GameNode

• Store GameNode id • Store Dialogue • Store Choice(s) • Store Pointer(s) associated to the Choice(s)	• Game

Game

• Store id of the Game • Store name of the Game • Store author of the Game • Store a "Tree" of GameNodes	• GameNode • GameManager

GameManager

• Get an array of public Games through GameLoadSaver • Get an array of private Games associated with a specific User through GameLoadSavor • Store an array of buffered Games • Create a new Game and store it into the file using GameLoadsaver • Edit a Game in the array of private Games and store it into the file using GameLoadsaver	• Game • GameCreator • GameEditor • GameLoadSavor • GamePlay • UserNavigator

GameCreator

• Retrive an array of Templates using TemplateManager • create game nodes and make it a tree for the whole Game(stick to the Template chosen) • store it in an array of buffer	• GameManager • TemplateManager

GameEditor

• edit a private Game	• GameManager

GamePlay

• Provide a UI for the User, • from the very first GameNode, display dislogue and choices, asking User for input(of choice) • redirect to the corresponding next GameNode	• GameManager

GameLoadSaver

• Read from file for public games • Save public/private games into the file	• GameManager

Template

• Store the id of the Template • Store the description of the Template(eg: "maximum of depth of 8 allowed") • Store the maximum depth allowed	• TemplateManager

TemplateManager

• Call TemplateLoadSaver to retrive an array of Templates • Save the updated array into the file calling TemplateLoadSaver	• AdminUserNavigator • GameCreator • TemplateEditor • TemplateLoadSaver

TemplateLoadSaver	
• Read and Write the file that store the Templates	• TemplateManager

TemplateEditor	
• Display the array of Templates(get from TemplateManager) to the User • Let User pick the one to edit • Save it back to the array in the TemplateManager	• TemplateManager • AdminUserNavigator