

TwoA API Manual

Contents

Summary	2
Adaptation Modules	3
Rating Scale	4
TwoA class.....	6
Methods for adaptation.....	7
Methods for assessment.....	10
Methods for scoring.....	12
Methods for controlling success rate parameter.	13
Methods for controlling the fuzzy selection intervals.	14
Methods for the uncertainty parameter.	15
Methods for K factor.....	17
Methods for the calibration parameters.	19
Methods controlling ELO parameters.....	23
Methods for player data.	24
Methods for scenario data.....	31
PlayerNode class	39
ScenarioNode class	41
Gameplay class.....	43

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	2018.01.22
Deployment side	client-side
Programming language	C#
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 indicate the module that should be used.

Adaptation Module 1

Adaptation 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 <i>double</i> 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 <i>double</i> type. The duration of time the player required to complete (either successfully or unsuccessfully) a game scenario. Time is measured in milliseconds.

Adaptation Module 2

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

TwoA class

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

Constant name	Constant description
public const string DATE_FORMAT	A string representation of the datetime format used by TwoA. The value is 'yyyy-MM-ddThh:mm:ss'.

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

Constructor	
Name	TwoA(IBridge bridge)
Description	Initializes a new instance of the TwoA class with an instance of an IBridge. If available, the bridge is used by TwoA for logging.

Parameter name	Parameter description
IBridge bridge	Instance of IBridge as defined in the RAGE client-side asset architecture.

Field name	Field description
public List<PlayerNode> players	A list of PlayerNode instances. An empty list is automatically initialized during constructor call. Each PlayerNode instance contains data of a single player. Refer to PlayerNode section for more information.
public List<ScenarioNode> scenarios	A list of ScenarioNode instances. An empty list is automatically initialized during constructor call. Each ScenarioNode instance contains data of a single game scenario. Refer to ScenarioNode section for more information.

public List<Gameplay> gameplays	A list of Gameplay instances. An empty list 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	string[,] AvailableAdapters()
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	string TargetScenarioID(string adaptID, string gameId, string playerId)
Description	Returns ID of a game scenario with a difficulty rating that matches the skill rating of a specified player. The player data is retrieved from the TwoA.players list. The recommended scenario is selected from TwoA.scenarios list.
Parameter name	Parameter description
string adaptID	ID of an adaptation module to be used (refer to “Adaptation modules”).
string gameId	ID of a game from which a scenario will be selected.
string playerId	ID 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	string TargetScenarioID(PlayerNode playerNode)
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
PlayerNode playerName	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	ScenarioNode TargetScenario(string adaptID, string gameId, string playerId)
Description	Returns an instance of ScenarioNode of a game scenario with a difficulty rating that matches the skill rating of a specified player. The player data is retrieved from the TwoA.players list. The recommended scenario is selected from TwoA.scenarios list.
Parameter name	Parameter description
string adaptID	ID of an adaptation module to be used (refer to “Adaptation modules”).
string gameId	ID of a game from which a scenario will be selected.
string playerId	ID 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	ScenarioNode TargetScenario(PlayerNode playerName)
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
PlayerNode playerName	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	ScenarioNode TargetScenario(PlayerNode playerName, List<ScenarioNode> scenarioList)

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
PlayerNode playerNode	PlayerNode of a player to whom the scenario difficulty should be matched.
List<ScenarioNode> scenarioList	A list of ScenarioNode instances from which a scenario will be selected and 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	double TargetDifficultyRating(string adaptID, double playerRating)
Description	Returns a recommended difficulty rating for a specified player skill rating.
Parameter name	Parameter description
string adaptID	ID of an adaptation module to be used (refer to "Adaptation modules").
double playerRating	Player's skill rating.
Return type	Return description
double	Difficulty rating.

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

Method	
Name	double TargetDifficultyRating(string adaptID, string gameId, string playerId)
Description	Returns a recommended difficulty rating for a specified player. The player data is retrieved from the TwoA.players list.

Parameter name	Parameter description
string adaptID	ID of an adaptation module to be used (refer to “Adaptation modules”).
string gameId	ID of a game from which a scenario will be selected.
string playerId	ID of a player to whom the scenario difficulty should be matched.
Return type	Return description
double	Difficulty rating.

Methods for assessment.

Method	
Name	bool UpdateRatings(string adaptID, string gameId, string playerId, string scenarioID, double rt, double correctAnswer, bool updateScenarioRating, double customKfct)
Description	Reassesses and updates player skill rating and, optionally, scenario difficulty rating based on player’s performance in a specified scenario. The player and scenario data is retrieved from the TwoA.players and TwoA.scenarios lists respectively.
Parameter name	Parameter description
string adaptID	ID of an adaptation module to be used (see “Adaptation Modules”).
string gameId	ID of a game from which a scenario will be selected.
string playerId	ID of a player to assess.
string scenarioID	ID of a scenario in which player’s performance was measured.
double rt	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.
double correctAnswer	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.
bool updateScenarioRating	If true scenario’s difficulty rating will be reassessed and updated.
double customKfct	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
bool	True if ratings were reassessed and updated successfully, and false otherwise.

Method

Name	bool UpdateRatings(PlayerNode playerNode, ScenarioNode scenarioNode, double rt, double correctAnswer, bool updateScenarioRating, double customKfct)
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
PlayerNode playerNode	PlayerNode of a player to assess.
ScenarioNode scenarioNode	ScenarioNode a scenario in which player's performance was measure.
double rt	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.
double correctAnswer	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.
bool updateScenarioRating	If true scenario's difficulty rating will be reassessed and updated.
double customKfct	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
bool	True if ratings were reassessed and updated successfully, and false otherwise.

Method

Name	void CreateNewRecord(string adaptID, string gameID, string playerID, string scenarioID, double rt, double accuracy, double playerRating, double scenarioRating, DateTime timestamp)
------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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
string adaptID	ID of an adaptation module to be used (see “Adaptation Modules”).
string gameId	ID of a game from which a scenario was selected.
string playerId	ID of a player that was assessed.
string scenarioID	ID of a scenario in which player’s performance was measured.
double rt	Player’s response time in milliseconds (see “Adaptation Modules”).
double accuracy	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.
double playerRating	Player’s skill rating after reassessment.
double scenarioRating	Scenario’s difficulty rating after reassessment.
DateTime timestamp	Date and time of reassessment.

Methods for scoring.

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

Method	
Name	double CalculateExpectedScore(string adaptID, double playerRating, double scenarioRating, double itemMaxDuration)

Description	Calculates player's expected score based on player's skill rating and scenarios difficulty rating.
Parameter name	Parameter description
string adaptID	ID of the adaptation module to use.
double playerRating	Player's skill rating.
double scenarioRating	Scenario's difficulty rating.
double itemMaxDuration	Max allowed time in millisecond given to player to solve the problem.
Return type	Return description
double	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	void SetTargetDistribution(string adaptID, double mean, double sd, double lowerLimit, double upperLimit)
Description	Sets the parameters for the target success rate that is used to select a scenario of a recommended difficulty.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see "Adaptation Modules").
double mean	Mean of a normal distribution. Any value between 0 and 1 (exclusive).
double sd	Standard deviation of a normal distribution. Any value between 0 and 1

	(exclusive).
double lowerLimit	Lower hard boundary. Any value between 0 and 1 (inclusive). Should be less than standard distribution mean.
double upperLimit	Upper hard boundary. Any value between 0 and 1 (inclusive). Should be higher than standard distribution mean.

Method	
Name	double[] GetTargetDistribution(string adaptID)
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
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
Return type	Return description
double[]	An array of four values: distribution mean, distribution standard deviation, lower hard boundary, and upper hard boundary.

Method	
Name	void SetDefaultTargetDistribution(string adaptID)
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
string adaptID	ID of an adaptation module (see “Adaptation Modules”).

Methods for controlling the fuzzy selection intervals.

Method	
Name	double GetFiSDMultiplier(string adaptID)
Description	Returns the multiplier for the standard deviation used for calculating the support fuzzy interval.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
Return type	Return description

double	Multiplier value, or 0 if the adapter is not found.
--------	-----------------------------------------------------

Method

Name	void SetFiSDMultiplier(string adaptID, double multiplier)
Description	Sets a value for the multiplier for the standard deviation used for calculating the support fuzzy interval.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
double multiplier	The value of the multiplier.

Method

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

Methods for the uncertainty parameter.

Method

Name	double GetMaxDelay(string adaptID)
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
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
Return type	Return description
double	The number of days as double value.

Method

Name	void SetMaxDelay(string adaptID, double maxDelay)
------	---------------------------------------------------

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
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
double maxDelay	The value in the number of days.

Method

Name	void SetDefaultMaxDelay(string adaptID)
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
string adaptID	ID of an adaptation module (see “Adaptation Modules”).

Method

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

Method

Name	void SetMaxPlay(string adaptID, double maxPlay)
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
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
double maxPlay	The value in the number of gameplays.

Method	
Name	void SetDefaultMaxPlay(string adaptID)
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
string adaptID	ID of an adaptation module (see "Adaptation Modules").

Methods for K factor.

Method	
Name	double GetKConst(string adaptID)
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
string adaptID	ID of an adaptation module (see "Adaptation Modules").
Return type	Return description
double	K factor value as double.

Method	
Name	void SetKConst(string adaptID, double kConst)
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
string adaptID	ID of an adaptation module (see "Adaptation Modules").
double kConst	The min value for the K factor.

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

Method	
Name	double GetKUp(string adaptID)
Description	Gets the step value by which the K factor should increase.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
Return type	Return description
double	K factor step value.

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

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

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

Method	
--------	--

Name	void SetKDown(string adaptID, double kDown)
Description	Sets the step value by which the K factor should decrease.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
double kDown	The step value.

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

Methods for the calibration parameters.

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

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

Method	
Name	void SetDefaultPlayerCalLength(string adaptID)
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
string adaptID	ID of an adaptation module (see "Adaptation Modules").

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

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

Method	
Name	void SetDefaultScenarioCalLength(string adaptID)
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
string adaptID	ID of an adaptation module (see "Adaptation Modules").

Method	
--------	--

Name	void SetCalLength(string adaptID, int calLength)
Description	Sets the scenario and player calibration lengths to the same value. The length is measured in number of gameplays.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
int calLength	The length in the number of gameplays.

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

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

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

Method	
Name	void SetDefaultPlayerCalK(string adaptID)

Description	Sets to its default value the custom K factor used during player's calibration.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see "Adaptation Modules").

Method	
Name	double GetScenarioCalK(string adaptID)
Description	Returns the custom K factor used during scenario's calibration.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see "Adaptation Modules").
Return type	Return description
double	K factor value as double.

Method	
Name	void SetScenarioCalK(string adaptID, double calK)
Description	Sets the custom K factor used during scenario's calibration.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see "Adaptation Modules").
double calK	K factor value as double.

Method	
Name	void SetDefaultScenarioCalK(string adaptID)
Description	Sets to its default value the custom K factor used during scenario's calibration.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see "Adaptation Modules").

Method	
Name	void SetCalK(string adaptID, double calK)
Description	Sets the custom K factor used during both player's and scenario's calibrations.
Parameter name	Parameter description

string adaptID	ID of an adaptation module (see “Adaptation Modules”).
double calK	K factor value as double.

Method

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

Parameter name Parameter description

string adaptID	ID of an adaptation module (see “Adaptation Modules”).
----------------	--------------------------------------------------------

Methods controlling ELO parameters.

Method

Name	double GetExpectScoreMagnifier(string adaptID)
Description	Returns the value of the magnifier for the expected score compared to the opponent’s score.

Parameter name Parameter description

string adaptID	ID of an adaptation module (see “Adaptation Modules”).
----------------	--------------------------------------------------------

Return type Return description

double	Magnifier value.
--------	------------------

Method

Name	void SetExpectScoreMagnifier(string adaptID, double expectScoreMagnifier)
Description	Sets the value of the magnifier for the expected score compared to the opponent’s score.

Parameter name Parameter description

string adaptID	ID of an adaptation module (see “Adaptation Modules”).
----------------	--------------------------------------------------------

double expectScoreMagnifier	Magnifier value.
--------------------------------	------------------

Method

Name	void SetDefaultExpectScoreMagnifier(string adaptID)
------	-----------------------------------------------------

Description	Sets to its default value the magnifier for the expected score compared to the opponent's score.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see "Adaptation Modules").

Method	
Name	double GetMagnifierStepSize(string adaptID)
Description	Returns the value of the magnifier step size.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see "Adaptation Modules").
Return type	Return description
double	Magnifier step size value.

Method	
Name	void SetMagnifierStepSize(string adaptID, double magnifierStepSize)
Description	Sets the value of the magnifier step size.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see "Adaptation Modules").
double magnifierStepSize	Magnifier step size value.

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

Methods for player data.

Method	
Name	bool AddPlayer(string adaptID, string gameId, string playerId, double rating, double playCount, double kFactor, double uncertainty, DateTime lastPlayed)

Description	Creates a new instance of <code>PlayerNode</code> and adds it to the <code>TwoA.players</code> list. Requires custom parameter values. Ensures that all player parameters have valid values before creating the instance.
Parameter name	Parameter description
string <code>adaptID</code>	ID of an adaptation module to be used (see “Adaptation Modules”).
string <code>gameID</code>	ID of a game to which the player instance belongs.
string <code>playerID</code>	ID of a player. This ID is not allowed to duplicate for the same combination of the <code>gameID</code> and <code>adaptID</code> already present in the <code>TwoA.players</code> list.
double <code>rating</code>	Player’s skill rating.
double <code>playCount</code>	The number of past gameplays that were used to assess player’s skill rating. Should be a non-0 value.
double <code>kFactor</code>	K factor. Should be higher than 0.
double <code>uncertainty</code>	Uncertainty in player’s rating. Should be a value between 0 and 1 (inclusive).
DateTime <code>lastPlayed</code>	The datetime of the last gameplay that was used to assess player’s skill rating.
Return type	Return description
bool	True if a new instance was successfully created, and False otherwise.

Method	
Name	bool <code>AddPlayer(string adaptID, string gameID, string playerID)</code>
Description	Creates a new instance of <code>PlayerNode</code> and adds it to the <code>TwoA.players</code> list. Assigns default values to all player parameters.
Parameter name	Parameter description
string <code>adaptID</code>	ID of an adaptation module to be used (see “Adaptation Modules”).
string <code>gameID</code>	ID of a game to which the player instance belongs.
string <code>playerID</code>	ID of a player. This ID is not allowed to duplicate for the same combination of the <code>gameID</code> and <code>adaptID</code> already present in the <code>TwoA.players</code> list.
Return type	Return description
bool	True if a new instance was successfully created, and False otherwise.

Method	
Name	bool <code>AddPlayer(PlayerNode playerNode)</code>
Description	Adds the instance of <code>PlayerNode</code> to the <code>TwoA.players</code> list. Ensures that all

	player parameters have valid values before adding the instance.
Parameter name	Parameter description
PlayerNode playerNode	PlayerNode instance with new player data.
Return type	Return description
bool	True if a new instance was successfully added, and False otherwise.

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

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

Method	
Name	PlayerNode Player(string adaptID, string gameId, string playerId)
Description	Returns the matching instance of PlayerNode from the TwoA.players list. The instance is matched by the ID combination.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.

string playerId	ID of a player.
Return type	Return description
PlayerNode	The matching instance. Null if no match is found or error occurred.

Method	
Name	List<PlayerNode> AllPlayers(string adaptID, string gameId)
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
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
Return type	Return description
List<PlayerNode>	The list of matching instances. Null if no match is found or error occurred.

Method	
Name	double PlayerRating(string adaptID, string gameId, string playerId)
Description	Returns the skill rating for the specified player. Throws NullReferenceException if PlayerNode instance is not found.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string playerId	ID of a player.
Return type	Return description
double	Skill rating.

Method	
Name	double PlayerPlayCount(string adaptID, string gameId, string playerId)
Description	Returns the play count for the specified player. Throws NullReferenceException if PlayerNode instance is not found.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.

string playerId	ID of a player.
Return type	Return description
double	Play count.

Method

Name	double PlayerKFactor(string adaptID, string gameId, string playerId)
Description	Returns the K factor for the specified player. Throws NullReferenceException if PlayerNode instance is not found.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string playerId	ID of a player.
Return type	Return description
double	K factor.

Method

Name	double PlayerUncertainty(string adaptID, string gameId, string playerId)
Description	Returns the rating uncertainty for the specified player. Throws NullReferenceException if PlayerNode instance is not found.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string playerId	ID of a player.
Return type	Return description
double	Rating uncertainty.

Method

Name	DateTime PlayerLastPlayed(string adaptID, string gameId, string playerId)
Description	Returns DateTime instance indicating the last timestamp the player’s skill rating was (re)assessed.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).

string gameId	ID of a game.
string playerId	ID of a player.
Return type	Return description
DateTime	DateTime instance.

Method	
Name	bool PlayerRating(string adaptID, string gameId, string playerId, double rating)
Description	Sets the skill rating for the specified player.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string playerId	ID of a player.
double rating	New skill rating value.
Return type	Return description
bool	True if parameter value was set successfully, and false otherwise.

Method	
Name	bool PlayerPlayCount(string adaptID, string gameId, string playerId, double playCount)
Description	Sets the play count for the specified player.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string playerId	ID of a player.
double playCount	New play count. Positive Integer value.
Return type	Return description
bool	True if parameter value was set successfully, and false otherwise.

Method	
Name	bool PlayerKFactor(string adaptID, string gameId, string playerId, double kFactor)

Description	Sets the K factor for the specified player.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string playerId	ID of a player.
double kFactor	New K factor. Positive non-0 value.
Return type	Return description
bool	True if parameter value was set successfully, and false otherwise.

Method	
Name	bool PlayerUncertainty(string adaptID, string gameId, string playerId, double uncertainty)
Description	Sets the rating uncertainty for the specified player.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string playerId	ID of a player.
double uncertainty	New uncertainty. Value between 0 and 1 (inclusive).
Return type	Return description
bool	True if parameter value was set successfully, and false otherwise.

Method	
Name	bool PlayerLastPlayed(string adaptID, string gameId, string playerId, DateTime lastPlayed)
Description	Sets the DateTime instance indicating the last timestamp the player’s skill rating was (re)assessed.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string playerId	ID of a player.
DateTime lastPlayed	Datetime of the last (re)assessment.
Return type	Return description

bool	True if parameter value was set successfully, and false otherwise.
------	--------------------------------------------------------------------

Methods for scenario data.

Method	
Name	bool AddScenario(string adaptID, string gameId, string scenarioID, double rating, double playCount, double kFactor, double uncertainty, DateTime lastPlayed, double timeLimit)
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
string adaptID	ID of an adaptation module to be used (see “Adaptation Modules”).
string gameId	ID of a game to which the scenario instance belongs.
string scenarioID	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.
double rating	Scenario’s skill rating.
double playCount	The number of past gameplays that were used to assess scenario’s difficulty rating. Should be a non-0 value.
double kFactor	K factor. Should be higher than 0.
double uncertainty	Uncertainty in scenario’s rating. Should be a value between 0 and 1 (inclusive).
DateTime lastPlayed	The datetime of the last gameplay that was used to assess scenario’s difficulty rating.
double timeLimit	Time limit within which a player should complete the scenario. Measured in milliseconds.
Return type	Return description
bool	True if a new instance was successfully created, and False otherwise.

Method	
Name	bool AddScenario(string adaptID, string gameId, string scenarioID)
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
string adaptID	ID of an adaptation module to be used (see “Adaptation Modules”).
string gameId	ID of a game to which the scenario instance belongs.
string scenarioID	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
bool	True if a new instance was successfully created, and False otherwise.

Method	
Name	bool AddScenario(ScenarioNode scenarioNode)
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
string adaptID	ID of an adaptation module to be used (see “Adaptation Modules”).
string gameId	ID of a game to which the scenario instance belongs.
ScenarioNode scenarioNode	ScenarioNode instance with data for the new scenario.
Return type	Return description
bool	True if a new instance was successfully created, and False otherwise.

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

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

Method	
Name	ScenarioNode Scenario(string adaptID, string gameId, string scenarioID)
Description	Returns the matching instance of ScenarioNode from the TwoA.scenarios list. The instance is matched by the ID combination.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string scenarioID	ID of a scenario.
Return type	Return description
ScenarioNode	The matching instance. Null if no match is found or error occurred.

Method	
Name	List<ScenarioNode> AllScenarios(string adaptID, string gameId)
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
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
Return type	Return description
List<ScenarioNode>	The list of matching instances. Null if no match is found or error occurred.

Method	
Name	double ScenarioRating(string adaptID, string gameId, string scenarioID)

Description	Returns the difficulty rating for the specified scenario. Throws <code>NullPointerException</code> if <code>ScenarioNode</code> instance is not found.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string scenarioID	ID of a scenario.
Return type	Return description
double	Difficulty rating.

Method	
Name	double ScenarioPlayCount(string adaptID, string gameId, string scenarioID)
Description	Returns the play count for the specified scenario. Throws <code>NullPointerException</code> if <code>ScenarioNode</code> instance is not found.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string scenarioID	ID of a scenario.
Return type	Return description
double	Play count.

Method	
Name	double ScenarioKFactor(string adaptID, string gameId, string scenarioID)
Description	Returns the K factor for the specified scenario. Throws <code>NullPointerException</code> if <code>ScenarioNode</code> instance is not found.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string scenarioID	ID of a scenario.
Return type	Return description
double	K factor.

Method	
Name	double ScenarioUncertainty(string adaptID, string gameId, string scenarioID)
Description	Returns the rating uncertainty for the specified scenario. Throws NullReferenceException if ScenarioNode instance is not found.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string scenarioID	ID of a scenario.
Return type	Return description
double	Rating uncertainty.

Method	
Name	DateTime ScenarioLastPlayed(string adaptID, string gameId, string scenarioID)
Description	Returns DateTime instance indicating the last timestamp the scenario’s difficulty rating was (re)assessed.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string scenarioID	ID of a scenario.
Return type	Return description
DateTime	DateTime instance.

Method	
Name	double ScenarioTimeLimit(string adaptID, string gameId, string scenarioID)
Description	Returns the time limit for the specified scenario. Throws NullReferenceException if ScenarioNode instance is not found.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string scenarioID	ID of a scenario.
Return type	Return description

double Time limit.

Method

Name	bool ScenarioRating(string adaptID, string gameID, string scenarioID, double rating)
Description	Sets the difficulty rating for the specified scenario.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameID	ID of a game.
string scenarioID	ID of a scenario.
double rating	New difficulty rating value.
Return type	Return description
bool	True if parameter value was set successfully, and false otherwise.

Method

Name	bool ScenarioPlayCount(string adaptID, string gameID, string scenarioID, double playCount)
Description	Sets the play count for the specified scenario.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameID	ID of a game.
string scenarioID	ID of a scenario.
double playCount	New play count. Positive Integer value.
Return type	Return description
bool	True if parameter value was set successfully, and false otherwise.

Method

Name	bool ScenarioKFactor(string adaptID, string gameID, string scenarioID, double kFactor)
Description	Sets the K factor for the specified scenario.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).

string gameId	ID of a game.
string scenarioID	ID of a scenario.
double kFactor	New K factor. Positive non-0 value.
Return type	Return description
bool	True if parameter value was set successfully, and false otherwise.

Method	
Name	bool ScenarioUncertainty(string adaptID, string gameId, string scenarioID, double uncertainty)
Description	Sets the rating uncertainty for the specified scenario.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string scenarioID	ID of a scenario.
double uncertainty	New uncertainty. Value between 0 and 1 (inclusive).
Return type	Return description
bool	True if parameter value was set successfully, and false otherwise.

Method	
Name	bool ScenarioLastPlayed(string adaptID, string gameId, string scenarioID, DateTime lastPlayed)
Description	Sets the DateTime instance indicating the last timestamp the scenario’s difficulty rating was (re)assessed.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameId	ID of a game.
string scenarioID	ID of a scenario.
DateTime lastPlayed	Datetime of the last (re)assessment.
Return type	Return description
bool	True if parameter value was set successfully, and false otherwise.

Method	
Name	bool ScenarioTimeLimit(string adaptID, string gameID, string scenarioID, double timeLimit)
Description	Sets the time limit for the specified scenario.
Parameter name	Parameter description
string adaptID	ID of an adaptation module (see “Adaptation Modules”).
string gameID	ID of a game.
string scenarioID	ID of a scenario.
double timeLimit	New time limit. Positive non-0 value.
Return type	Return description
bool	True if parameter value was set successfully, and false otherwise.

PlayerNode class

Class	
Class name	PlayerNode
Namespace	TwoANS
Assembly	TwoA
Description	Stores adaptation and assessment data for a player.

Constructor	
Name	PlayerNode(string adaptID, string gameId, string playerId)
Description	Initializes a new instance of the PlayerNode class with default parameter values.
Parameter name	Parameter description
string adaptID	ID of an adaptation module to be used (see “Adaptation Modules”).
string gameId	ID of a game to which the player instance belongs.
string playerId	ID of a player.

Constructor	
Name	PlayerNode(string adaptID, string gameId, string playerId, double rating, double playCount, double kFct, double uncertainty, DateTime lastPlayed)
Description	Initializes a new instance of the PlayerNode class with custom parameter values.
Parameter name	Parameter description
string adaptID	ID of an adaptation module to be used (see “Adaptation Modules”).
string gameId	ID of a game to which the player instance belongs.
string playerId	ID of a player.
double rating	Player’s skill rating.
double playCount	The number of past gameplays that were used to assess player’s skill rating. Should be a 0 or higher.
double kFactor	K factor. Should be higher than 0.
double uncertainty	Uncertainty in player’s rating. Should be a value between 0 and 1 (inclusive)

at both ends).

DateTime lastPlayed	The datetime of the last gameplay that was used to assess player's skill rating.
---------------------	----------------------------------------------------------------------------------

Property name	Property description
String AdaptationID	Gets or sets ID of an adaptation module to be used (see "Adaptation Modules").
String GameID	Gets or sets ID of a game to which the player instance belongs.
String PlayerID	Gets or sets ID of a player.
Double Rating	Gets or sets player's skill rating.
Double PlayCount	Gets or sets the number of past gameplays that were used to assess player's skill rating. Should be 0 or higher.
Double KFactor	Gets or sets the K factor. Should be higher than 0.
Double Uncertainty	Gets or sets the uncertainty in player's rating. Should be a value between 0 and 1 (inclusive at both ends).
DateTime LastPlayed	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
Assembly	TwoA
Description	Stores adaptation and assessment data for a scenario.

Constructor	
Name	ScenarioNode(string adaptID, string gameId, string scenarioID)
Description	Initializes a new instance of the ScenarioNode class with default parameter values.
Parameter name	Parameter description
string adaptID	ID of an adaptation module to be used (see “Adaptation Modules”).
string gameId	ID of a game to which the scenario instance belongs.
string scenarioID	ID of a scenario.

Constructor	
Name	ScenarioNode(string adaptID, string gameId, string scenarioID, double rating, double playCount, double kFct, double uncertainty, DateTime lastPlayed, double timeLimit)
Description	Initializes a new instance of the ScenarioNode class with custom parameter values.
Parameter name	Parameter description
string adaptID	ID of an adaptation module to be used (see “Adaptation Modules”).
string gameId	ID of a game to which the scenario instance belongs.
string scenarioID	ID of a scenario.
double rating	Scenario’s difficulty rating.
double playCount	The number of past gameplays that were used to assess scenario’s difficulty rating. Should be a 0 or higher.
double kFactor	K factor. Should be higher than 0.

double uncertainty	Uncertainty in scenario's rating. Should be a value between 0 and 1 (inclusive at both ends).
DateTime lastPlayed	The datetime of the last gameplay that was used to assess scenario's difficulty rating.
double timeLimit	Time limit within which a player should complete the scenario. Measured in milliseconds. Should be higher than 0.

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

Method	
Name	ScenarioNode ShallowClone()
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
Assembly	TwoA
Description	Stores results of a player assessment. An instance of this class is stored in 'public List<Gameplay> gameplays' of the TwoA class.

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

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