores, [85](#)
 - GetMinStudentScore, [86](#)
 - GetSections, [86](#)
 - GetStudentAssignmentScores, [87](#)
 - GetStudentClass, [87](#)
 - GetStudents, [87](#)
 - GetStudentScores, [87, 88](#)
 - GetWorlds, [88](#)
 - GetWorldSections, [88](#)
- Student, [89](#)
- StudentBL, [89](#)
 - CheckFacebookCharExist, [90](#)
 - CheckFacebookExist, [91](#)
 - CheckGoogleCharExist, [91](#)
 - CheckGoogleExist, [91](#)

- CheckStudentCharExist, 92
- CheckStudentExist, 92
- GetFacebookStudentId, 92
- GetGoogleStudentId, 93
- GetStudentCharacter, 93
- InsertFacebookStudent, 94
- InsertGoogleStudent, 94
- UpdateStudentCharacter, 94
- StudentCustomQuestion, 95
- StudentDaoImpl, 95
 - CheckFacebookCharExist, 96
 - CheckFacebookExist, 97
 - CheckGoogleCharExist, 97
 - CheckGoogleExist, 97
 - CheckStudentExist, 98
 - GetFacebookStudent, 98
 - GetGoogleStudent, 98
 - GetStudentCharacter, 99
 - InsertFacebookStudent, 99
 - InsertGoogleStudent, 100
 - UpdateStudentCharacter, 100
- StudentName
 - Global, 61
- StudentScore, 100
- StudentScoreBL, 101
 - GetAvgWorldScores, 101
 - GetCampaignRanking, 102
 - GetStudentScores, 102
 - InsertStudentScore, 102
- StudentScoreDaoImpl, 103
 - GetAvgWorldScores, 103
 - GetCampaignRanking, 105
 - GetStudentScores, 105
 - InsertStudentScore, 105
- Teacher, 106
- TeacherAssignment, 106
 - _Ready, 107
- TeacherBL, 107
 - CheckEmail, 108
 - CheckFacebookLogin, 108
 - CheckGoogleLogin, 108
 - GetFacebookTeacherId, 109
 - GetGoogleTeacherId, 109
 - UpdateFacebookId, 109
 - UpdateGooglId, 110
- TeacherCustomQuestion, 110
- TeacherDaoImpl, 111
 - CheckEmail, 111
 - CheckFacebookLogin, 111
 - CheckGoogleLogin, 112
 - GetFacebookTeacher, 112
 - GetGoogleTeacher, 113
 - UpdateFacebookId, 113
 - UpdateGooglId, 113
- TeacherLogin, 114
 - _Ready, 114
- TeacherMainMenu, 114
 - _Ready, 115
- UpdateFacebookId
 - TeacherBL, 109
 - TeacherDaoImpl, 113
- UpdateGooglId
 - TeacherBL, 110
 - TeacherDaoImpl, 113
- UpdateLevel
 - EditLevelBL, 48
- UpdateLevelInitInfo
 - EditLevelBL, 48
 - EditLevelDaoImpl, 49
- UpdateQuestion
 - EditLevelDaoImpl, 50
- UpdateStudentCharacter
 - StudentBL, 94
 - StudentDaoImpl, 100
- UserCreatedQuestion, 115
- UserProfile, 115
 - _Ready, 116
- ViewAssignment, 116
 - _Ready, 116
- ViewCreatedLevels, 117
 - _Ready, 117
- ViewCustomLevel, 117
 - _Ready, 117
- World, 118
- WorldBL, 118
 - GetCompletedWorldCount, 118
 - GetTotalWorldCount, 119
- WorldDaoImpl, 119
 - GetCompletedWorldCount, 119
 - GetWorlds, 120
- WorldScreen, 120
 - _Ready, 120