TwoA TypeScript API Manual

Contents

Summary

Document information	
Document version	1.2.5
Document date	2018.07.30
Document author	Enkhbold Nyamsuren (E.Nyamsuren@gmail.com, Enkhbold.Nyamsuren@ou.nl)

Asset author information	
Name	Enkhbold Nyamsuren
E-mail	E.Nyamsuren@gmail.com Enkhbold.Nyamsuren@ou.nl
Organization	OUNL

Asset information	
Current version	1.2.5
Date	2018.06.18
Deployment side	client-side
Programming language	TypeScript
Required libraries	Microsoft .NET 3.5 Framework or higher
Recommended platform	Windows OS

Adaptation Modules

The asset provides two different modules for adaptation and assessment. Modules differ in terms of required input to the asset from a game. In this context, input refers to player's performance measures such as response time and accuracy. Adaptation module ID should be passed to the TwoA asset in order to indicated the module that should be used.

Adaptation Module 1	
Adatation ID (string type)	"Game difficulty - Player skill"
Description	Assess and adapts game difficulty to player skill. Skill ratings are evaluated for individual players. Requires player accuracy and response time. Uses a modified version of the CAP algorithm.
Input 1	Player's accuracy. The value should have *double* type. The value should be either 0 or 1. The value is 1 if the player successfully completed a game scenario. The value is 0 if the player failed the game scenario.
Input 2	Player's response time. The value should have *double* type. The duration of time the player required to complete (either successfully or unsuccessfully) a game scenario. Time is measured in milliseconds.

Adaptation Module 1	
Adatation ID (string type)	"SkillDifficultyElo"
Description	Assess and adapts game difficulty to player skill. Skill ratings are evaluated for individual players. Requires player accuracy and response time. Uses a modified version of the CAP algorithm.
Input 1	Player's accuracy. The value should have *double* type. The value should be between 0 and 1. The value of 0 represents the worst possible player performance in a game scenario. The value of 1 represents the best possible player performance in a game scenario.

Rating Scale

Given a player with a skill rating θ, the table below shows the player's expected success rate (column "*P*") in a scenario with a specific difficulty rating (column "Difficulty rating"). For example, if the player's skill rating is equal to one ($\theta = 1$) then the player has 74% chance of successfully completing a scenario with a difficulty rating -0.046 (1 - 1.046).

P	*P* in %	Difficulty rating
0.02	2%	$\theta + 3.8918$
0.04	4%	$\theta + 3.1781$
0.06	6%	$\theta + 2.7515$
0.08	8%	$\theta + 2.4423$
0.1	10%	$\theta + 2.1972$
0.12	12%	$\theta + 1.9924$
0.14	14%	$\theta + 1.8153$
0.16	16%	$\theta + 1.6582$
0.18	18%	$\theta + 1.5163$
0.2	20%	$\theta + 1.3863$
0.22	22%	$\theta + 1.2657$
0.24	24%	$\theta + 1.1527$
0.26	26%	$\theta + 1.046$
0.28	28%	$\theta + 0.9445$
0.3	30%	$\theta + 0.8473$
0.32	32%	$\theta + 0.7538$
0.34	34%	$\theta + 0.6633$
0.36	36%	$\theta + 0.5754$
0.38	38%	$\theta + 0.4895$
0.4	40%	$\theta + 0.4055$
0.42	42%	$\theta + 0.3228$
0.44	44%	$\theta + 0.2412$
0.46	46%	$\theta + 0.1603$

0.48	48%	θ + 0.08
0.5	50%	θ + 0
0.52	52%	θ - 0.08
0.54	54%	θ - 0.1603
0.56	56%	θ - 0.2412
0.58	58%	θ - 0.3228
0.6	60%	θ - 0.4055
0.62	62%	θ - 0.4895
0.64	64%	θ - 0.5754
0.66	66%	θ - 0.6633
0.68	68%	θ - 0.7538
0.7	70%	θ - 0.8473
0.72	72%	θ - 0.9445
0.74	74%	θ - 1.046
0.76	76%	θ - 1.1527
0.78	78%	θ - 1.2657
0.8	80%	θ - 1.3863
0.82	82%	θ - 1.5163
0.84	84%	θ - 1.6582
0.86	86%	θ - 1.8153
0.88	88%	θ - 1.9924
0.9	90%	θ - 2.1972
0.92	92%	θ - 2.4423
0.94	94%	θ - 2.7515
0.96	96%	θ - 3.1781
0.98	98%	θ - 3.8918

TwoA class

Class	
Class name	TwoA
Namespace	TwoANS
Description	The main class of the asset. An instance of this class should be created to access asset's API.

Constructor	
Name	constructor()
Description	Initializes a new instance of the TwoA class.

Field name	Field description
players: PlayerNode[]	A list of PlayerNode instances. An empty array is automatically initialized during constructor call. Each PlayerNode instance contains data of a single player. Refer to PlayerNode section for more information.
scenarios: ScenarioNode[]	A list of ScenarioNode instances. An empty array is automatically initialized during constructor call. Each ScenarioNode instance contains data of a single game scenario. Refer to ScenarioNode section for more information.
gameplays: Gameplay[]	A list of Gameplay instances. An empty array is automatically initialized during constructor call. Each Gameplay instance contains a single assessment record created at the end of UpdateRatings method. Refer to Gameplay section for more information.

Methods for adaptation.

Method	
Name	AvailableAdapters(): string[][]
Description	Returns Nx3 array of string. Each row contains information about an available adaptation module. The first column contains the class name. The second column stores adaptation module ID. The third column stores a description for the adaptation module.
Return type	**Return description**

string[][]	2D array with a N number of rows and 3 columns.

Method	
Name	TargetScenarioID(p_playerNode: PlayerNode): string
Description	Returns ID of a game scenario with a difficulty rating that matches the skill rating of a specified player. The recommended scenario is selected from TwoA.scenarios list.
Parameter name	**Parameter description**
p_playerNode: PlayerNode	PlayerNode of a player to whom the scenario difficulty should be matched.
Return type	**Return description**
string	ID of a game scenario of recommended difficulty. Null if any error occurred.

Method	
Name	TargetScenario(p_playerNode: PlayerNode): ScenarioNode
Description	Returns an instance of ScenarioNode of a game scenario with a difficulty rating that matches the skill rating of a specified player. The recommended scenario is selected from TwoA.scenarios list.
Parameter name	**Parameter description**
p_playerNode: PlayerNode	PlayerNode of a player to whom the scenario difficulty should be matched.
Return type	**Return description**
ScenarioNode	An instance of ScenarioNode of a game scenario of recommended difficulty. Null if any error occurred.

Method	
Name	TargetScenarioCustom(p_playerNode: PlayerNode, p_scenarioList: ScenarioNode[]): ScenarioNode
Description	Returns an instance of ScenarioNode of a game scenario with a difficulty rating that matches the skill rating of a specified player. Requires a custom list of scenarios from which a recommendation should be made.
Parameter name	**Parameter description**
p_playerNode: PlayerNode	PlayerNode of a player to whom the scenario difficulty should be matched.
p_scenarioList:	A list of ScenarioNode instances from which a scenario will be selected and

ScenarioNode[]	matched to player's skill level.
Return type	**Return description**
ScenarioNode	An instance of ScenarioNode of a game scenario of recommended difficulty. Null if any error occurred.

Method	
Name	TargetDifficultyRatingCustom(p_adaptID: string, p_playerRating: number): number
Description	Returns a recommended difficulty rating for a specified player skill rating.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module to be used (refer to "Adaptation modules").
p_playerRating: number	Player's skill rating.
Return type	**Return description**
number	Difficulty rating.

Method	
Name	TargetDifficultyRating(p_playerNode: PlayerNode): number
Description	Returns a recommended difficulty rating for a specified player.
Parameter name	**Parameter description**
p_playerNode: PlayerNode	PlayerNode of a player to whom the scenario difficulty should be matched.
Return type	**Return description**
number	Difficulty rating.

Methods for assessment.

Method	
Name	UpdateRatings(p_playerNode: PlayerNode, p_scenarioNode: ScenarioNode, p_rt: number, p_correctAnswer: number, p_updateScenarioRating: boolean, p_customKfct: number): boolean
Description	Reassesses and updates player skill rating and, optionally, scenario difficulty rating based on player's performance in a specified scenario.
Parameter name	**Parameter description**

p_playerNode: PlayerNode	PlayerNode of a player to assess.
p_scenarioNode: ScenarioNode	ScenarioNode a scenario in which player's performance was measure.
p_rt: number	Player's response time measured in milliseconds (see "Adaptation Modules"). If adaptation is based on accuracy only then this parameter will be automatically ignored. Should be higher than 0.
p_correctAnswer: number	Player's accuracy (see "Adaptation Modules"). Depending on the adaptation module should be either binary (0 or 1) or a value between 0 and 1. Assessment is not performed if accuracy value does not match module's requirements.
p_updateScenarioRating: boolean	If true scenario's difficulty rating will be reassessed and updated.
p_customKfct: number	A custom K factor to control the scale of changes in player's and scenario's ratings. Requires a positive value. A higher value results in a bigger change in the rating. If value is 0 then TwoA uses a dynamically estimated K factor. Use this parameter with care since it can drastically influence TwoA's performance of adaptation and assessment. Consult the table in section "Rating Scale" to decide on the K factor appropriate for you.
Return type	**Return description**
boolean	True if ratings were reassessed and updated successfully, and false otherwise.

Method	
Name	CreateNewRecord(p_adaptID: string, p_gameID: string, p_playerID: string, p_scenarioID: string, p_rt: number, p_accuracy: number, p_playerRating: number, p_scenarioRating: number, p_timestamp: string): void
Description	Records results of player assessment by creating a new instance of Gameplay class. The instance is stored in TwoA.gameplays list. The player and scenario data is retrieved from the TwoA.players and TwoA.scenarios lists respectively. This method is automatically called by UpdateRatings methods.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module to be used (see "Adaptation Modules").
p_gameID: string	ID of a game from which a scenario was selected.
p_playerID: string	ID of a player that was assessed.
p_scenarioID: string	ID of a scenario in which player's performance was measured.
p_rt: number	Player's response time in milliseconds (see "Adaptation Modules").

p_accuracy: number	Player's accuracy (see "Adaptation Modules"). Depending on the adaptation module should be either binary (0 or 1) or a value between 0 and 1.
p_playerRating: number	Player's skill rating after reassessment.
p_scenarioRating: number	Scenario's difficulty rating after reassessment.
p_timestamp: string	Date and time of reassessment.

Methods for scoring.

Method	
Name	CalculateScore(p_correctAnswer: number, p_responseTime: number, p_itemMaxDuration: number): number
Description	Transforms player's accuracy and response time into a single score measured in the range (-1, 1).
Parameter name	**Parameter description**
p_correctAnswer: number	Player's accuracy that is either 0 or 1. 1 is for success, and 0 is for fails.
p_responseTime: number	Player's response time in milliseconds.
p_itemMaxDuration: number	Max amount of time a player is allowed to spend to complete a game scenario. Measured in milliseconds.
Return type	**Return description**
number	A score between -1 and 1.

Method	
Name	CalculateExpectedScore(p_adaptID: string, p_playerRating: number, p_scenarioRating: number, p_itemMaxDuration: number): number
Description	Calculates player's expected score based on player's skill rating and scenarios difficulty rating.
Parameter name	**Parameter description**
p_adaptID: string	ID of the adaptation module to use.
p_playerRating: number	Player's skill rating.
p_scenarioRating: number	Scenario's difficulty rating.
p_itemMaxDuration: number	Max allowed time in millisecond given to player to solve the problem.

Return type	Return description
number	A score between -1 and 1.

Methods for controlling success rate parameter.

When recommending scenarios, TwoA tries to ensure that a player can maintain an average success rate *P* of successfully completing the scenarios. For example, if *P* = 0.75 then the player is ideally expected to successfully complete 75% of all scenarios recommended by the TwoA asset. In a more realistic case, player's actual success rate will not be exactly 75% but close to it (little bit more or little bit less). More specifically, player's real success rate will follow a normal distribution with mean at 0.75.
The game developer can change the success rate parameters to suit needs of specific games or player audience. Four parameters are needed to set the success rate. The first two are the mean and standard deviation defining the normal distribution. There are also two hard boundaries within which player's real success rate is expected to lie.
The default success rate is defined by a normal distribution N(M=0.75, SD=0.1) with mean at 0.75 and a standard deviation of 0.1. Hard boundaries are 0.5 and 0.99. This means there is 95% chance that player actual success rate will be between 55% (M – 2*SD) and 95% (M + 2*SD), but it will never drop below 50% and never go above 99%.
Similarly, a game developer can set success rate to follow a normal distribution N(M=0.5, SD=0.1) with hard boundaries at 0.25 and 0.75. This means there is 95% chance that player actual success rate will be between 30% (M – 2*SD) and 70% (M + 2*SD), but it will never drop below 25% and never go above 75%.

Method	
Name	GetTargetDistribution(p_adaptID: string): number[]
Description	Returns the four parameters defining the target success rate that is used to select a scenario of a recommended difficulty.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see "Adaptation Modules").
Return type	**Return description**
number[]	An array of four double values: distribution mean, distribution standard deviation, lower hard boundary, and upper hard boundary.

Method	
Name	SetTargetDistribution(p_adaptID: string, p_mean: number, p_sd: number, p_lowerLimit: number, p_upperLimit: number): void
Description	Sets the parameters for the target success rate that is used to select a scenario of a recommended difficulty.
Parameter name	**Parameter description**

p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_mean: number	Mean of a normal distribution. Any value between 0 and 1 (exclusive).
p_sd: number	Standard deviation of a normal distribution. Any value between 0 and 1 (exclusive).
p_lowerLimit: number	Lower hard boundary. Any value between 0 and 1 (inclusive). Should be less than standard distribution mean.
p_upperLimit: number	Upper hard boundary. Any value between 0 and 1 (inclusive). Should be higher than standard distribution mean.

Method	
Name	SetDefaultTargetDistribution(p_adaptID: string): void
Description	Sets the parameters for the target success rate to its default values that is used to select a scenario of a recommended difficulty. The default values are 0.75 for distribution mean, 0.1 for standard deviation, 0.5 for lower boundary, and 1 for upper boundary.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Methods for controlling the fuzzy selection intervals.

Method	
Name	GetFiSDMultiplier(p_adaptID: string): number
Description	Returns the multiplier for the standard deviation used for calculating the support fuzzy interval.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
Return type	**Return description**
number	Multiplier value, or 0 if the adapter is not found.

Method	
Name	SetFiSDMultiplier(p_adaptID: string, p_multiplier: number): void
Description	Sets a value for the multiplier for the standard deviation used for calculating the support fuzzy interval.
Parameter name	**Parameter description**

p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_multiplier: number	The value of the multiplier.

Method	
Name	SetDefaultFiSDMultiplier(p_adaptID: string): void
Description	Sets to its default value the multiplier for the standard deviation used for calculating the support fuzzy interval.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Methods for the uncertainty parameter.

Method	
Name	GetMaxDelay(p_adaptID: string): number
Description	Gets value for the max number of days after which player's or item's uncertainty reaches the maximum of 1. This a continuous number of days during which the player has not played any scenario, or the scenario was not played by any player.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
Return type	**Return description**
number	The number of days as double value.

Method	
Name	SetMaxDelay(p_adaptID: string, p_maxDelay: number): void
Description	Sets a value for the max number of days after which player's or item's uncertainty reaches the maximum of 1. This a continuous number of days during which the player has not played any scenario, or the scenario was not played by any player.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_maxDelay: number	The value in the number of days.

Method	
Name	SetDefaultMaxDelay(p_adaptID: string): void
Description	Sets to its default value the max number of days after which player's or item's uncertainty reaches the maximum of 1. This a continuous number of days during which the player has not played any scenario, or the scenario was not played by any player.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Method	
Name	GetMaxPlay(p_adaptID: string): number
Description	Gets value for the max number of gameplays after which player’s or scenario’s rating uncertainty reaches 0.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
Return type	**Return description**
number	The number of gameplays as double value.

Method	
Name	SetMaxPlay(p_adaptID: string, p_maxPlay: number): void
Description	Sets a value for the max number of gameplays after which player’s or scenario’s rating uncertainty reaches 0.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_maxPlay: number	The value in the number of gameplays.

Method	
Name	SetDefaultMaxPlay(p_adaptID: string): void
Description	Sets to its default value the max number of gameplays after which player’s or scenario’s rating uncertainty reaches 0.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Methods for K factor.

Method	
Name	GetKConst(p_adaptID: string): number
Description	Gets the min value for the K factor that is used when there are no uncertainties in player's and scenario's ratings.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see "Adaptation Modules").
Return type	**Return description**
number	K factor value as double.

Method	
Name	SetKConst(p_adaptID: string, p_kConst: number): void
Description	Sets the min value for the K factor that is used when there are no uncertainties in player's and scenario's ratings.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see "Adaptation Modules").
p_kConst: number	The min value for the K factor.

Method	
Name	SetDefaultKConst(p_adaptID: string): void
Description	Sets the min value for the K factor to its default value.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see "Adaptation Modules").

Method	
Name	GetKUp(p_adaptID: string): number
Description	Gets the step value by which the K factor should increase.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see "Adaptation Modules").
Return type	**Return description**

number	K factor step value.

Method	
Name	SetKUp(p_adaptID: string, p_kUp: number): void
Description	Sets the step value by which the K factor should increase.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_kUp: number	The step value.

Method	
Name	SetDefaultKUp(p_adaptID: string): void
Description	Sets to its default value the step size by which the K factor should increase.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Method	
Name	GetKDown(p_adaptID: string): number
Description	Gets the step value by which the K factor should decrease.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
Return type	**Return description**
number	K factor step value.

Method	
Name	SetKDown(p_adaptID: string, p_kDown: number): void
Description	Sets the step value by which the K factor should decrease.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_kDown: number	The step value.

Method	
Name	SetDefaultKDown(p_adaptID: string): void
Description	Sets to its default value the step size by which the K factor should decrease.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Methods for the calibration parameters.

Method	
Name	GetPlayerCalLength(p_adaptID: string): number
Description	Returns the length of a player’s calibration phase. The length is measured in number of gameplays.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
Return type	**Return description**
number	Number of gameplays.

Method	
Name	SetPlayerCalLength(p_adaptID: string, p_calLength: number): void
Description	Sets the length of a player’s calibration phase. The length is measured in number of gameplays.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_calLength: number	The length in the number of gameplays.

Method	
Name	SetDefaultPlayerCalLength(p_adaptID: string): void
Description	Sets to its default value the length of a player’s calibration phase. The length is measured in number of gameplays.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Method	
Name	GetScenarioCalLength(p_adaptID: string): number
Description	Returns the length of a scenario's calibration phase. The length is measured in number of gameplays.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see "Adaptation Modules").
Return type	**Return description**
number	Number of gameplays.

Method	
Name	SetScenarioCalLength(p_adaptID: string, p_calLength: number): void
Description	Sets the length of a scenario's calibration phase. The length is measured in number of gameplays.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see "Adaptation Modules").
p_calLength: number	The length in the number of gameplays.

Method	
Name	SetDefaultScenarioCalLength(p_adaptID: string): void
Description	Sets to its default value the length of a scenario's calibration phase. The length is measured in number of gameplays.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see "Adaptation Modules").

Method	
Name	SetCalLength(p_adaptID: string, p_calLength: number): void
Description	Sets the scenario and player calibration lengths to the same value. The length is measured in number of gameplays.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see "Adaptation Modules").
p_calLength: number	The length in the number of gameplays.

Method	
Name	SetDefaultCalLength(p_adaptID: string): void
Description	Sets scenario and player calibration lengths to its default values. The length is measured in number of gameplays.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Method	
Name	GetPlayerCalK(p_adaptID: string): number
Description	Returns the custom K factor used during player’s calibration.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
Return type	**Return description**
number	K factor value as double.

Method	
Name	SetPlayerCalK(p_adaptID: string, p_calK: number): void
Description	Sets the custom K factor used during player’s calibration.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_calK: number	K factor value as double.

Method	
Name	SetDefaultPlayerCalK(p_adaptID: string): void
Description	Sets to its default value the custom K factor used during player’s calibration.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Method	
Name	GetScenarioCalK(p_adaptID: string): number
Description	Returns the custom K factor used during scenario’s calibration.

Parameter name	Parameter description
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
Return type	**Return description**
number	K factor value as double.

Method	
Name	SetScenarioCalK(p_adaptID: string, p_calK: number): void
Description	Sets the custom K factor used during scenario’s calibration.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_calK: number	K factor value as double.

Method	
Name	SetDefaultScenarioCalK(p_adaptID: string): void
Description	Sets to its default value the custom K factor used during scenario’s calibration.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Method	
Name	SetCalK(p_adaptID: string, p_calK: number): void
Description	Sets the custom K factor used during both player’s and scenario’s calibrations.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_calK: number	K factor value as double.

Method	
Name	SetDefaultCalK(p_adaptID: string): void
Description	Sets to its default values the custom K factors used during player’s and scenario’s calibrations.

Parameter name	Parameter description
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Methods controlling ELO parameters.

Method	
Name	GetExpectScoreMagnifier(p_adaptID: string): number
Description	Returns the value of the magnifier for the expected score compared to the opponent’s score.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
Return type	**Return description**
number	Magnifier value.

Method	
Name	SetExpectScoreMagnifier(p_adaptID: string, p_expectScoreMagnifier: number): void
Description	Sets the value of the magnifier for the expected score compared to the opponent’s score.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_expectScoreMagnifier: number	Magnifier value.

Method	
Name	SetDefaultExpectScoreMagnifier(p_adaptID: string): void
Description	Sets to its default value the magnifier for the expected score compared to the opponent’s score.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Method

Name	GetMagnifierStepSize(p_adaptID: string): number
Description	Returns the value of the magnifier step size.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
Return type	**Return description**
number	Magnifier step size value.

Method	
Name	SetMagnifierStepSize(p_adaptID: string, p_magnifierStepSize: number): void
Description	Sets the value of the magnifier step size.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_magnifierStepSize: number	Magnifier step size value.

Method	
Name	SetDefaultMagnifierStepSize(p_adaptID: string): void
Description	Sets the magnifier step size to its default value.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

Methods for player data.

Method	
Name	AddPlayer(p_adaptID: string, p_gameID: string, p_playerID: string, p_rating: number, p_playCount: number, p_kFactor: number, p_uncertainty: number, p_lastPlayed: string): PlayerNode
Description	Creates a new instance of PlayerNode and adds it to the TwoA.players list. Requires custom parameter values. Ensures that all player parameters have valid values before creating the instance.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module to be used (see “Adaptation Modules”).
p_gameID: string	ID of a game to which the player instance belongs.

p_playerID: string	ID of a player. This ID is not allowed to duplicate for the same combination of the gameID and adaptID already present in the TwoA.players list.
p_rating: number	Player's skill rating.
p_playCount: number	The number of past gameplays that were used to assess player's skill rating. Should be a non-0 value.
p_kFactor: number	K factor. Should be higher than 0.
p_uncertainty: number	Uncertainty in player's rating. Should be a value between 0 and 1 (inclusive).
p_lastPlayed: string	The datetime of the last gameplay that was used to assess player's skill rating.
Return type	**Return description**
PlayerNode	Returns the newly created instance of the PlayerNode class.

Method	
Name	AddPlayerDefault(p_adaptID: string, p_gameID: string, p_playerID: string): PlayerNode
Description	Creates a new instance of PlayerNode and adds it to the TwoA.players list. Assigns default values to all player parameters.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module to be used (see "Adaptation Modules").
p_gameID: string	ID of a game to which the player instance belongs.
p_playerID: string	ID of a player. This ID is not allowed to duplicate for the same combination of the gameID and adaptID already present in the TwoA.players list.
Return type	**Return description**
PlayerNode	Returns the newly created instance of the PlayerNode class.

Method	
Name	AddPlayerNode(p_playerNode: PlayerNode): boolean
Description	Adds the instance of PlayerNode to the TwoA.players list. Ensures that all player parameters have valid values before adding the instance.
Parameter name	**Parameter description**
p_playerNode: PlayerNode	PlayerNode instance with new player data.
Return type	**Return description**

boolean	True if a new instance was successfully added, and False otherwise.

Method	
Name	RemovePlayer(p_adaptID: string, p_gameID: string, p_playerID: string): boolean
Description	Removes the matching instance of PlayerNode from the TwoA.players list. The instance is matched by the ID combination.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_playerID: string	ID of a player.
Return type	**Return description**
boolean	True if the instance was removed successfully, and False otherwise.

Method	
Name	RemovePlayerNode(p_playerNode: PlayerNode): boolean
Description	Removes the instance of PlayerNode from the TwoA.players list.
Parameter name	**Parameter description**
p_playerNode: PlayerNode	The instance to remove.
Return type	**Return description**
boolean	True if the instance was removed successfully, and False otherwise.

Method	
Name	Player(p_adaptID: string, p_gameID: string, p_playerID: string, p_showWarning: boolean = true): PlayerNode
Description	Returns the matching instance of PlayerNode from the TwoA.players list. The instance is matched by the ID combination.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_playerID: string	ID of a player.
p_showWarning: boolean	If true and if no matching player can be found then prints warning in the

	log.
Return type	**Return description**
PlayerNode	The matching instance. Null if no match is found or error occurred.

Method	
Name	AllPlayers(p_adaptID: string, p_gameID: string): PlayerNode[]
Description	Returns the all matching instances of PlayerNode from the TwoA.players list. The instances are matched by the ID combination.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
Return type	**Return description**
PlayerNode[]	The list of matching instances. Null if no match is found or error occurred.

Method	
Name	GetPlayerRating(p_adaptID: string, p_gameID: string, p_playerID: string): number
Description	Returns the skill rating for the specified player. Throws ReferenceError if PlayerNode instance is not found.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_playerID: string	ID of a player.
Return type	**Return description**
number	Skill rating.

Method	
Name	GetPlayerPlayCount(p_adaptID: string, p_gameID: string, p_playerID: string): number
Description	Returns the play count for the specified player. Throws ReferenceError if PlayerNode instance is not found.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).

p_gameID: string	ID of a game.
p_playerID: string	ID of a player.
Return type	**Return description**
number	Play count.

Method	
Name	GetPlayerKFactor(p_adaptID: string, p_gameID: string, p_playerID: string): number
Description	Returns the K factor for the specified player. Throws ReferenceError if PlayerNode instance is not found.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_playerID: string	ID of a player.
Return type	**Return description**
number	K factor.

Method	
Name	GetPlayerUncertainty(p_adaptID: string, p_gameID: string, p_playerID: string): number
Description	Returns the rating uncertainty for the specified player. Throws ReferenceError if PlayerNode instance is not found.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_playerID: string	ID of a player.
Return type	**Return description**
number	Rating uncertainty.

Method	
Name	GetPlayerLastPlayed(p_adaptID: string, p_gameID: string, p_playerID: string): string
Description	Returns a string representation of datetime indicating the last timestamp

	the player's skill rating was (re)assessed.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see "Adaptation Modules").
p_gameID: string	ID of a game.
p_playerID: string	ID of a player.
Return type	**Return description**
string	String representation of datetime.

Method	
Name	SetPlayerRating(p_adaptID: string, p_gameID: string, p_playerID: string, p_rating: number): boolean
Description	Sets the skill rating for the specified player.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see "Adaptation Modules").
p_gameID: string	ID of a game.
p_playerID: string	ID of a player.
p_rating: number	New skill rating value.
Return type	**Return description**
boolean	True if parameter value was set successfully, and false otherwise.

Method	
Name	SetPlayerPlayCount(p_adaptID: string, p_gameID: string, p_playerID: string, p_playCount: number): boolean
Description	Sets the play count for the specified player.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see "Adaptation Modules").
p_gameID: string	ID of a game.
p_playerID: string	ID of a player.
p_playCount: number	New play count. Positive Integer value.
Return type	**Return description**
boolean	True if parameter value was set successfully, and false otherwise.

Method	
Name	SetPlayerKFactor(p_adaptID: string, p_gameID: string, p_playerID: string, p_kFactor: number): boolean
Description	Sets the K factor for the specified player.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_playerID: string	ID of a player.
p_kFactor: number	New K factor. Positive non-0 value.
Return type	**Return description**
boolean	True if parameter value was set successfully, and false otherwise.

Method	
Name	SetPlayerUncertainty(p_adaptID: string, p_gameID: string, p_playerID: string, p_uncertainty: number): boolean
Description	Sets the rating uncertainty for the specified player.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_playerID: string	ID of a player.
p_uncertainty: number	New uncertainty. Value between 0 and 1 (inclusive).
Return type	**Return description**
boolean	True if parameter value was set successfully, and false otherwise.

Method	
Name	SetPlayerLastPlayed(p_adaptID: string, p_gameID: string, p_playerID: string, p_lastPlayed: string): boolean
Description	Sets the datetime indicating the last timestamp the player’s skill rating was (re)assessed.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_playerID: string	ID of a player.

p_lastPlayed: string	String representation of datetime of the last (re)assessment.
Return type	**Return description**
boolean	True if parameter value was set successfully, and false otherwise.

Methods for scenario data.

Method	
Name	AddScenario(p_adaptID: string, p_gameID: string, p_scenarioID: string, p_rating: number, p_playCount: number, p_kFactor: number, p_uncertainty: number, p_lastPlayed: string, p_timeLimit: number): ScenarioNode
Description	Creates a new instance of ScenarioNode and adds it to the TwoA.scenarios list. Requires custom parameter values. Ensures that all scenario parameters have valid values before creating the instance.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module to be used (see “Adaptation Modules”).
p_gameID: string	ID of a game to which the scenario instance belongs.
p_scenarioID: string	ID of a scenario. This ID is not allowed to duplicate for the same combination of the gameID and adaptID already present in the TwoA.scenarios list.
p_rating: number	Scenario’s skill rating.
p_playCount: number	The number of past gameplays that were used to assess scenario’s difficulty rating. Should be a non-0 value.
p_kFactor: number	K factor. Should be higher than 0.
p_uncertainty: number	Uncertainty in scenario’s rating. Should be a value between 0 and 1 (inclusive).
p_lastPlayed: string	The datetime of the last gameplay that was used to assess scenario’s difficulty rating.
p_timeLimit: number	Time limit within which a player should complete the scenario. Measured in milliseconds.
Return type	**Return description**
ScenarioNode	Returns the newly created instance of the ScenarioNode class.

Method	
Name	AddScenarioDefault(p_adaptID: string, p_gameID: string, p_scenarioID:

	string): ScenarioNode
Description	Creates a new instance of ScenarioNode and adds it to the TwoA.scenarios list. Assigns default values to all scenario parameters.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module to be used (see “Adaptation Modules”).
p_gameID: string	ID of a game to which the scenario instance belongs.
p_scenarioID: string	ID of a scenario. This ID is not allowed to duplicate for the same combination of the gameID and adaptID already present in the TwoA.scenarios list.
Return type	**Return description**
ScenarioNode	Returns the newly created instance of the ScenarioNode class.

Method	
Name	AddScenarioNode(p_scenarioNode: ScenarioNode): boolean
Description	Adds the instance of ScenarioNode to the TwoA.scenarios list. Ensures that all scenarios parameters have valid values before adding the instance.
Parameter name	**Parameter description**
p_scenarioNode: ScenarioNode	ScenarioNode instance with data for the new scenario.
Return type	**Return description**
boolean	True if a new instance was successfully created, and False otherwise.

Method	
Name	RemoveScenario(p_adaptID: string, p_gameID: string, p_scenarioID: string): boolean
Description	Removes the matching instance of ScenarioNode from the TwoA.scenarios list. The instance is matched by the ID combination.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.
Return type	**Return description**
boolean	True if the instance was removed successfully, and False otherwise.

Method	
Name	RemoveScenarioNode(p_scenarioNode: ScenarioNode): boolean
Description	Removes the instance of ScenarioNode from the TwoA.scenarios list.
Parameter name	**Parameter description**
p_scenarioNode: ScenarioNode	The instance to remove.
Return type	**Return description**
boolean	True if the instance was removed successfully, and False otherwise.

Method	
Name	Scenario(p_adaptID: string, p_gameID: string, p_scenarioID: string, p_showWarning: boolean = true): ScenarioNode
Description	Returns the matching instance of ScenarioNode from the TwoA.scenarios list. The instance is matched by the ID combination.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.
p_showWarning: boolean	If true and if no matching scenario can be found then prints warning in the log.
Return type	**Return description**
ScenarioNode	The matching instance. Null if no match is found or error occurred.

Method	
Name	AllScenarios(p_adaptID: string, p_gameID: string): ScenarioNode[]
Description	Returns the all matching instances of ScenarioNode from the TwoA.scenarios list. The instances are matched by the ID combination.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
Return type	**Return description**
ScenarioNode[]	The list of matching instances. Null if no match is found or error occurred.

Method	
Name	GetScenarioRating(p_adaptID: string, p_gameID: string, p_scenarioID: string): number
Description	Returns the difficulty rating for the specified scenario. Throws ReferenceError if ScenarioNode instance is not found.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.
Return type	**Return description**
number	Difficulty rating.

Method	
Name	GetScenarioPlayCount(p_adaptID: string, p_gameID: string, p_scenarioID: string): number
Description	Returns the play count for the specified scenario. Throws ReferenceError if ScenarioNode instance is not found.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.
Return type	**Return description**
number	Play count.

Method	
Name	GetScenarioKFactor(p_adaptID: string, p_gameID: string, p_scenarioID: string): number
Description	Returns the K factor for the specified scenario. Throws ReferenceError if ScenarioNode instance is not found.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.

p_scenarioID: string	ID of a scenario.
Return type	**Return description**
number	K factor.

Method	
Name	GetScenarioUncertainty(p_adaptID: string, p_gameID: string, p_scenarioID: string): number
Description	Returns the rating uncertainty for the specified scenario. Throws ReferenceError if ScenarioNode instance is not found.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.
Return type	**Return description**
number	Rating uncertainty.

Method	
Name	GetScenarioLastPlayed(p_adaptID: string, p_gameID: string, p_scenarioID: string): string
Description	Returns a string representing datetime indicating the last timestamp the scenario’s difficulty rating was (re)assessed.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.
Return type	**Return description**
string	String representing datetime.

Method	
Name	GetScenarioTimeLimit(p_adaptID: string, p_gameID: string, p_scenarioID: string): number
Description	Returns the time limit for the specified scenario. Throws ReferenceError if

	ScenarioNode instance is not found.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.
Return type	**Return description**
number	Time limit.

Method	
Name	SetScenarioRating(p_adaptID: string, p_gameID: string, p_scenarioID: string, p_rating: number): boolean
Description	Sets the difficulty rating for the specified scenario.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.
p_rating: number	New difficulty rating value.
Return type	**Return description**
boolean	True if parameter value was set successfully, and false otherwise.

Method	
Name	SetScenarioPlayCount(p_adaptID: string, p_gameID: string, p_scenarioID: string, p_playCount: number): boolean
Description	Sets the play count for the specified scenario.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.
p_playCount: number	New play count. Positive Integer value.
Return type	**Return description**
boolean	True if parameter value was set successfully, and false otherwise.

Method	
Name	SetScenarioKFactor(p_adaptID: string, p_gameID: string, p_scenarioID: string, p_kFactor: number): boolean
Description	Sets the K factor for the specified scenario.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.
p_kFactor: number	New K factor. Positive non-0 value.
Return type	**Return description**
boolean	True if parameter value was set successfully, and false otherwise.

Method	
Name	SetScenarioUncertainty(p_adaptID: string, p_gameID: string, p_scenarioID: string, p_uncertainty: number): boolean
Description	Sets the rating uncertainty for the specified scenario.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.
p_uncertainty: number	New uncertainty. Value between 0 and 1 (inclusive).
Return type	**Return description**
boolean	True if parameter value was set successfully, and false otherwise.

Method	
Name	SetScenarioLastPlayed(p_adaptID: string, p_gameID: string, p_scenarioID: string, p_lastPlayed: string): boolean
Description	Sets the datetime indicating the last timestamp the scenario’s difficulty rating was (re)assessed.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.

p_lastPlayed: string	Datetime of the last (re)assessment.
Return type	**Return description**
boolean	True if parameter value was set successfully, and false otherwise.

Method	
Name	SetScenarioTimeLimit(p_adaptID: string, p_gameID: string, p_scenarioID: string, p_timeLimit: number): boolean
Description	Sets the time limit for the specified scenario.
Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module (see “Adaptation Modules”).
p_gameID: string	ID of a game.
p_scenarioID: string	ID of a scenario.
p_timeLimit: number	New time limit. Positive non-0 value.
Return type	**Return description**
boolean	True if parameter value was set successfully, and false otherwise.

PlayerNode class

Class	
Class name	PlayerNode
Namespace	TwoANS
Description	Stores adaptation and assessment data for a player. An instance of this class is stored in 'public players: PlayerNode[]' of the TwoA class.

Constructor	
Name	constructor(p_adaptID: string, p_gameID: string, p_playerID: string)
Description	Initializes a new instance of the PlayerNode class with default parameter values.

Parameter name	**Parameter description**
p_adaptID: string	ID of an adaptation module to be used (see "Adaptation Modules").
p_gameID: string	ID of a game to which the player instance belongs.
p_playerID: string	ID of a player.

Property name	**Property description**
AdaptationID: string	Gets or sets ID of an adaptation module to be used (see "Adaptation Modules").
GameID: string	Gets or sets ID of a game to which the player instance belongs.
PlayerID: string	Gets or sets ID of a player.
Rating: number	Gets or sets player's skill rating.
PlayCount: number	Gets or sets the number of past gameplays that were used to assess player's skill rating. Should be an integer that is 0 or higher.
KFactor: number	Gets or sets the K factor. Should be a double that is higher than 0.
Uncertainty: number	Gets or sets the uncertainty in player's rating. Should be a value between 0 and 1 (inclusive at both ends).
LastPlayed: string	Gets or sets the datetime of the last gameplay that was used to assess player's skill rating.

Method

Name	PlayerNode ShallowClone()
Description	Creates and returns a shallow clone of the instance.
Return type	**Return description**
PlayerNode	New instance of PlayerNode.

ScenarioNode class

Class	
Class name	ScenarioNode
Namespace	TwoANS
Description	Stores adaptation and assessment data for a scenario. An instance of this class is stored in 'public scenarios: ScenarioNode[]' of the TwoA class.

Constructor	
Name	constructor(p_adaptID: string, p_gameID: string, p_scenarioID: string)
Description	Initializes a new instance of the ScenarioNode class with default parameter values.

Parameter name	Parameter description
p_adaptID: string	ID of an adaptation module to be used (see "Adaptation Modules").
p_gameID: string	ID of a game to which the scenario instance belongs.
p_scenarioID: string	ID of a scenario.

Property name	Property description
AdaptationID: string	Gets or sets ID of an adaptation module to be used (see "Adaptation Modules").
GameID: string	Gets or sets ID of a game to which the player instance belongs.
ScenarioID: string	Gets or sets ID of a scenario.
Rating: number	Gets or sets scenario's difficulty rating.
PlayCount: number	Gets or sets the number of past gameplays that were used to assess scenario's difficulty rating. Should be an integer that is 0 or higher.
KFactor: number	Gets or sets the K factor. Should be double value higher than 0.
Uncertainty: number	Gets or sets the uncertainty in scenario's rating. Should be a double value between 0 and 1 (inclusive at both ends).
LastPlayed: string	Gets or sets the datetime of the last gameplay that was used to assess scenario's difficulty rating.
TimeLimit: number	Gets or sets the time limit within which a player should complete the scenario. Measured in milliseconds. Should be an integer higher than 0.

Method	
Name	ShallowClone(): ScenarioNode
Description	Creates and returns a shallow clone of the instance.
Return type	**Return description**
ScenarioNode	New instance of ScenarioNode.

Gameplay class

Class	
Class name	Gameplay
Namespace	TwoANS
Description	Stores results of a player assessment. An instance of this class is stored in 'public gameplays: Gameplay[]' of the TwoA class.

Constructor	
Name	constructor()
Description	Initializes a new instance of the Gameplay class.

Property name	**Property description**
AdaptationID: string	Gets or sets ID of an adaptation module to be used (see "Adaptation Modules").
GameID: string	Gets or sets ID of a game to which the player instance belongs.
PlayerID: string	Gets or sets ID of a player that was assessed.
ScenarioID: string	Gets or sets ID of a scenario in which player's performance was measured.
Timestamp: string	Gets or sets the datetime of assessment in the format of 'yyyy-MM-ddThh:mm:ss'.
RT: number	Gets or sets the player's response time. Measured in milliseconds.
Accuracy: number	Gets or sets the player's accuracy. A value between 0 and 1 inclusive.
PlayerRating: number	Gets or sets the player's skill rating after assessment.
ScenarioRating: number	Gets or sets the scenario's difficulty rating after assessment.