

Prom Week meets Skyrim

Developing a Social Agent Architecture in a commercial game

Manuel Guimaraes
INESC-ID
Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal

manuel.m.guimaraes@tecnico.ulisboa.pt

Pedro Santos
INESC-ID
Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal

pedro.santos@tecnico.ulisboa.pt

Arnav Jhala
Department of Computer
Science
North Carolina State
University

Raleigh, North Carolina, USA
ahjhala@ncsu.edu

ABSTRACT

We present and describe CIF-CK — a social agent architecture that models reasoning about persistent social interactions to improve narrative engagement and play experience for human interactors. The architecture is inspired by McCoy et al’s *Comme il-Faut (CiF)* architecture that represented rich social interactions between agents that included feelings, social and relationship contexts, and longer term mood. The key contribution of this work is in adapting the richness of social interactions from CiF to a first-person interaction experience and a released distribution of its implementation on the Skyrim game engine. The released modification has been successful in the player community for the popular game.

Keywords

non-playable character, social interaction, social architecture model

1. INTRODUCTION

Modern video game have reached a new level of graphic fidelity. Furthermore with the rise of popularity of new technologies such as Virtual Reality and Augmented Reality, gamers are more immersed than ever within virtual worlds and virtual characters. With the level of real life likeness of the game world increasing so does the player expectation of real life like interaction [8]. This expectation is extended to the characters that compose the environment, typically called Non Playable Characters (NPCs), because their behaviour is defined by the programmer. Both computer controlled characters and player controlled avatars need to act in a believable manner so that the illusion of reality created by exquisite graphics and physics, the “player immersion”, is not broken [26].

The credibility and believability of NPCs requires characters to have basic human traits like emotions and the ability to make decisions on their own [9]. One of the most important human traits is our social ability and awareness. People’s thoughts, feelings, and behaviours are influenced by

the presence, even if just imagined, of other human beings. This makes rich agent modeling important for human interaction within these environments. One important aspect for agent models is affinity with the player’s social concerns and behaviours [7].

There has been significant advances in the agents research communities on such rich agent models. With the amount of human interaction in AAA games, these provide a rich sandbox to deploy, gather data on, and validate agent models and programs. Most modern day AAA video games are heavily dependent on a high number of NPCs and rely on the Player’s interaction with them to advance the game’s narrative. Unfortunately, most of the NPCs do not exhibit deep social reasoning for their player interactions and most of the times are simply frozen in time, repeating the same action, if any, over and over again [1]. Even in games that boast of great AI, the characters are given the ability to play the role necessary in that game, but no other skills or personality, quickly becoming forgetful background characters (like extras in a film).

Modern social architectures/models, originating from academic research groups, have the potential to transform game NPC interaction to open up rich narrative design spaces for players to explore. These models allow the system to automatically manage and keep up with the complexity of social interactions, reducing the number of experiences that need to be explicitly authored [14]. Reasoning about the social context in terms of relationship goals and desires, social status, and emotional changes is central to believable behaviour (cite: goffman).

Academic research on AI in games and commercial game AI development efforts are rather disjoint and there is a wide divergence in methods and results. This is the result of academic research and commercial game development trying to solve different problems: academic researchers want general solutions to generic and preferably deep problems, whereas commercial developers want something that works well enough and looks good to the player in a particular context[27]. For example, as a result of researching Interactive Storytelling(IS), the video game “Façade” [13] was created and quickly became regarded as the future of interactive storytelling (and still is widely cited in IS research) [22]. But despite its success and popularity as a game, there is still a gap between the technology and development tools that could make it accessible to a wider population of developers and users of such interactive experiences.

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AA-MAS 2017)*, S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.

Copyright © 2017, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

This project’s goal was to develop and implement a social architecture model, inspired in academic research, in a modern and commercially successful video game and investigate its impact on player experience. We chose to implement the social architecture in the video game “The Elder Scrolls V:Skyrim”[24] because of its popularity and high “mod-ability”.

2. RELATED WORK

Several research groups have addressed the problem of socially intelligent agents and created various different social simulation models, however, different architectures fit different games, and, before deciding which is more appropriate for this project we need to take a deeper look at the game itself.

2.1 A Single-player RPG Example: The Elder Scrolls V: Skyrim

The Elder Scrolls V: Skyrim is an action role-playing open world video game developed by Bethesda Game Studios and published by Bethesda Softworks. It is the fifth instalment in The Elder Scrolls series, following The Elder Scrolls IV: Oblivion. Skyrim was released worldwide on November 11, 2011, and received critical acclaim. The game shipped over seven million copies to retailers within the first week of its release, and sold over 20 million copies across all three platforms (PC, PS3 and Xbox 360) ¹. Over the course of the game, the player completes quests and develops the character by improving skills, its main plot is built upon a series of main quests but a lot of players opt not to do play them and focus on exploring the world and completing side quests only.

The game was developed using the Creation Engine, rebuilt specifically for the game. Bethesda decided to open the game to Mods and provided software to do so by releasing the **Creation Kit** tool. Today, Skyrim, a game that was released 4 years ago, has about 30 000 different daily concurrent players, and is the 11th most played game on Steam, a popular Internet-based game distributing platform. Todd Howard, Director and Executive Producer at Bethesda Game Studios, stated in an interview with Rock, Paper, Shotgun that “Skyrim did better than we’ve ever done on PC by a large, large number. And that’s where the mods are. That feeds the game for a long time.”

2.2 Creation Kit

Creation Kit is the name of Skyrim’s modding kit that was made available by Bethesda to facilitate the development of mods by the community. It was released on February 7th, 2012 and it includes tools needed to create custom content, using the game’s files.

Creation Kit is a very powerful tool as it is the same tool that Bethesda itself used to create Skyrim. Almost everything the developers used is available to the “modder”. Users can use those resources or add new ones, such as creating new items, effects, actors, locations, quests, etc. Despite its power, nothing the user can do in Creation Kit can corrupt the main game, at least, not permanently. The master data file can’t be changed, instead, created mods are added onto the top and can be switched on or off individually [4].

¹<http://www.statisticbrain.com/skyrim-the-elder-scrolls-v-statistics/>

Plugins, or “.esp” files, are smaller collections of data which can be loaded “on top” of master files. These plugins may modify or reference data contained within a master file, or they may introduce entirely new data. Multiple plugins may be loaded by the game or editor. When working in the Creation Kit, only one plugin may be considered the “active file”, meaning any changes will be saved to that plugin when the user saves.

The Elder Scrolls V: Skyrim is built with thousands of objects and thousands of objects and they are all available to the modder in Creation Kit. These objects are presented in the Object Window that separates each entity in their proper folder and hierarchy. Objects can belong to: Actors, Audio, Character, Items, Magic, Miscellaneous, Special Effects, World Data and World Objects. For example if I want to find a particular NPC called “Sabjorn” I will find it in the Actor folder.

2.2.1 AI Packages

In order to give Non-Playable Characters its behaviour and not have to manage it constantly, the Creation Engine gives every Actor a list of AI Packages that the actor will execute. Packages are the main way in which one can control an Actor’s behaviour. Each Package represents a behaviour that the Actor will perform under certain conditions ².

The Package system consists of a number of components:

- All Actors have a Package Stack, an ordered list of potential behaviours.
- These stacks are composed of Packages, individual behaviours that the Actor can perform.
- Most Packages are instances of a Package Template, which provides standardized, reusable functionality for common behaviours.
- Packages can modify the behaviour inherited from their Templates in predefined ways, such as specifying data values.
- Package Templates, in turn, are composed of a structured Tree of Procedures, the atomic actions that make up the behaviour.
- Periodically, the game will re-evaluate each Actor’s Package Stack. The topmost package in the stack whose conditions are satisfied will be run.

As an example, AI Packages are the reason why most NPCs return to their homes when the day is over in Skyrim. The NPC evaluates his Package Stack checking the conditions on each Package from the top down. If the conditions are satisfied in the first one, for example: “NPC should go Home if it is past midnight”, then that package proceeds to carry out the proper behaviour.

2.2.2 Quests

The original Skyrim game has hundreds and hundred of quests. Quests have multiple functions besides delivering a narrative. Some exist to simply store dialogue, others to manage random events that happen in the world. There are other quests that define complex and important storylines and, in turn, create secondary quests to deal with its consequence in the world. In the Creation Kit tool, the list of quests is displayed in the Object Window under Character. The main components of a quest are:

²<http://www.creationkit.com/index.php?title=Category:Packages>

- **Quest ID**, identifies the quest and is unique to the whole game
- **Quest Stages**, associated with an integer number, represent the phases of a quest, all of the stages can have starting conditions and ending conditions, for every quest there is always a starting stage and an ending stage, in some cases, they might be the same.
- **Quest Aliases** are the references to the objects that are used during the quest, they might be characters, items and even other quests. Quests need all of its Alias filled to start, they might be manually filled by the modder or the modder can set the system to do it automatically..
- **Quest Scripts** are used to apply the effects of the quest in the Game World, often used to move from stage to stage and to execute scenes.
- **Quest Scenes**, the physical performance of a quest. Scenes use temporary AI packages that override the others. A scene can be, for example, making a farmer deliver a letter to the Player.

Quest Aliases are names or tags assigned to actors, objects, and locations used by the quest. This allows various data elements (script, packages, dialogue) to be tagged to the alias rather than to a specific object in the world, allowing quests to specify their aliases at runtime instead of being predefined³. This particular ability makes it possible to reuse each Quest with different participants or values.

An example of how the Quest mechanism works is available in the official Wiki⁴. The main Creation Kit tutorial is about creating a mod where the Player character receives a letter whenever he sits on a specific bench in a specific bar. There are multiple steps required to create any quest that can also be applied to this particular one:

- Creating all the new objects the Player will interact, in this case the letter and the NPC.
- Create the Quest and define all the stages, each stage represents the state of completion of the quest, has it started? Has the player sit in the bench? Has the Player gotten the letter? Should the quest finish?
- Add Aliases to the Quest, what objects part of the quest, in this particular case there are three aliases: the NPC that will deliver the letter, the particular bench, the letter itself and the Player.
- Creating the Scene, this particular scene is composed by the NPC going to where the player is and giving him the letter along with some dialogue.
- Inform the Game Engine when to start this quest, in this case the quest should start when the Player controlled character enters the bar. This is usually the last step.

2.3 Social Architectures

There are many systems, developed by Academic Groups, whose goal is to model interactions between characters. In our search for a model that we could implement in a commercial game we looked at some of those social simulation architectures.

2.3.1 PsychSim

Psychsim is an agent based system to simulate social interactions. A unique aspect of the PsychSim design is that agents have fully specified models of others. These models are recursive and determine how the beliefs of an agent are affected by the agents around him [10]. PsychSim allows a user to quickly construct a social scenario where a diverse set of entities, groups or individuals, interact and communicate. Each entity has its own preferences, relationships, such as friendship, hostility, authority, with other entities. The simulation tool generates the behaviour for these entities and provides explanations of the result in terms of each entity's preferences and beliefs [20].

2.3.2 FAtiMA

FAtiMA (Fearnot Affective Mind Architecture) is an agent architecture with planning capabilities that uses emotions and personality to influence the agent's behaviour that is authorable in XML. In recent years, the architecture was used in several different projects, such as FearNot! [19], ORIENT [2], a process Model of Empathy[21] and RAGE [11], and by different research institutions [5]. For this reason, a modular version of the architecture was created, where functionalities and processes are divided into independent components. Based on cognitive appraisal theory of emotions [23], the FAtiMA architecture offers a generic appraisal framework where a set of different basic emotions and the coping behaviours can be generated according to a set of goals, preferences and action tendencies predefined for a virtual agent [3].

2.3.3 Comme il Faut

Comme il Faut (CiF), that roughly translates to "as it should be", is an artificial intelligence system and authoring strategy for creating game-based interactive stories about relationships and social interactions between characters [16]. In CiF the characters use many attributes of the current social state, including the history of prior interactions, to decide how to engage in these multi-character social exchanges. The goal of this architecture is provide a rich social environment for the characters to interact.

Instead of compressing all domain knowledge in nodes or states, like many AI techniques do, such as Behaviour Trees and Hierarchical Task Networks, CiF chooses characters' behaviours based on rules in a large rulebase that depict normal social behaviour in a particular story world [17]. CiF doesn't create a static, or even branching, series of events, but rather the logic of a social world, a set of characters, and a series of scenario goals. Because CiF is driven by simulating social interaction, goals may be met in unplanned ways, but are always consistent with the designed story-world.

2.3.4 Discussion

It is fundamental that the architecture used can be adapted to an already finished video game (as in, no access to the source code or the internal structure of the game). Instead of building completely new agents with new AI we are adding to the existing game world characters additional behaviours, without overriding the original ones. Of the architectures discussed previously, the Comme il Faut architecture does not require any major changes to an already built game world and its structures. It is relatively "light" (when compared to the others), easy to comprehend and seemingly simple to implement. Additionally the structures

³http://www.creationkit.com/index.php?title=Quest_Alias_Tab

⁴<http://www.creationkit.com>

present in CiF, namely the notion of Social Exchange, can be mapped to the elements behind Skyrim and the Creation Engine. As a result the architecture we've chosen to test our thesis, and implement it in Skyrim, is the *Comme il Faut* social architecture.

The *Comme il Faut* architecture been previously described in depth[14], here, we present a brief descriptions of its structure and components.

2.4 The Comme il Faut Architecture

The work on *Comme il Faut* (CiF) started with the goal of generalizing those multi-character exchanges into reusable units, this allows the creation of multi-character social exchanges more generally and then targeted to specific characters in specific situations. The CiF architecture can be described in four essential components:

- **Social State**
- **Characters**
- **Social Exchanges**
- **Trigger Rules**

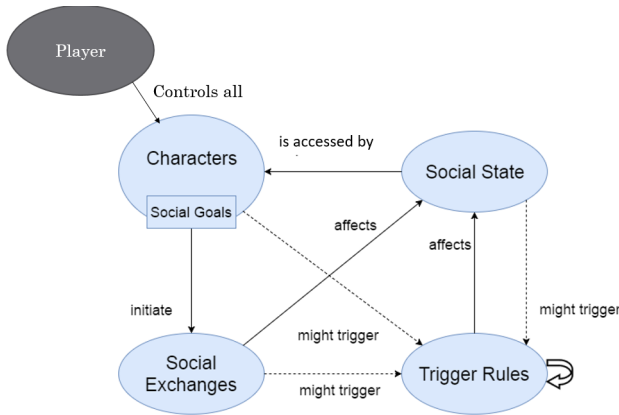


Figure 1: Simple representation of CiF components and the relation between them in Prom Week

Figure 1 describes how each component relates to the other. In short, NPCs try to perceive the Social State around them and try to change it to accomplish their Social Goals. To change the Social State they make Social Exchanges such as, Flirting with another NPC. These Social Moves might be successful or unsuccessful, in both cases they have consequences once they have been executed. According to its success or the lack thereof the effects of each Social Move change the Social State and we go back to beginning of the cycle. Furthermore, at any moment in the cycle any stage can have unforeseen consequences. These unforeseen consequences are handled by the Trigger Rules that if they are “fired” also affect and change the Social State. The relation between these components can be seen in Figure 1.

2.4.1 Prom Week

The *Comme il Faut* architecture was used in the video game “Prom Week”. Prom Week is a game that goes through the week leading up to a high school prom. The Player can control all the characters and decide what social exchange should they perform and with who. Prom Week simulates the results of the interaction chosen and shows its effects

to the Player. The consequences of each character’s action can influence its “Coolness”, its “Friendliness” and its “Romance”. CiF manages which social exchanges are available and how each changes the social stat [25]. Figure ?? describes all of the CiF elements that were implemented in Prom Week.

3. SOLUTION ARCHITECTURE: CIF-CK

Our adaptation of the *Comme il Faut* architecture to the Creation Kit is called the **CIF-CK** (*Comme il Faut* Creation Kit) architecture. When describing the *Comme il Faut* architecture we mentioned four essential components to this model: the Social State, the Characters, the Social Exchanges and finally the Trigger Rules. Lets now take a look at how each essential component was adapted to the Creation Kit Engine and some innovations made in CIF-CK to CiF.

3.1 Social Exchanges

In most Role Playing Games everything revolves around quests. Skyrim follows this tradition, from making an NPC talk to the player or to sit on a specific chair, from complex main narrative storylines to simple “collect some plants” missions, at least one quest needs to be executed. Quests are used to store variables, dialogue, actors, performances and even locations.

The primary knowledge representation element in CiF is the Social Exchange, a collection of patterns of (primarily dialogue) interaction where the exact performance and social outcome varies based on the personality-specific attributes of the characters involved and the current social state [12].

The similarity between Quests, in Creation Kit, and Social Exchanges, in CiF, allow us to adapt one to the other. We can use Quests the same way Social Exchanges are used in CiF, with some adaptations. We demonstrate this transition in Figure 1.

For example to execute the Social Exchange “Flirt” initiated by the NPC “Sarah”, the system starts the quest ‘SocialMoveFlirt’ and fills the Initiator Alias with the Sarah Actor reference and the Target Alias with Sarah’s Target reference, let’s say in this case, “John” NPC. After being filled the quest can start its scene. The scene in, this case, is composed by: Sarah walking to John, delivering a line of dialogue complimenting John and then walking to her previous position. After a scene ends the quest moves to its ending phase. The ending phase of a quest is essentially applying its consequences to the Social State. The Social State consequences include improving the NPC’s relationship and perspective of each other, adding or removing Status and with those changes some Trigger Rules might be fired. These would bring even more changes, even if small ones, to the Social State.

3.2 Characters

As we mentioned, the likeness between quests in Skyrim and Social Exchanges in CiF allows us to build a bridge between the two different architectures, however, it is not enough. In order to fully capture CiF’s architecture we need to improve upon the Characters created in Skyrim (and in most RPGs) by adding additional behavior. In order to capture that additional behavior we have implemented certain variables described in Table 2.

CiF	Function	Creation Kit
Name	Identify the Social Exchange	Quest ID
Intent	Description of the pretended change in the Social State	Can be declared in the final stage of a Quest making sure they are only executed if the Quest is completed
Preconditions	Conditions that must be true for the social exchange to happen	Quest Start Conditions usually declared in the initial stage of a Quest
Initiator Influence Rules	Determine the desire for a character to initiate this specific social exchange	Set of Rules computed before the Social Exchange starts
Responder Influence Rules	How the responder reacts to the Social Exchange initiated	Set of Rules computed after the Social Exchange starts
Effects	Social State Consequences if the social move is successful or unsuccessful	Creation Kit provides a Success and Failures stage for each quest where the effects of the Social Exchange can be implemented
Instantiations	Performance of each Social Exchange	Each Quest can have Scenes where they can perform the Social Exchange through dialogue, actions, etc...

Table 1: Social Exchanges: From CiF to Creation Kit.

3.3 Social State

In CiF, the social state of the world is captured by four different representations: Social networks, Relationships, the Cultural Knowledge Base, and the Social Facts Knowledge Base [12].

Because of the huge dimension of these types of games and in order to not take a huge load on memory of the players computers, Bethesda made the Creation Engine to render and load only what is in the surrounding area of the player controlled character. If the player is in city A the engine does not compute city B because it is a waste of resources. As a consequence any script associated with an NPC that is in a different place as the player will not be processed. This fact will influence our implementation as NPCs will only be considered and will only consider those in the same place as the Player.

The Creation Engine does not provide an easy way to

CiF	Function	Creation Kit
Name	Unique Identifier	Unique identifier generated by the Engine
Gender	Describe the Character's Gender	Binary Variable in all Characters
Traits	Permanent Traits	List of variables that affect social exchanges
Status	Temporary Traits	List of variables that affect social exchanges
Prospective Memory	Set of desires of social exchanges with specific goals	Set of quests with specific actors (Targets)

Table 2: How Characters in CiF translate into the Creation Kit Engine

store or access complex script data as the Papyrus scripting language is quite limited. However the Creation Engine does create an entity that is in every scene: the Player entity itself. Skyrim uses this entity to store any piece of static data it requires, as the Engine allows it to be accessed by anyone at any time. Using a simple line of code scripts can access all information stored in the Player entity:

```
Game.GetPlayer()
```

In our project this entity is used to store all the static data the scripts need, specifically some of the Social State components. In Figure 3 the implementation of the Social State of CiF in Creation Kit is described.

CiF	Function	Creation Kit
Social Networks	Private feelings characters have for each other	Unique and different for each Character
Relationships	Publicly recognised relationships	Public information should be stored in a static reference and accessible to all
Cultural Knowledge Base	Defines what each NPC likes	Unique and different for each Character
Social Facts Knowledge Base	Social History of the World	Public information should be stored in a static reference and accessible to all

Table 3: How the Social State in CiF will be implemented in Creation Kit.

3.4 Trigger Rules

In Prom Week, Trigger Rules can be “fired” at any point in the game and have cascading effects in the Social State [12]. In order to be more efficient and avoid stressing the Game Engine, CIF-CK only verifies the trigger rules when a Social Exchange ends. In our implementation only quests directly influence the social state, as such, CIF-CK runs all trigger rules when a quest ends, either by failing or succeeding, making sure if one of them triggers we can apply it and its consequences to the social state right afterwards.

3.5 Beliefs and Social Networks

In CiF, Social Networks are scalar, non-reciprocal and private feelings from one character toward another. In Prom Week there are three networks are: Buddy, Romance and Cool [15]. In our implementation there are two networks: Attraction (Romance) and Friendship (Buddy). These networks model the relation of social attraction and friendship, first studied by Moreno [18], which reflects the affective ties that one person establishes with the others.

In CIF-CK we added a new feature to the Comme il Faut architecture. We considered that the Social State is something Characters perceive, however, it might not be the actual reality [6]. For example, Sarah might think John likes her, despite the fact that John actually hates her. Because Sarah believes John likes her, she will act accordingly. This falls under the Social Networks category, specifically in the Private Feelings each characters has for each other segment.

Figure ?? gives us an overview of our architecture and where each of the CiF’s elements are and it also serves as a parallel to Figure ??.

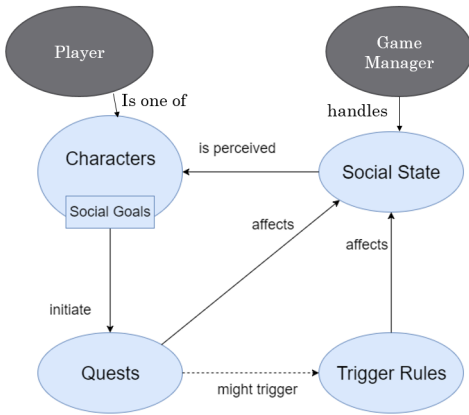


Figure 2: Simple representation of CiF-CK components and the relation between them

4. IMPLEMENTATION

We implemented the CIF-CK architecture in a mod for Skyrim, in order to test its success and impact on the original game. In the resulting mod we created 12 different social moves described in Table 4. Each Social Move, along with its consequences, is based on similar Social Exchanges described in Prom Week.

Additionally we’ve also created 5 different Traits and 4 types of Status, each one influences the NPCs decision making, as shown in Figures 5 and 6.

The internal structure of the CIF-CK mod can be defined by three components:

Type	Name	Intent
Romantic	Flirt	Increase Romance
	Offer Romantic Gift	Increase Romance
	Ask Out	Add Status: "isDating"
	Share Feelings	Become Lovers
Friendly	Compliment	Increase Friendship
	Offer Gift	Increase Friendship
UnFriendly	Insult	Decrease Friendship
	Embarrass	Decrease Friendship
	Insult Other NPC	Improve Friendship and decreasing others' Friendship
Hostile	Fight	Drastically decrease Friendship
Special	Break-up	Stop being Lovers
	Hello	Greeting NPCs that haven't met

Table 4: Types of Quests/Social Moves available in the mod

Trait	Consequence
Friendly	Higher volition for Friendly Social Exchanges
Charming	Higher volition for Romantic Social Exchanges
Hostile	Higher volition for Unfriendly and Hostile Social Exchanges
Shy	Lower Volition for Romantic and Hostile Social Exchanges

Table 5: Traits’ Influence when calculating Social Exchanges Volitions

Status	Consequence
Embarrassed	Temporary Shy trait
Angry At	Higher volition for Unfriendly and Hostile Social Exchanges
Drunk	Higher probability of performing any Social Exchange
Dating	Higher volition for Romantic Social Exchanges

Table 6: Statuses’ Influence when calculating Social Exchanges Volitions

- CIF-CK script, the main script that manages the NPC’s decision making cycle.
- Influence Rules script, the auxiliary script that calculates all the volitions for all the social moves to be

performed by its owner NPC.

- GameManager script, manages the social state of the location that the Player is in. It receives information from all the “Social” NPCs and decides which one is going to perform a social interaction next, it also notifies all other NPCs of what has happened in the social state around them.

Every Character has both the CIF-CK script and Influence Rules. There is just one GameManager throughout all the gaming experience that is associated with the Player Character. The internal structured is described in Figure 3.

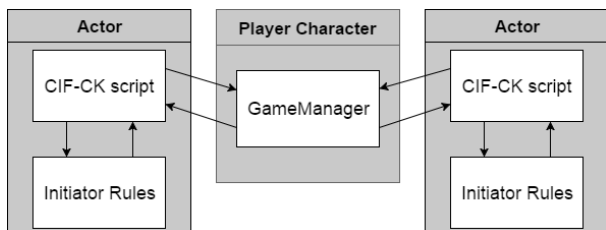


Figure 3: Internal Structure of the mod

4.1 Player to NPC interaction

CIF-CK considers, not just NPC to NPC interaction but also the Player to NPC interaction. When playing the CIF-CK mod the Player can interact with any Social (modded) NPC it wants. These interactions are done through dialogue. When the Player starts a conversation with an NPC it is presented some options that were added with this mod. The idea is to “mimic” the actions that NPC perform towards other NPCs and give the Player the option to perform some of those social moves. In the final version the Player can:

- Greet the NPC if they haven’t met yet.
- Offer the NPC a gift from the Player’s inventory.
- Compliment the NPC.
- Insult the NPC.
- Flirt with the NPC.
- Ask the NPC out on a date.
- Propose to the NPC.
- If they are on a relationship, break up with the NPC.
- “Bad Mouth” other NPCs around them.
- “Recommend” other NPCs around them to the NPC.

All dialogue options also come with a response. The NPC can either accept or reject any one of these “moves”. For example if the NPC does not trust the Player it will refuse any kind of “bad-mouth” coming from him. The other moves’ results are calculated using the CIFCKRules, just like a normal social move quest response. We can see some of the dialogue options, in game, in Figure 4.

5. VALIDATION

In order to test the impact of CIF-CK in the Player Experience we designed two different playable scenarios within the mod.

5.1 Play Scenarios

- **Quest Scenario**, the first scenario is a small Narrative experience that Players can work around, using CIF-CK, with specific Characters and within specific



Figure 4: Player dialogue options when speaking to a Social NPC.

situations. The idea behind this scenario to give the players a taste of what some of stories and Quests could be if they used a similar Narrative Structure to normal Skyrim Side-Quest but with Social NPCs instead.

- **Open Scenario**, the second scenario is a place where the Player can experience the addition of Social Ability to already existing NPCs within a more open and “sand-box” situation. This scenario is a lot closer to the Prom Week, discussed earlier in Section 2.4.1. The idea behind this is to try to understand if players interact more than they used to with previous NPCs. It is also of our interest to understand if players, without any “special” motivation, can create storylines on their own judging from the way NPCs react to the Player and to the other NPCs in the same location.

5.2 Mod Release

The mod containing CIF-CK implementation and the Player Scenarios is called “Social NPCs” and was released in the 26th of August, both in Steam Workshop⁵ and in the popular mod website: “Nexus Mods”⁶.

In the first 40 days after its release it had been played by more than 6 000 different players and more than 70 000 users had visited the mod’s web page. In Steam it reached a 93% approval rating, out of 194 ratings (181 positive ratings and 13 negative) in and became “Top Mod of the Week” because of its high popularity and rating.

In total there were over 180 comments, spread across both platforms and Reddit, the vast majority of which were very positive and provided very encouraging feedback along with very interesting suggestions. Furthermore because there are so few mods of this kind, by doing a quick search in any of these platforms the mod can be found on the top of its category: “NPC” and it is the only mod that appears when the keyword “Social” is used.

5.3 Survey Results

In the mod description we provided a link so that users could participate in this project by completing our survey. We didn’t have a lot of answers when compared to the

⁵<http://steamcommunity.com/sharedfiles/filedetails/?id=751622677>

⁶<http://www.nexusmods.com/skyrim/mods/77792>

amount of people that played the mod, despite that we had some interesting and promising results.

The general CIF-CK mod user is a veteran Skyrim player that enjoys the game at least 5 hours a week and has played Skyrim more than 100 hours in total, additionally, 80% (99 out of 124 answers) of players have more than 20 mods activated each time they load the game.

Regarding the “Quest Scenario”, that we created, users felt that the Quest we designed was more flexible when compared to Normal Skyrim Quests (73%, 17 out of 23 answers). Additionally players enjoyed interacting with the NPCs (95% of the answers, 22 out of 23) and tried to manipulate them in order to achieve their own goals (91%, 21 out of 23 participants).

Because the “Open Scenario” added functionality to previous existing NPCs we used it to compare the differences users felt between Skyrim original NPCs and CIF-CK based “Social NPCs”. In order to achieve that goal we used a 5-point Likert scale twice, where users were asked to identify their level of agreement with a set of sentences. The phrases tested for believability and user interest. The results are shown in Figure 5 the first graph is regarding normal Skyrim NPCs and the second is regarding the CIF-CK NPCs.

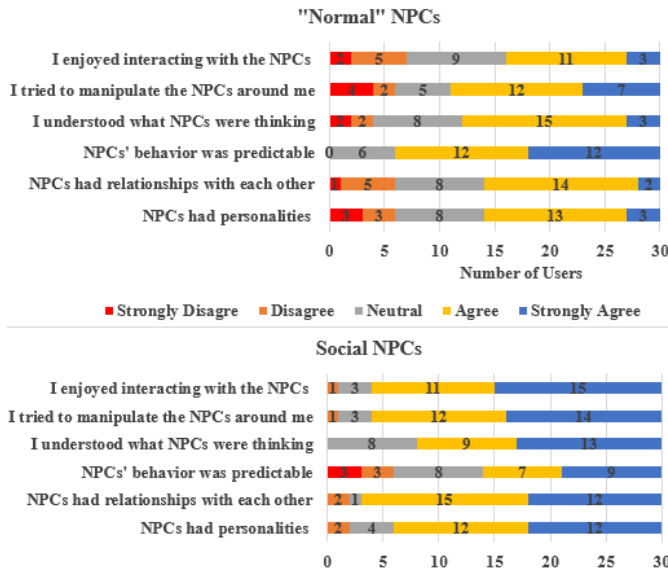


Figure 5: Using a 5-point Likert scale we asked users to try to measure their agreement level with the following statements

We used this data and performed Wilcoxon Signed Rank Tests in order to compare the difference between results:

- NPCs predictability levels were lower using Social NPCs when compared to Normal NPCs, $T = 26$, $p=0.008$, $r = 0.4873$
- Users comprehension levels were significantly higher using Social NPCs when compared to Normal NPCs, $T = 24$, $p=0.001$, $r = 0.582$
- For users, enjoyment levels were significantly higher using Social NPCs when compared to Normal NPCs, $T = 0$, $p=0.000007$, $r = 0.8188$

All results were significant with $p<0.01$ leading us to some very interesting conclusions. While the player’s perception

of the predictability of the NPCs decreased, their understanding of the NPCs’ actions increased significantly. Our interpretation of this result is that NPCs performed actions that the Player was not expecting⁷, however, these actions made sense and were plausible. Additionally almost every participant enjoyed interacting more with the CIF-CK based NPCs than with the “vanilla” ones.

In the final section of the survey we inquired users about their general experience with the mod, its results are shown in Figure 6.

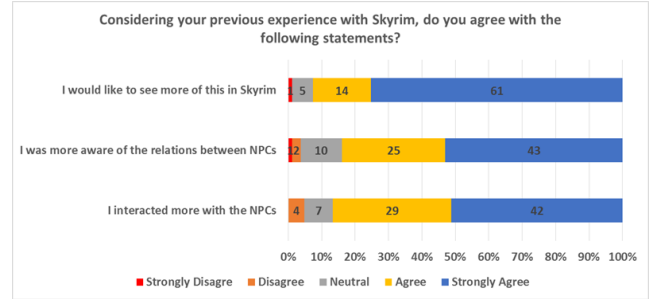


Figure 6: Using a 5-point Likert scale we asked users to try to measure their agreement level with the following statements

There is no doubt users enjoyed playing the mod and interacting with its Characters. Our mod adds something new to the NPCs present in Skyrim and that is something players clearly want. We could not anticipate such a great and positive response to this project. The amount of support and feedback we’ve received allows us to conclude that our goals were achieved and that we were successful in proving our thesis. Users feel the differences between Normal and Social Characters and prefer the latter, the CIF-CK architecture, in their gaming experiences.

6. CONCLUSIONS

We were successful in adapting the Comme il Faut architecture to Skyrim, creating the CIF-CK architecture, and published the resulting mod online making it available to every user to experiment. The mod is called “Social NPCs” and was released online in the 25th of August both in Steam Workshop and in a popular mod website: “Nexus Mods”.

The mod’s reception and popularity lead us to conclude that this project was successful in achieving its goals. Additionally the on-line surveys support our conclusion, we were able to implement a social architecture model in a modern and commercial video game and it did improve the Player experience. Using a Social Simulation architecture we were able to create more believable characters and give the player more interesting choices.

If a MSc student can do a project such as this one in nine months, with no budget whatsoever and no access to the source code, and can have very popular and successful results, maybe it is time for video game companies to invest in this area and develop or use already developed Social Simulation Systems in their games.

⁷Due, probably, to the normally low levels of NPC interactivity and authoring

REFERENCES

- [1] N. Afonso and R. Prada. *Agents That Relate: Improving the Social Believability of Non-Player Characters in Role-Playing Games*, pages 34–45. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [2] R. Aylett, M. Kriegel, M. Y. Lim, J. Dias, K. Leichtenstern, W. C. Ho, and P. Rizzo. Orient: interactive agents for stage-based role-play. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 1371–1372. International Foundation for Autonomous Agents and Multiagent Systems, 2009.
- [3] A. Bresó, J. Martínez-Miranda, C. Botella, R. Baños, and J. García-Gómez. Usability and acceptability assessment of an empathic virtual agent to prevent major depression. *Expert Systems*, 2016.
- [4] S.-H. Chen. Skyrim survival needs modification. 2015.
- [5] J. Dias, S. Mascarenhas, and A. Paiva. Fatima modular: Towards an agent architecture with a generic appraisal framework. In *Emotion Modeling*, pages 44–56. Springer, 2014.
- [6] A. Haddadi and K. Sundermeyer. Belief-desire-intention agent architectures. *Foundations of distributed artificial intelligence*, pages 169–185, 1996.
- [7] K. Isbister. *Better game characters by design: A psychological approach*. Elsevier/Morgan Kaufmann, 2006.
- [8] A. Kuo, J. L. Hiler, and R. J. Lutz. From super mario to skyrim: A framework for the evolution of video game consumption. *Journal of Consumer Behaviour*, 2016.
- [9] A. B. Loyall and J. Bates. Personality-rich believable agents that use language. In *Proceedings of the first international conference on Autonomous agents*, pages 106–113. ACM, 1997.
- [10] S. C. Marsella, D. V. Pynadath, and S. J. Read. Psychsim: Agent-based modeling of social interactions and influence. In *Proceedings of the international conference on cognitive modeling*, volume 36, pages 243–248, 2004.
- [11] S. Mascarenhas, R. Prada, P. Santos, K. Star, and M. Freire Moran. D3. 5–first storytelling framework. 2016.
- [12] M. Mateas and J. McCoy. An architecture for character-rich social simulation. In S. Rabin, editor, *Game AI Pro: Collected Wisdom of Game AI Professionals*, chapter 43, pages 515–530. A. K. Peters, Ltd., Natick, MA, USA, 2013.
- [13] M. Mateas and A. Stern. *Façade*. Microsoft Windows , Macintosh, 2005.
- [14] J. McCoy, M. Mateas, and N. Wardrip-Fruin. Comme il faut: A system for simulating social games between autonomous characters. In *Proceedings of the 8th Digital Art and Culture Conference (DAC)*, Irvine, CA, December 12-15 2009.
- [15] J. McCoy, M. Treanor, B. Samuel, M. Mateas, and N. Wardrip-Fruin. Prom week: social physics as gameplay. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 319–321. ACM, 2011.
- [16] J. McCoy, M. Treanor, B. Samuel, B. Tearse, M. Mateas, and N. Wardrip-Fruin. Authoring game-based interactive narrative using social games and comme il faut. 2010.
- [17] J. McCoy, M. Treanor, B. Samuel, N. Wardrip-Fruin, and M. Mateas. Comme il faut: A system for authoring playable social models. In *AIIDE*, 2011.
- [18] J. L. Moreno. Who shall survive?: A new approach to the problem of human interrelations. 1934.
- [19] A. Paiva, J. Dias, D. Sobral, R. Aylett, S. Woods, L. Hall, and C. Zoll. Learning by feeling: Evoking empathy with synthetic characters. *Applied Artificial Intelligence*, 19(3-4):235–266, 2005.
- [20] D. V. Pynadath and S. C. Marsella. Psychsim: Modeling theory of mind with decision-theoretic agents. In *IJCAI*, volume 5, pages 1181–1186, 2005.
- [21] S. H. Rodrigues, S. F. Mascarenhas, J. Dias, and A. Paiva. “i can feel it too!”: Emergent empathic reactions between synthetic characters. In *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, pages 1–7. IEEE, 2009.
- [22] C. Roth, C. Klimmt, I. E. Vermeulen, and P. Vorderer. The experience of interactive storytelling: comparing “fahrenheit” with “façade”. In *International Conference on Entertainment Computing*, pages 13–21. Springer, 2011.
- [23] K. R. Scherer, A. Schorr, and T. Johnstone. *Appraisal processes in emotion: Theory, methods, research*. Oxford University Press, 2001.
- [24] B. G. Studios. The elder scrolls v: Skyrim. Microsoft Windows, Playstation 3, Xbox 360, 2011.
- [25] A. Sullivan, A. Grow, T. Chirrick, M. Stokols, N. Wardrip-Fruin, and M. Mateas. Extending crpgs as an interactive storytelling form. In *Interactive Storytelling*, pages 164–169. Springer, 2011.
- [26] P. R. Thrainsson, A. L. Petursson, and H. H. Vilhjálmsson. Dynamic planning for agents in games using social norms and emotions. In *Intelligent Virtual Agents*, pages 473–474. Springer, 2011.
- [27] G. N. Yannakakis and J. Togelius. A panorama of artificial and computational intelligence in games. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(4):317–335, 2015.