

First Person Role-playing Game

- Source Code -

MANUAL

email support: nenadradojcic@yahoo.com

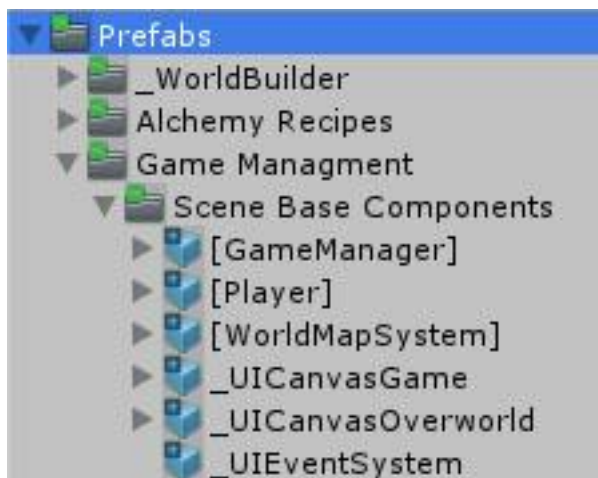
Table of Contents

STARTUP.....	Error! Bookmark not defined.
PLAYER	Error! Bookmark not defined.
NPCs	Error! Bookmark not defined.
ITEMS	7

First Person Role-playing Game Source Code is a complete source code done for game “The Fallen Chronicler”. Every aspect from scripts to graphics come bundled together in this great project. Project has been written in such a way that every script is easily readable and understandable.

SETTING UP YOUR SCENE

One basic thing to do when making new scene is that you need to place some certain prefabs. Those are prefabs located in ***Prefabs/Game Management/Scene Base Components*** and are



[Game Manager] This one holds scripts for weather, date, music player and other game management things.

[Player] This is our player character that holds scripts for movement, camera, our leveling and skill system, weapon and shield blocking systems.

[WorldMapSystem] This is camera object that we use for our overworld map view.

_UICanvasGame This game object holds every User Interface component of our game.

_UICanvasOverworld This game object is same as one above, expect it holds location icons and names on overworld map view.

_UIEventSystem A Unity5 UI Event System needed for UI input.

When all of these components are placed in our system we can start tinkering them and making adjustments for our game. Every variable is self-explanatory, and those that are not could be just moused over for a more detailed explanation.

PLAYER

Player is core feature of the system. In an example demo scene, you can see how we made a player skills and attributes system based on chosen race. Every race gives a boost to a certain skill.

Attributes are divided into Body, Agility and Mind. Body determines our power and might, Agility determines swiftness and balance while Mind determines instincts and cleverness. Every attribute governs three of nine skills our player possesses. Body governs Melee, Block and Heavy Armor. Agility governs Marksmen, Acrobatics and Light Armor. Mind governs Speechcraft, Alchemy and Focus. Our skills are predetermined by the choice of our race, but can and will be increased while doing actions that are tied to them. Say you swing your sword on every living thing you meet in your adventures, you will raise your Melee skill. When 3 skills have been raised you gain one level up and 3 attribute points to raise your attributes.

NPC

You want to fill your world with npcs? Just head over to **Prefabs/Npcs** and drag **NPCPrefab** to your scene. Characters in our game are custom made sprites that have a billboard script on them so they who are 2D always follow players camera in 3D world. Only script that you would need to modify is a NPC.cs script (first one from top to bottom). Here you will find many different setting that will make every of your npcs unique.

Loot bag - Here you place all items you wish this npc to drop after death. If this is left blank, in game manager game object there is a script that will randomly assign some items here for you.

Bones Prtcl - A particle effect that will play on npcs death. By default a blood particle is set.

NPC Name - Here we write name that will show up in game when crosshair is pointed toward this npc.

NPC Faction - Here we choose faction for our npc. Faction determines how this npc will react to player and npcs belonging to their or other factions. Commoners, Guards and Merchant are allied against Bandits, Monsters and Animals, who will attack one another.

NPC Disposition - What is disposition of this npc toward player. NPC Faction will always overlap this if they are not equal. Neutral will immediately attack you if you attack him, Friendly will attack you if you are persistent and attack him multiple times in a row, Ally will never attack you, while Hostile will attack always on sight.

NPC Difficulty - Here we set how difficult would be to fight this npc.

► Loot Bag
Bones Prtcl Blood01FX

Anim Base base (Animator)
Anim Hair hair (Animator)
Anim Facial Hair facial hair (Animator)
Anim Helm Hat helmet (Animator)
Anim Outfit outfit (Animator)
Anim Weapon weapon (Animator)

Npc Name

Npc Faction None
Npc Disposition Neutral
Npc Difficulty Very Easy

Race Elinian
Gender Male
Hair Color 
Hair None
Facial Hair None
Helmet Hat None
Outfit None
Weapon None
Shield None

Set Looks ☐
Merchant Topic None (NPC Topic)
Merchant Counter 0

Will Npc Respawn ☐
Respawn After Days 0

Race - We choose race of our npc. (Races listed are races from The Fallen Chronicler cancelled game)

Gender - This one talks for itself.

Hair Color - This one talks for itself.

Hair - Here we choose style of npcs hair.

Facial Hair - Here we choose style of npcs facial hair.

Helmet Hat - Here we set our npcs headwear. (should he wear any)

Outfit - Here we set our npcs outfit. (should he wear any)

Weapon - Here we set our npcs weapon. (should he wear any)

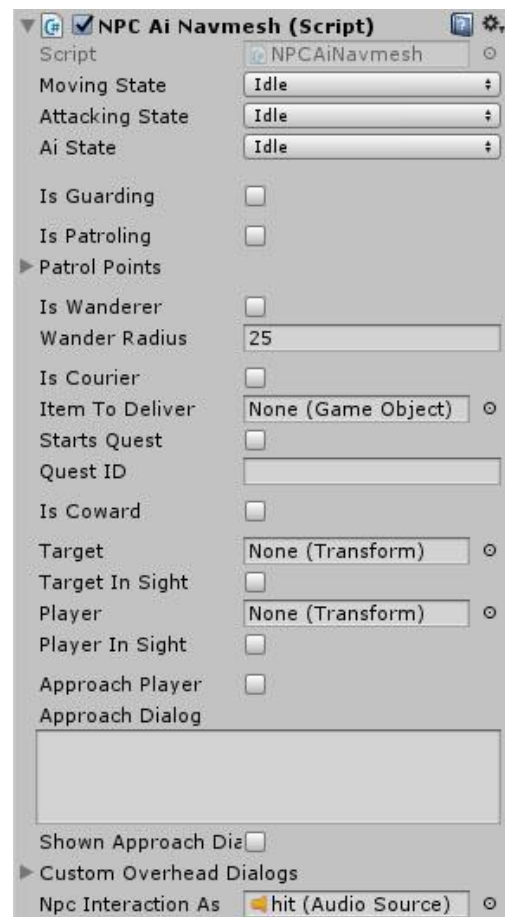
Shield - Here we set our npcs Shield. (should he wear any)

Merchant Topic - If we want our npc to be able to sell/buy from our player here we need to set a topic prefab (that we place as a child of npc in *NPCPrefab/Topics*). More on this in Custom Topics section.

Merchant Counter - This is a counter for adding money to merchants inventory. Whenever this value of seconds that we put in this field has passed 40 gold coin pieces will be added to his store.

Will Npc Respawn - Will this npc respawn....

Respawn After Days -....if he will, after how many days since his death.



Next thing in making our npc is choosing his AI routine. Our AI is a simple system that will give a certain point of life into our characters. Those are configured in script named NPCAiNavmesh.cs. Here we can choose *Is Guarding* (if selected our

npc will store his position on scene start and will guard it. If he moves it, he will always return to this points) , ***Is Patrolling*** (in Patrol Points we set points in game world that our npc will walk to in order as they are placed in the list) , ***Is Courier*** (if this is selected we need to assign Item To Deliver. It means thins npc will approach player saying some line that we write in Approach Dialog down below. If we want this action to start some quest we will tick Starts Quest and will add quest ID in Quest ID Field - more on quests in Quests section) , ***Is Coward*** (if we fight this npc and his health drops to some level bellow half of his health points, he will run away from us), ***Approach Player*** (this is same as Is Courier but will not deliver any item to player and cannot start any quest. This we can use when we want our npc to begin dialog with player) and ***Custom Overhead Dialog*** (our npcs have a overhead dialog text game object as a child. They will often comment some randomly assigned lines depending on weather, time of day, players race and etc. When they do so a text will appear over their head representing their words. Here we can place some custom lines that they will use to comment while near player.

If you want your NPC to have a house where at night he will go to sleep while continuing his A.I. routine over day, you need to assign him in the DoorManager.cs script on the door that is attached to the house model object that you want to use as his house.

That would be everything you need to know when you want to add new NPC to your game.

ITEMS

Our project comes with already many added items. Here we will show you how to add your very own items to the game world.

Firstly you want to go to **Prefabs/Items** and drop **ItemPrefab** game object into your scene. Here you only need to configure Item.cs script.

Owner - Does this item belong to some npc. If it does, here we will put that npc.

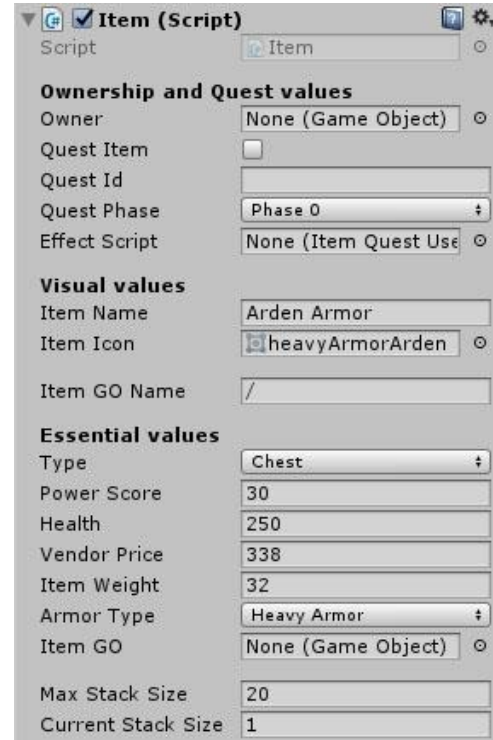
Quest Item - Determine if this item is quest item. If it is we cannot destroy it or drop it from our inventory once we pick it up. Other variables concerning quest are configured in Quest.cs script.

Item Name - What name will be showed up in game when crosshair is over this item and when item is in inventory.

Item Icon - This will be the icon item will display while in inventory. It must be sprite.

Item GO Name - This is more advanced. We use this only if we are adding weapon or shield or misc. item that player can wear in hands and is usable (like torch etc.) Go to your player game object and find **[Player]/FP_CameraGO/EquipmentGO** and depending on item type you want to add expand either **ShieldsGO** or **WeaponsMeleeGo** or **WeaponsRangedGo**. There you will find many object that represent item equipped in players hand. Duplicate one of them and change sprite and name. Keep in mind that if you are adding Shield you need to name your item **ShieldSomething** or if you are adding mace **MaceSomething** and so on. This is needed for our attack animation. Swords are fastest when attacking followed by axes, while maces are slowest. When you do that, you then place name of that object in this field (**Item GO Name**).

Type - What type is this armor? We choose if it is armor, potion etc.



The screenshot shows the Unity Inspector for the 'Item (Script)' component. The 'Ownership and Quest values' section includes fields for Owner (None (Game Object)), Quest Item (checkbox), Quest Id, Quest Phase (Phase 0), and Effect Script (None (Item Quest Use)). The 'Visual values' section includes Item Name (Arden Armor), Item Icon (heavyArmorArden), and Item GO Name (/). The 'Essential values' section includes Type (Chest), Power Score (30), Health (250), Vendor Price (338), Item Weight (32), Armor Type (Heavy Armor), Item GO (None (Game Object)), Max Stack Size (20), and Current Stack Size (1).

Power Score - If this item is armor this will be defense bonus it bring, if it is weapon it will be damage it deals etc.

Health - Base health of this item. Health reduces with usage.

Vendor Price - Sets the price of this item when buying/selling.

Item Weight - Sets weight of this item when placed in inventory.

Armor Type - Sets armor type (light or heavy) if item is armor or shield.

Max Stack Size- Sets what is the item max stack size before it will create another slot in inventory.

After configuring all of that, you would like to put your own sprite for *sprite* child game object of item for representing the item in the game world. That would be it about item creation.

QUESTS

Quests are game object who holds Quest.cs script on them. It is a script that hold different variables that will make up an objective that when bundled makes a unique adventuring experience.

To create your first quest create empty game object and assign script Quest.cs to it. Now configure it to your liking.

Quest ID - Quest ID should be the name of game object that holds the script.

Quest Name - Quest Name that will show up in games Quest Log.

Quest Is Triggered By - Determines what will trigger the quest. There are multiple options here and every option needs different configuration. *Topic Trigger* will ask for npc game object together with the name of topic that will show up in dialog window and topic text that will show up when topic active. This topic will begin the quest. Bundled together with **Approach Player** (look at NPC section) this can give awesome results. We then have *Area Trigger* that will start quest when player enters selected trigger collider. *Other Quest* means that other quest upon completion will activate this quest. For *Courier Trigger* quest startup look in NPC section for more details.

Quest Requirements - Here we can set that to begin this quest other quest had to be completed, or that player should reach certain level, or both.

Objective 1, 2 ,3, 4 - When we open Objective 1, 2 and 3 we will find more variables needed for configuration. Every of this objectives can be set to those actions needed from player: *Kill Npc*, *Find Npc*, *Talk to Npc*, *Finish Quest*, *Continue Quest*, *Get Item* and *Deliver Item*. Objective 4 can be and should be set only to *Finish Quest* or *Continue Quest*.

Take note that if you want some objects to enable or disable, you will need to go into debug mode and assing them to their objects lists!

QUEST BASIC INFO

Quest ID: Quest01

Quest Name: The Quest!

Quest Is Triggred By: Topic Trigger

Quest NPC: NPCPrefabTheQuest

Topic Name: The Quest

Topic Text: I want you to go north and kill bauk residing near those huts.

QUEST REQUIREMENTS

Requirements To Start The Quest: None

No requirements needed to start the quest.

QUEST OBJECTIVES

- Objective - 1
- Objective - 2
- Objective - 3
- Objective - 4

Find Npc means that we need to find certain npc set in the field *NpcToFind* and *DistanceTrigger* also needs to be set. (this is the distance from where you will notice the npc).

Kill Npc means that we need to kill npc set in the field *NpcToKill*.

Talk To Npc means that we need to talk to npc set in the field *NpcToTalkTo*. *TopicName* is the name of the topic that will pop up in dialog when we talk to this npc while *TopicText* is the dialog the npc will say when topic is activated. This will activate objective.

FinishQuest sets this quest as finished and configured prize is given to player. Prize can be gold or item or both. Together with valor points that represents players reputation among npcs.

ContinueQuest we select this when we have no more objectives to use in our quest script but the quest is not over. It is used to create big quests. In the *QuestToStart* field we need to enter *QuestID* of the quest that is the continuation of this one.

GetItem and *DeliverItem* are objectives telling us to retrieve/deliver some item. If *GetItem* in field *ItemToPickup* we need to place some item from the game world. If *DeliverItem* is selected we need to select item prefab that will be used in quest, usually from game world, that we need to deliver to some npc. We will create new topic that will give npc the item.

Quest Log Description - This is a description for every object that will be written in our quest log so that we can more easily follow our progress.

Inform Player of Quest State - Every Quest Objective have this option where we choose if we want a window to appear in middle of screen where a quest name, date when quest was started and current objective description will be shown, or if we want a simple “*Quest Updated/Completed*” message to be shown in the top right corner of the window.

With this knowledge and some practice you will be creating some awesome quests.

OTHER

This source code holds many more things that will make your game awesome and they are so numerous that even I cannot remember what everything there is.

We have sorted folders in a best possible way so you can find your way through our source code while using it for your own project.