

## Contents

Demo.....	1
How it works?.....	2
How to integrate it?.....	3
How to create dialogue file?.....	4
Pages.....	5
Lines.....	5
Variables.....	6
How to control dialogue?.....	7

## Demo

Import asset and run Scene1 for features demonstration.

## How it works?

CSV based Dialogue System uses Comma-Separated Values (CSV) files to display dialogue between player and Non-Player Character (NPC). Such files can be created with any text/spreadsheets editor.

*Example: part of dialog CSV file*

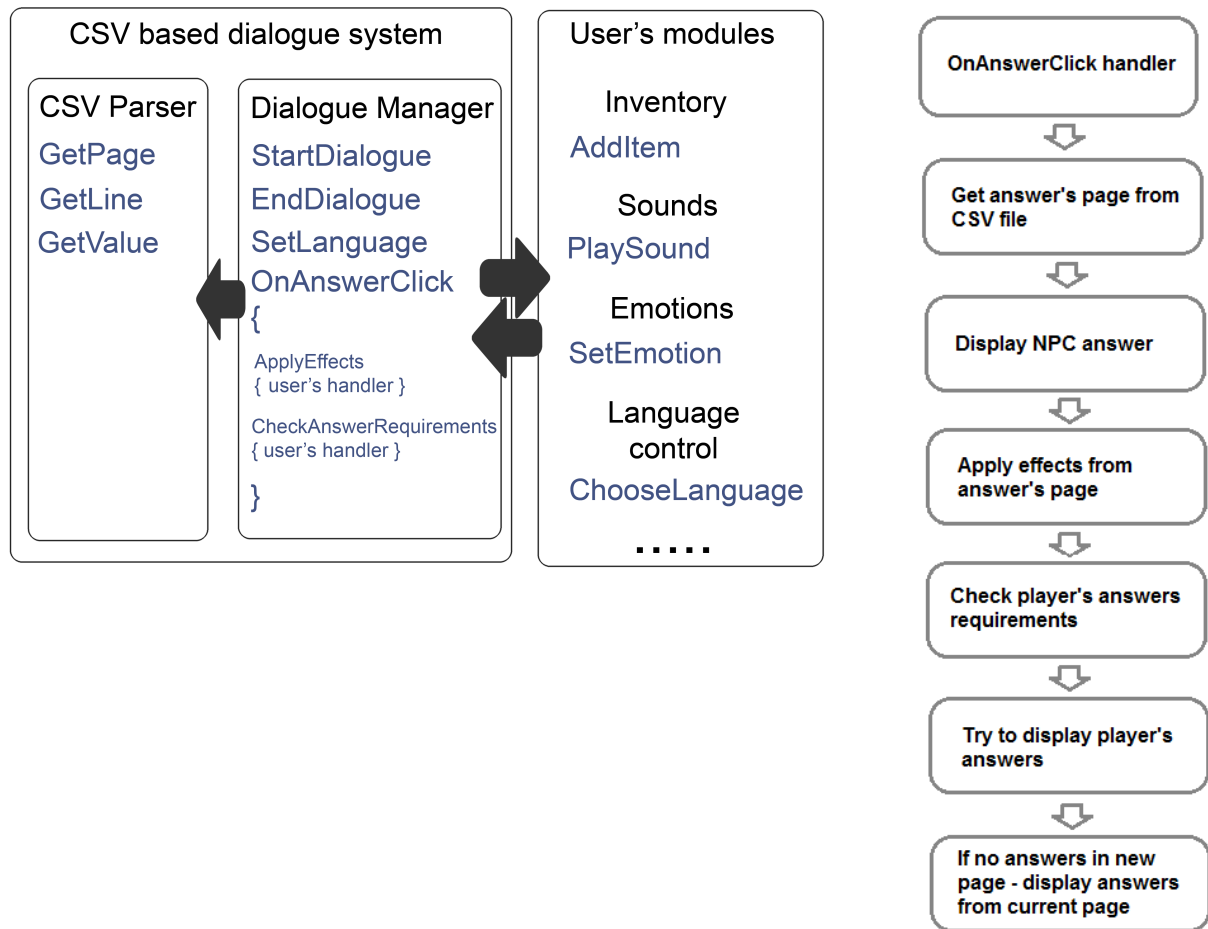
	A	B	C	D	E
1	Page_Desc				
2	Line_NpcName	Eng	John	Ital	Gianni
3					
4	Page_Welcome				
5	Line_NpcAnswer	Eng	Do you speak English?	Ital	Parla Italiano?
6	Line_Common	Eng	Sure	Ital	Certamente
7					
8	Page_Common				
9	Line_NpcAnswer	Eng	Can I help you?	Ital	Posso aiutarla?
10	Line_BuyProvision	Eng	Bring me some food. (1 gold coin)	Ital	Mi porti un po' di cibo. (1 moneta d'oro )
11	Line_BuyBarrel	Eng	I want to buy a barrel of beer. (2 gold coins)	Ital	Voglio comprare un barile di birra
12	Line_StayByNight	Eng	I need a room for a night	Ital	Ho bisogno di una stanza per una notte
13	Line_BorrowGold	Eng	I need money. Can you lend me some?	Ital	Ho bisogno di soldi. Mi puoi prestarmi?
14	Line_Leave	Eng	I have to go	Ital	Devo andare
15					

File has “.csv” filename extension and Unity converts it to TextAsset format.

There are two C# scripts to operate dialogue from files and display it in game:

- **CsvParser**
- **DialogueManager**

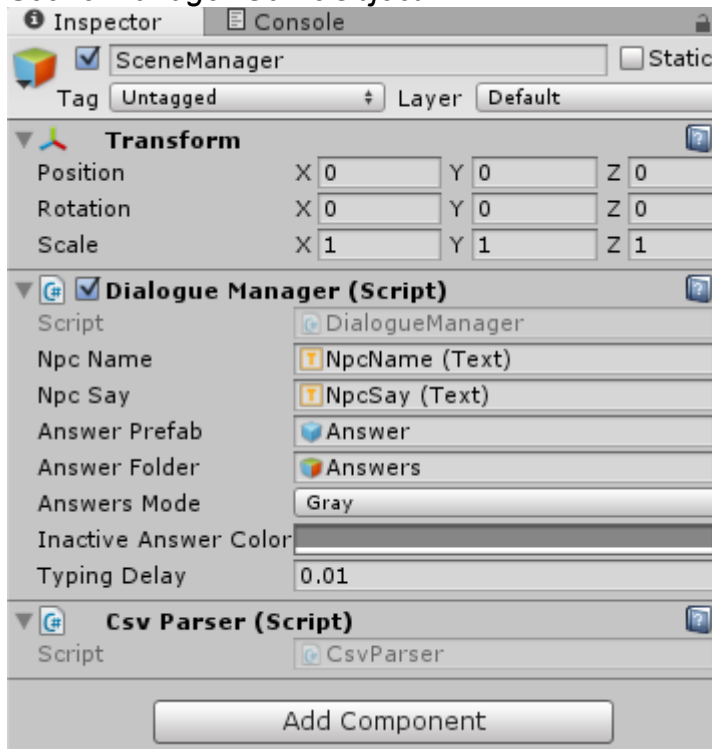
## Common operate diagram



## How to integrate it?

- 1) Create GameObject and attach DialogueManager and CsvParser scripts;
- 2) Set all link fields in DialogueManager's interface:
  - **NpcName** – text field for displaying NPC's name;
  - **NpcSay** – text field for displaying NPC's speech;
  - **AnswerPrefab** – GameObject with Text, Button and AnswerHandler scripts. It will be clone on each player's answer displaying. Simple way – take it from Asset's resources;
  - **AnswerFolder** – parent GameObect for player's answers. Using here VerticalLayaoutGroup script is good idea;
- 3) Configure remaining settings:
  - **AnswersMode** – choose display mode for inactive answers (not pass requirements): **Gray** – inactive answers will be displayed with InactiveAnswerColor, **Invisible** – inactive answers will not be displayed;
  - **InactiveAnswerColor** – set color for inactive answers (only in Gray answers mode);
  - **TypingDelay** – set delay in seconds between every letter displaying (for gradual typing effect). If delay is setted to zero - text will be displayed instantly.

## Scene manager GameObject



**Info:** *CsvParser* and *DialogueManager* are singletons and each of them has one at a time instance in project. The rest of code must only refer to static Instances (for example: *DialogueManager.Instance.EndDialogue()*).

## How to create dialogue file?

First thing – take a look at separators in CsvParser script (defined as private variables). CsvParser uses this separators for searching necessary data.

### Default separators

Spreadsheet field separator	;
Spreadsheet line separator	\n
Data page sign	__Page__
Data line sign	__Line__
Data variable sign	—
NPC's and player's text	<i>Have to contain no defined above separators</i>
Data field with numeric value	<i>Have no separators</i>

Create new document in any spreadsheet editor. Press “Save As” and choose “.csv”. Set field separator and line separator same as in CsvParser. Now you have blank CSV file for further dialogue creation.

## Pages

Dialogue file consists of pages. Page is a dialogue phase with NPC's text and player's answers (except page "Desc"). Name of each page must begin with Data page sign. There are two pages which are necessary for correct dialogue beginning, these pages must be added to CSV file:

- **Desc** – dialogue descriptor page. It may contain initial condition of dialogue (for example: NPC's name, characteristics, items). All this operations may be added in DialogueManager.StartDialogue() code;

	A	B	C	D	E
1	Page_Desc				
2	Line_NpcName	Eng	John	Ital	Gianni
3					

- **Welcome** – dialogue start page. It is the first displayed dialogue phase at the dialogue start.

3					
4	Page_Welcome				
5	Line_NpcAnswer	Eng	Do you speak English?	Ital	Parla Italiano?
6	Line_Common	Eng	Sure	Ital	Certamente
7					

## Lines

Dialogue page consists of lines. Name of each line must begin with Data line sign. Stuff lines:

- **NpcName** – NPC's name in supported languages (in page "Desc" only);

	A	B	C	D	E
1	Page_Desc				
2	Line_NpcName	Eng	John	Ital	Gianni
3					

- **NpcAnswer** – NPC's text. This is mandatory line in each dialogue phase;

4	Page_Welcome				
5	Line_NpcAnswer	Eng	Do you speak English?	Ital	Parla Italiano?
6	Line_Common	Eng	Sure	Ital	Certamente

- **Effect** – lines with effect description. This is optional line(s). Page may contain several Effect lines. All effects in page are applied when this dialogue phase occurs;

22	Page_BuyBarrel				
23	Line_NpcAnswer	Eng	Take it	Ital	Prenda
24	Line_Requirements	Req	Gold	2	Req
25	Line_Effect	Gold	-2		
26	Line_Effect	Item	Add	Barrel	

- **Requirements** – lines with the list of dialogue phase requirements. This is optional line(s). If requirements are failed – this dialogue phase can't occur (such answer can't be chosen). Requirements may be united by logical operations AND/OR.

16	Page_BuyProvision				
17	Line_NpcAnswer	Eng	Try this tasty pumpkin pie	Ital	Prova la questa gusta torta di zucca
18	Line_Requirements	Req	Gold	1	
19	Line_Effect	Gold	-1		
20	Line_Effect	Food		1	

48	Page_Example				
49	Line_NpcAnswer	Eng	Blablabla		
50	Line_Requirements	Req	_req1	Req	_req2
51	Line_Requirements	Req	_req3	Req	_req4
52					
53					
54					
55					
56					
57					
58					

**bool result = ( req1 && req2 ) || ( req3 && req4 );**

All other lines are player's answers lines that will be displayed on appropriate dialogue phase. Answer line must contain text for displaying. When answer is clicked on it calls dialogue phase with same page's name. If there are no active answers on new page, current dialogue phase will be displayed again (however effects from new page occur).

8	Page_Common				
9	Line_NpcAnswer	Eng	Can I help you?	Ital	Posso aiutarla?
10	Line_BuyProvision	Eng	Bring me some food. (1 gold coin)	Ital	Mi porti un po' di cibo. (1 moneta d'oro )
11	Line_BuyBarrel	Eng	I want to buy a barrel of beer. (2 gold coins)	Ital	Voglio comprare un barile di birra
12	Line_StayByNight	Eng	I need a room for a night	Ital	Ho bisogno di una stanza per una notte
13	Line_BorrowGold	Eng	I need money. Can you lend me some?	Ital	Ho bisogno di soldi. Mi puoi prestarmi?
14	Line_Leave	Eng	I have to go	Ital	Devo andare
15					

*Info: pay attention to correct dialogue compilation. If there are no active answers on new page and on current page – dialogue will be forcibly ended.*

## Variables

Variables are data fields which can be used for different purposes (to describe the conditions of effects or requirements). The number of variables in sequence are not limited. Name of each variable must begin with Data variable sign.

Stuff variables:

- **LanguageSign** – specified name (for example: **Eng**, **Ital**) that is used to determine start of answer's text field. Switching this variable changes dialogue language. Supported languages are defined in `DialogManager.Languages` enum and may be switched by `DialogManager.Instance.SetLanguage();`

3					
4	Page_Welcome				
5	Line_NpcAnswer	Eng	Do you speak English?	Ital	Parla Italiano?
6	Line_Common	Eng	Sure	Ital	Certamente
7					

- **Req** – name that indicates start of new requirement's sequence in **Requirements** line;

21					
22	Page_BuyBarrel				
23	Line_NpcAnswer	Eng	Take it	Ital	Prenda
24	Line_Requirements	Req	Gold	2	Req
25	Line_Effect	Gold	-2		
26	Line_Effect	Item	Add	Barrel	
27					

*Info: all stuff names are defined in private variables of `DialogueManager` and can be configured. Stuff names can not be used in names of player's answers.*

## How to control dialogue?

After creation CSV dialogue file you can control dialogue in your game. There are several useful methods in code:

- **public DialogueManager.Instance.StartDialogue()** - call it from any part of code and place TextAsset dialogue file as parameter;
- **public DialogueManager.Instance.SetLanguage()** - choose dialogue language from supported;
- **public DialogueManager.Instance.EndDialogue()** - you can manually stop dialog at any time;
- **private DialogueManager.CheckAnswerRequirements()** - the place where you can add your own requirements parsing. For parsing requirements line use methods from CsvParser.Instance (for example: GetTextValue(), GetNumValue()). If requirements fail code must return false;
- **private DialogueManager.ApplyEffects()** - the place for description of possible effects. Use CsvParser.Instance for parsing.